The Internet Protocol Journal

June 1998 Volume 1, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

Welcome to the first edition of *The Internet Protocol Journal* (IPJ). This publication is designed to bring you in-depth technical articles on current and emerging Internet and intranet technologies. We will publish technology tutorials, as well as case studies on all aspects of internetworking.

Our first article is a detailed look at *Virtual Private Networks* (VPNs). Many organizations are turning to VPNs as a cost-effective way to implement enterprise networking, but the industry has not yet settled for a single approach, nor even a single definition of the VPN concept. The article by Paul Ferguson and Geoff Huston is in two parts. Part II will follow in our second issue, due out in September.

When the Internet Protocol suite (TCP/IP) was first designed, security was not a major consideration. Indeed, the primary goal in the early days of networking was sharing of information among academics and researchers. Today, TCP/IP is being used for mission-critical applications and for the emerging area of electronic commerce. As a result, security mechanisms are being added at all levels of the protocol stack. In this issue, we take a closer look at the *Secure Sockets Layer* (SSL), which is used for Web transactions. William Stallings explains how SSL works and how it is becoming the standard for Web security.

If you want to learn about computer networks, many options are available, including conferences, journals, standards documents, Web sites, glossaries and, of course, books. Our *Fragments* page gives you some pointers for further reading, and every issue will include at least one book review.

A detailed description of the scope of this journal can be found on page 30 in our *Call for Papers*. We want your input in this new publication. Please send comments, suggestions or questions to `ipj@cisco.com`. You may also use this address to request a complimentary copy of the next issue of IPJ. If you would like to write an article, send me e-mail and I will send you author guidelines.

—Ole J. Jacobsen, Editor and Publisher
`ole@cisco.com`

What Is a VPN? — Part I

by Paul Ferguson, Cisco Systems
and Geoff Huston, Telstra

The term "VPN," or *Virtual Private Network*, has become almost as recklessly used in the networking industry as has "QoS" (Quality of Service) to describe a broad set of problems and "solutions," when the objectives themselves have not been properly articulated. This confusion has resulted in a situation where the popular trade press, industry pundits, and vendors and consumers of networking technologies alike generally use the term VPN as an offhand reference for a set of different technologies. This article provides a common-sense definition of a VPN, and an overview of different approaches to building one.

> "The wonderful thing about virtual private networks is that its myriad definitions give every company a fair chance to claim that its existing product is actually a VPN. But no matter what definition you choose, the networking buzz-phrase doesn't make sense. The idea is to create a private network via tunneling and/or encryption over the public Internet. Sure, it's a lot cheaper than using your own frame relay connections, but it works about as well as sticking cotton in your ears in Times Square and pretending nobody else is around."[1]

A Common-Sense Definition

As *Wired Magazine* notes in the quotation, the myriad definitions of a VPN are less than helpful in this environment. Accordingly, it makes sense to begin this examination of VPNs to see if it is possible to provide a common-sense definition of a VPN. Perhaps the simplest method of attempting to arrive at a definition for VPNs is to look at each word in the acronym individually, and then tie each of them together in a simple, common-sense, and meaningful fashion.

Let's start by examining the word "network." This term is perhaps the least difficult one for us to define and understand, because the commonly accepted definition is fairly uncontroversial and generally accepted throughout the industry. A network consists of any number of devices that can communicate through some arbitrary method. Devices of this nature include computers, printers, routers, and so forth, and they may reside in geographically diverse locations. They may communicate in numerous ways because the electronic signaling specifications, and data-link, transport, and application-layer protocols are countless. For the purposes of simplicity, let's say that a "network" is a collection of devices that can communicate in some fashion, and can successfully transmit and receive data among themselves.

The term "private" is fairly straightforward, and is intricately related to the concept of "virtualization" insofar as VPNs are concerned, as we'll discuss in a moment. In the simplest of definitions, "private" means communications between two (or more) devices is, in some

fashion, secret—that the devices that are not participating in the "private" nature of communications are not privy to the communicated content, and that they are indeed completely unaware of the private relationship altogether. Accordingly, data privacy and security (data integrity) are also important aspects of a VPN that need to be considered when implementing any particular VPN.

Another means of expressing this definition of "private" is through its antonym, "public." A "public" facility is one that is openly accessible, and is managed within the terms and constraints of a common public resource, often via a public administrative entity. By contrast, a private facility is one where access is restricted to a defined set of entities, and third parties cannot gain access. Typically, the private resource is managed by the entities who have exclusive right of access. Examples of this type of private network can be found in any organizational network that is not connected to the Internet, or to any other external organizational network, for that matter. These networks are private because there is no external connectivity, and thus no external network communications.

Another important aspect of privacy in a VPN is through its technical definition. For example, privacy in an addressing and routing system means that the addressing used within a VPN community of interest is separate and discrete from that of the underlying shared network, and from that of other VPN communities. The same holds true for the routing system used within the VPN and that of the underlying shared network. The routing and addressing scheme within a VPN should, in general, be self-contained, but this scenario degenerates into a philosophical discussion of the context of the term "VPN." Also, it is worthwhile to examine the differences between the "peer" and "overlay" models of constructing VPNs—both of which are discussed in more detail later under the heading "Network-Layer VPNs."

"Virtual" is a concept that is slightly more complicated. *The New Hacker's Dictionary* (formerly known as the Jargon File)[2] defines virtual as:

> virtual /adj./ [via the technical term "virtual memory," prob. from the term "virtual image" in optics] 1. Common alternative to {logical}; often used to refer to the artificial objects (like addressable virtual memory larger than physical memory) simulated by a computer system as a convenient way to manage access to shared resources. 2. Simulated; performing the functions of something that isn't really there. An imaginative child's doll may be a virtual playmate. Oppose {real}.

Insofar as VPNs are concerned, the second definition is perhaps the most appropriate comparison for virtual networks. The "virtualization" aspect is one that is similar to what we briefly described previously as private, but the scenario is slightly modified—the private communication is now conducted across a network infrastructure that

is shared by more than a single organization. Thus, the private resource is actually constructed by using the foundation of a logical partitioning of some underlying common, shared resource rather than by using a foundation of discrete and dedicated physical circuits and communications services. Accordingly, the private network has no corresponding private physical communications system. Instead, the private network is a virtual creation that has no physical counterpart.

The virtual communications between two (or more) devices is because the devices that are not participating in the virtual communications are not privy to the content of the data, and they are also altogether unaware of the private relationships between the virtual peers. The shared network infrastructure could, for example, be the global Internet and the number of organizations or other users not participating in the virtual network may literally number into the thousands or even millions.

A VPN can also said to be a discrete network[3]:

> (discrete \dis*crete"\, a. [L. discretus, p.p. of discernere. See Discreet.] 1. Separate; distinct; disjunct).

The discrete nature of VPNs allows both privacy and virtualization. Although VPNs are not completely separate, intrinsically, the distinction is that they operate in a discrete fashion across a shared infrastructure, providing exclusive communications environments that do not share any points of interconnection.

The combination of these terms produces VPN—a private network, where the privacy is introduced by some method of virtualization. A VPN could be built between two end systems or between two organizations, between several end systems within a single organization or between multiple organizations across the global Internet, between individual applications, or any combination.

It should be noted that there is really no such thing as a nonvirtual network, if the underlying common public transmission systems and other similar public infrastructure components are considered to be the base level of carriage of the network. What separates a VPN from a truly private network is whether the data transits a shared versus a nonshared infrastructure. For instance, an organization could lease private line circuits from various telecommunications providers and build a private network on the base of these private circuit leases, but the circuit-switched network owned and operated by the telecommunications companies are actually circuits connected to their *Digital Access and Crossconnect Systems* (DACSs) network and subsequently their fiber-optics infrastructure. This infrastructure is shared by any number of organizations through the use of multiplexing technologies. Unless an organization is actually deploying private fiber and layered transmission systems, any network is layered with "virtualized" connectivity services in this fashion.

A VPN doesn't necessarily mean communications isolation, but rather the controlled segmentation of communications for communities of interest across a shared infrastructure.

The common and somewhat formal characterization of the VPN, and perhaps the most straightforward and strict definition, follows:

> A VPN is a communications environment in which access is controlled to permit peer connections only within a defined community of interest, and is constructed though some form of partitioning of a common underlying communications medium, where this underlying communications medium provides services to the network on a nonexclusive basis.

A simpler, more approximate, and much less formal description follows:

> A VPN is private network constructed within a public network infrastructure, such as the global Internet.

It should also be noted that although VPNs may be constructed to address any number of specific business needs or technical requirements, a comprehensive VPN solution provides support for dial-in access, support for multiple remote sites connected by leased lines (or other dedicated means), the ability of the VPN service provider (SP) to "host" various services for the VPN customers (for example, Web hosting), and the ability to support not just intra-, but also inter-VPN connectivity, including connectivity to the global Internet.

VPN Motivations

There are several motivations for building VPNs, but a common thread is that they all share the requirement to "virtualize" some portion of an organization's communications—in other words, make some portion (or perhaps all) the communications essentially "invisible" to external observers, while taking advantage of the efficiencies of a common communications infrastructure.

The base motivation for VPNs lies in the economics of communications. Communications systems today typically exhibit the characteristic of a high fixed-cost component, and smaller variable-cost components that vary with the transport capacity, or bandwidth, of the system. Within this economic environment, it is generally financially attractive to bundle numerous discrete communications services onto a common, high-capacity communications platform, allowing the high fixed-cost components associated with the platform to be amortized over a larger number of clients. Accordingly, a collection of virtual networks implemented on a single common physical communications plant is cheaper to operate than the equivalent collection of smaller, physically discrete communications plants, each servicing a single network client.

Therefore, if aggregation of communications requirements leads to a more cost-effective communications infrastructure, why not pool all these services into a single public communications system? Why is there still the requirement to undertake some form of partitioning within this common system that results in these "virtual private" networks?

In response to this question, the second motivation for VPNs is that of communications privacy, where the characteristics and integrity of communications services within one closed environment is isolated from all other environments that share the common underlying plant. The level of privacy depends greatly on the risk assessment performed by the subscriber organization—if the requirement for privacy is low, then the simple abstraction of discretion and network obscurity may serve the purpose. However, if the requirement for privacy is high, then there is a corresponding requirement for strong security of access and potentially strong security applied to data passed over the common network.

History

This article cannot do justice to the concept of VPNs without some historical perspective, so we need to look at why VPNs are an evolving paradigm, and why they will continue to be an issue of confusion, contention, and disagreement. This examination is important because opinions on VPN solutions are quite varied, as well as how they should be approached.

Historically, one of the precursors to the VPN was the *Public Data Network* (PDN), and the current familiar instance of the PDN is the global Internet. The Internet creates a ubiquitous connectivity paradigm, where the network permits any connected network entity to exchange data with any other connected entity. The parallels with the global *Public Switched Telephone Network* (PSTN) are, of course, all too obvious—where a similar paradigm of ubiquitous public access is the predominate characteristic of the network.

The Public Data Network has no inherent policy of traffic segregation, and any modification to this network policy of permitting ubiquitous connectivity is the responsibility of the connecting entity to define and enforce. The network environment is constructed using a single addressing scheme and a common routing hierarchy, which allows the switching elements of the network to determine the location of all connected entities. All these connected entities also share access to a common infrastructure of circuits and switching.

However, the model of ubiquity in the "Internet PDN" does not match all potential requirements, especially the need for data privacy. For organizations that wish to use this public network for private purposes within a closed set of participants (for example, connecting a set of geographically separated offices), the Internet is not always a palatable possibility. Numerous factors are behind this mismatch, including issues of Quality of Service (QoS), availability and reliability, use of

public addressing schemes, use of public protocols, site security, and data privacy and integrity (the possibility of traffic interception). Additionally, a corporate network application may desire more stringent levels of performance management than are available within the public Internet, or indeed may wish to define a management regime that differs from that of the underlying Internet PDN.

Service-Level Agreements

It is worthwhile at this point to briefly examine the importance of *Service-Level Agreements* (SLAs) in regards to the deployment of VPNs. SLAs are negotiated contracts between VPN providers and their subscribers; they contain the service criteria to which the subscriber expects specific services to be delivered. The SLA is arguably the only binding tool at the subscriber's disposal with which to ensure that the VPN provider delivers the service(s) to the level and quality as agreed, and it is in the best interest of the subscribers to monitor the criteria outlined in the SLA for compliance. However, SLAs present some challenging technical issues for both the provider and the subscriber.

For the subscriber, the challenge is to devise and operate service measurement tools that can provide a reasonable indication as to what extent the SLA is being honored by the provider. Also, it should be noted that a subscriber may use an SLA to bind one or more providers to a contractual service level, but if the subscriber's VPN spans multiple providers' domains, the SLA must also encompass the issue of provider interconnection and the end-to-end service performance.

For the provider, the challenge lies in honoring multiple SLAs from a number of service providers. In the case of an Internet PDN provider, the common mode of best-effort service levels is not conducive to meeting SLAs, given the unpredictable nature of the host's resource allocation mechanisms. In such environments, the provider either has to ensure that the network is generously engineered in terms of the ratio of subscriber access capacity to internal switching capacity, or the provider can deploy service differentiation structures to ensure that minimum resource levels are allocated to each SLA subscriber. It must be noted that the former course of action does tend to reduce the benefit of aggregation of traffic, which in turn has an ultimate cost implication, while the latter course of action has implications in terms of operational management complexity and scalability of the network.

Alternatives to the VPN

The alternative to using the Internet as a VPN today is to lease circuits, or similar dedicated communications services, from the public network operators (the local telephone company in most cases), and create a completely private network. It is a layering convention that allows us to label this as "completely private," because these dedicated communications services are (at the lower layers of the protocol

stack) again instances of virtual private communications systems constructed atop a common transmission bearer system. Of course, this scenario is not without precedent, and it must be noted that most of the early efforts in data networking, and many of the current data networking architectures, do not assume a deployment model of ubiquitous public access.

It is interesting to note that this situation is odd, when you consider that the inherent value of an architecture where ubiquitous public access over a chaotic collection of closed private networks had been conclusively demonstrated in the telephony marketplace since the start of the 20th century. Although the data communications industry appears to be moving at a considerable technological pace, the level of experiential learning, and consequent level of true progress as distinct from simple motion, still leaves much to be desired!

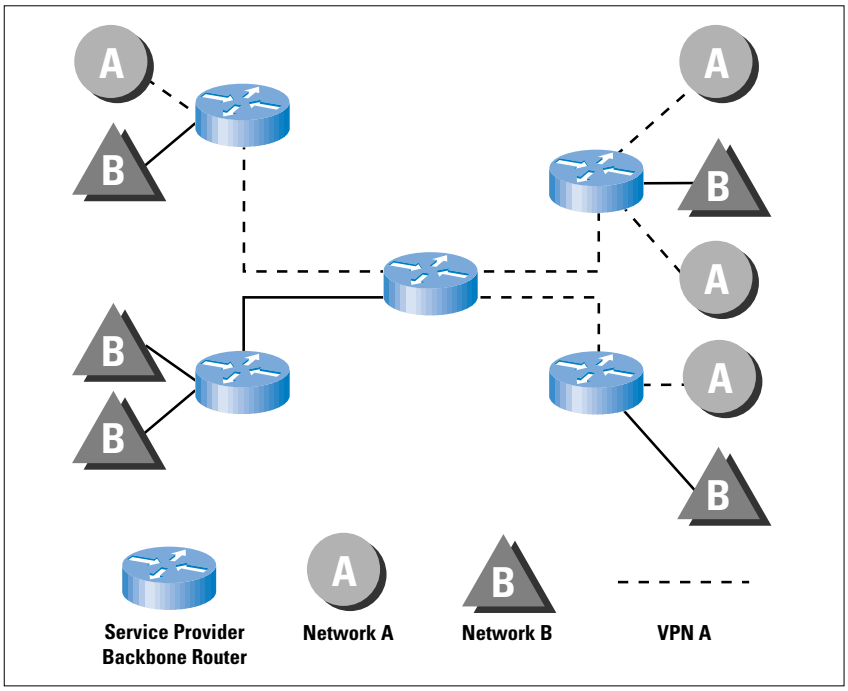
Instead of a public infrastructure deployment, the deployment model used has been that of a closed (or private) network environment where the infrastructure, addressing scheme, management, and services were dedicated to a closed set of subscribers. This model matched that of a closed corporate environment, where the network was dedicated to serve a single corporate entity as the sole client. This precursor to the VPN, which could be called the private data network, was physically constructed using dedicated local office wiring and dedicated leased circuits (or private virtual circuits from an underlying switching fabric such as X.25) to connect geographically diverse sites.

However, this alternative does have an associated cost, in that the client now has to manage the network and all its associated elements, invest capital in network switching infrastructure, hire trained staff, and assume complete responsibility for the provisioning and ongoing maintenance of the network service. Such a dedicated use of transport services, equipment, and staff is often difficult to justify for many small-to-medium sized organizations, and whereas the functionality of a private network system is required, the expressed desire is to reduce the cost of the service through the use of shared transport services, equipment, and management. Numerous scenarios can address this need, ranging from outsourcing the management of the switching elements of the network (managed network services), to outsourcing the capital equipment components (leased network services), to outsourcing the management, equipment, and transport elements to a service provider altogether.

An Example VPN

In the simple example illustrated in Figure 1, Network "A" sites have established a VPN (depicted by the dashed lines) across the service provider's backbone network, where Network "B" is completely unaware of its existence. Both Networks "A" and "B" can harmoniously coexist on the same backbone infrastructure.

Figure 1:
A Virtual Private
Network of
"A" Sites



This type of VPN is, in fact, the most common type of VPN—one that has geographically diverse subnetworks that belong to a common administrative domain, interconnected by a shared infrastructure outside their administrative control (such as the global Internet or a single service provider backbone). The principal motivation in establishing a VPN of this type is that perhaps most of the communications between devices within the VPN community may be sensitive (again, a decision on the level of privacy required rests solely on a risk analysis performed by the administrators of the VPN), yet the total value of the communications system does not justify the investment in a fully private communications system that uses discrete transmission elements.

On a related note, the level of privacy that a VPN may enjoy depends greatly on the technology used to construct the VPN. For example, if the communications between each VPN subnetwork (or between each VPN host) is securely encrypted as it transits the common communications infrastructure, then it can be said that the privacy aspect of the VPN is relatively high.

In fact, the granularity of a VPN implementation can be broken down further to a single end-to-end, one-to-one connectivity scenario. Examples of these types of one-to-one VPNs are single dialup users who establish a VPN connection to a secure application, such as an online banking service, or a single user establishing a secure, encrypted session between a desktop and server application, such as a purchasing transaction conducted on the World Wide Web. This type of one-to-one VPN is becoming more and more prevalent as secure electronic commerce applications become more mature and are further deployed in the Internet. (See article starting on page 20.)

It is interesting to note that the concept of virtualization in networking has also been considered in regard to deploying both research and production services on a common infrastructure. The challenge in the research and education community is one in which there is a need to satisfy both network research and production requirements. VPNs have also been considered as a method to segregate traffic in a network such that research and production traffic behave as "ships in the night," oblivious to one another's existence, to the point that major events (for example, major failures, instability) within one community of interest are completely transparent to the other. This concept is further documented in MORPHnet[4].

It should also be noted that VPNs may be constructed to span more than one host communications network, so that the "state" of the VPN may be supported on one or more VPN provider networks. This scenario is perhaps at its most robust when all the providers explicitly support the resultant distributed VPN environment, but other solutions that do not necessarily involve knowledge of the overlay VPN are occasionally deployed with mixed results.

Types of VPNs

The confusion factor comes into play in the most basic discussions regarding VPNs, principally because there are actually several different types of VPNs, and depending on the functional requirements, several different methods of constructing each type of VPN are available. The process of selection should include consideration of what problem is being solved, risk analysis of the security provided by a particular implementation, issues of scale in growing the size of the VPN, and the complexity involved in implementation of the VPN, as well as ongoing maintenance and troubleshooting.

To simplify the description of the different types of VPNs, they are broken down in this article into categories that reside in the different layers of the TCP/IP protocol suite; Link Layer, Network Layer, Transport Layer, and Application Layer.

Network-Layer VPNs

The network layer in the TCP/IP protocol suite consists of the IP routing system—how reachability information is conveyed from one point in the network to another. There are a few methods to construct VPNs within the network layer—each is examined in the following paragraphs. A brief overview of non-IP VPNs is provided in Part II of this article.

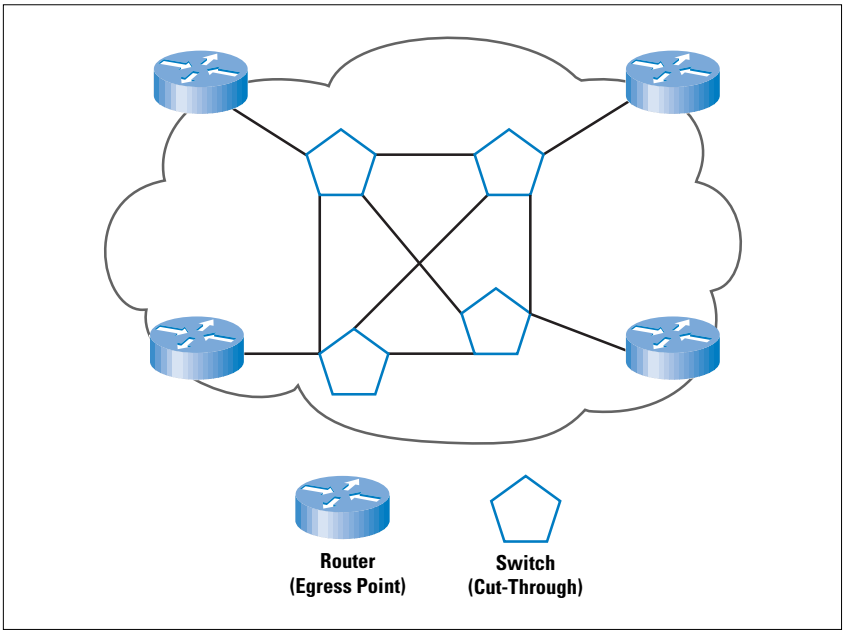
A brief overview of the differences in the "peer" and "overlay" VPN models is appropriate at this point. Simply put, the "peer" VPN model is one in which the network-layer forwarding path computation is done on a hop-by-hop basis, where each node in the intermediate data transit path is a peer with a next-hop node. Traditional routed networks are examples of peer models, where each router in the network

path is a peer with its next-hop adjacencies. Alternatively, the "overlay" VPN model is one in which the network-layer forwarding path is not done on a hop-by-hop basis, but rather, the intermediate link-layer network is used as a "cut-through" to another edge node on the other side of a large cloud. Examples of "overlay" VPN models include ATM, Frame Relay, and tunneling implementations.

Having drawn these simple distinctions between the peer and overlay models, it should be noted that the overlay model introduces some serious scaling concerns in cases where large numbers of egress peers are required because the number of adjacencies increases in direct proportion to the number of peers—the amount of computational and performance overhead required to maintain routing state, adjacency information, and other detailed packet forwarding and routing information for each peer becomes a liability in very large networks. If all the egress nodes in a cut-through network become peers in an effort to make all egress nodes one "Layer 3" hop away from one another, the scalability of the VPN overlay model is limited quite remarkably.

For example, as the simple diagram (Figure 2) illustrates, the routers that surround the interior switched infrastructure represent egress peers, because the switches in the core interior could be configured such that all egress nodes are one Layer 3 hop away from one another, creating what is commonly known as a "cut-through." This scenario forms the foundation of an overlay VPN model.

Figure 2: A Cut-Through VPN



Alternatively, if the switches in the interior are replaced with routers, then the routers positioned at the edge of the cloud become peers with their next-hop router nodes, not other egress nodes. This scenario forms the foundation of the peer VPN model.

Controlled Route Leaking

"Controlled route leaking" (or *route filtering*) is a method that could also be called "privacy through obscurity" because it consists of nothing more than controlling route propagation to the point that only certain networks receive routes for other networks that are within their own community of interest. This model can be considered a "peer" model, because a router within a VPN site establishes a routing relationship with a router within the VPN provider's network, instead of an edge-to-edge routing peering relationship with routers in other sites of that VPN. Although the common underlying Internet generally carries the routes for all networks connected to it, this architecture assumes that only a subset of such networks form a VPN. The routes associated with this set of networks are filtered such that they are not announced to any other set of connected networks, and all other non-VPN routes are not announced to the networks of the VPN. For example, in Figure 1, if the SP routers "leaked" routing information received from one site in Network "A" to only other sites in Network "A," then sites not in Network "A" (for instance, sites in Network "B") would have no explicit knowledge of any other networks which where attached to the service provider's infrastructure (as shown in Figure 3). Given this lack of explicit knowledge of reachability to any location other than other members of the same VPN, privacy of services is implemented by the inability of any of the VPN hosts to respond to packets which contain source addresses from outside the VPN community of interest.

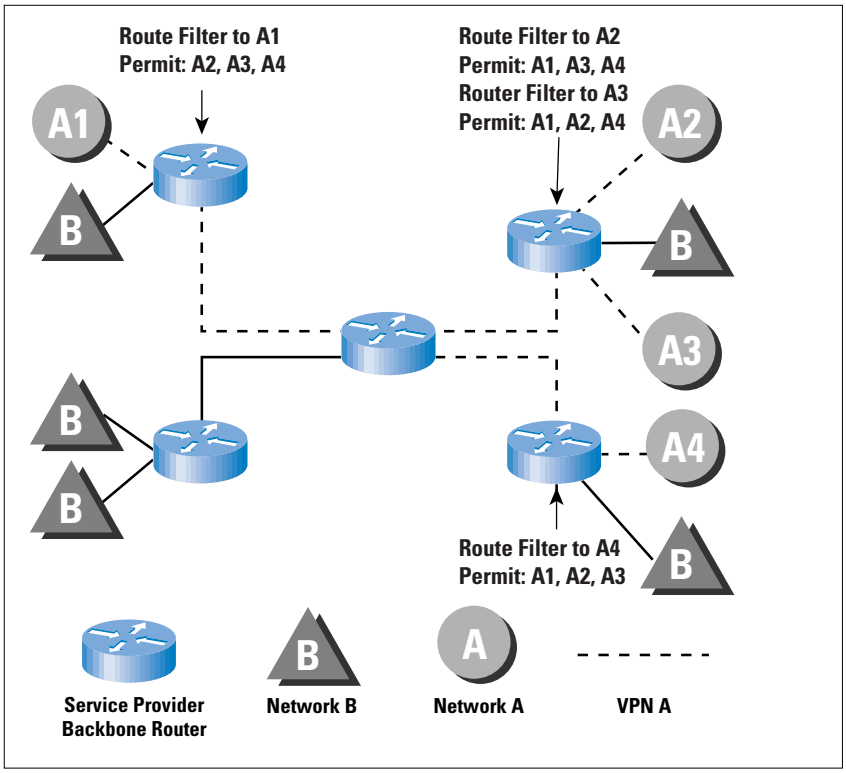


Figure 3: Controlled Route Leaking

This use of partial routing information is prone to many forms of misconfiguration. One potential problem with route leaking is that it is extremely difficult, if not impossible, to prohibit the subscriber networks from pointing default to the upstream next-hop router for traffic destined for networks outside their community of interest. From within the VPN subscriber's context, this action may be reasonable, in that "default" for the VPN is reachability to all other members of the same VPN, and pointing a default route to the local egress path is, within a local context, a reasonable move. Thus, it is no surprise that this is a common occurrence in VPNs in which the customer configures and manages the customer premise equipment (CPE) routers. If the SP manages the configuration of the CPE routers, then this is rarely a problem. Otherwise, the SP might be wise to place traffic filters on first-hop routers to prohibit all traffic destined for networks outside the VPN community of interest.

It should also be noted that this environment implicitly assumes a common routing core. A common routing core, in turn, implies that each VPN must use addresses that do not clash with those of any other VPN on the same common infrastructure, and cannot announce arbitrary private addresses into the VPN. Another, perhaps less obvious, side effect of this form of VPN structure is that it is not possible for two VPNs to have a single point of interconnection, nor is it possible for a VPN to operate a single point of interconnection to the public Internet in such an environment. (This single point would be a so-called "gateway," where all external traffic is passed through a control point that can enforce some form of access policy and record a log of external transactions.) The common routing core uses a single routing paradigm, based solely on destination address.

It should also be noted that this requirement highlights one of the dichotomies of VPN architectures. VPNs must assume that they operate in a mutually hostile environment, where any vulnerability that exposes the private environment to access by external third parties may be exploited in a hostile fashion. However, VPNs rarely are truly isolated communications environments, and typically all VPNs do have some form of external interface that allows controlled reachability to other VPNs and to the broader public data network. The trade-off between secure privacy and the need for external access is a constant feature of VPNs.

Implementation of inter-VPN connectivity requires the network to route externally originated packets to the VPN interconnection point, and if they are admitted into the VPN at the interconnection point, the same packet may be passed back across the network to the ultimate VPN destination address. Without the use of *Network Address Translation* (NAT) technologies at the interconnection point of ingress into the VPN, this kind of communications structure is insupportable within this architecture (Figure 4).

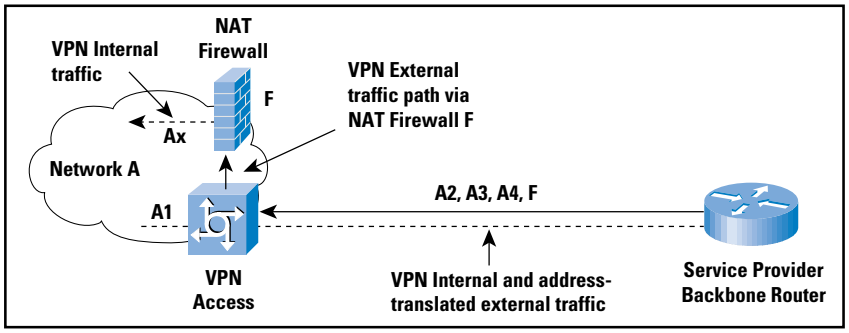


Figure 4: Segregating VPN traffic via address translation

In general, the technique of supporting private communities of interest simply by route filtering can at best be described as a primitive method of VPN construction, which is prone to administrative errors, and admits an undue level of insecurity and network inflexibility. Even with comprehensive traffic and route filtering, the resulting environment is not totally robust. The operational overhead required to support complementary sets of traditional routing and traffic filters is a relevant consideration, and this approach does not appear to possess the scaling properties desirable to allow the number of VPNs to grow beyond the bounds of a few hundred, using today's routing technologies.

Having said that, however, a much more scalable approach is to use *Border Gateway Protocol* (BGP) *communities*[5] as a method to control route propagation. The use of BGP communities scales much better than alternative methods with respect to controlling route propagation and is less prone to human misconfiguration. Briefly, the use of the BGP communities attribute allows a VPN provider to "mark" BGP *Network-Layer Reachability Information* (NLRI) with a community attribute, such that configuration control allows route information to propagate in accordance with a community profile.

Because traffic from different communities of interest must traverse a common shared infrastructure, there is no significant data privacy in the portion of the network where traffic from multiple communities of interest share the infrastructure. Therefore, it can be said that although connected subnetworks—or rather, subscribers to the VPN service—may not be able to detect the fact that there are other subscribers to the service, multiple interwoven streams of subscriber data traffic pass unprotected in the core of the service provider's network.

Tunneling

Sending specific portions of network traffic across a tunnel is another method of constructing VPNs. Some tunneling methods are more effective than others. The most common tunneling mechanisms are *Generic Routing Encapsulation* (GRE)[6] tunneling between a source and destination router, router-to-router or host-to-host tun-neling protocols such as *Layer 2 Tunneling Protocol* (L2TP)[7] and *Point-to-Point Tunneling Protocol* (PPTP)[8], and *Distance Vector Multicast Routing Protocol* (DVMRP)[9] tunnels.

Tunneling can be considered an overlay model, but the seriousness of the scaling impact depends on whether the tunnels are point-to-point or point-to-multipoint. Point-to-point tunnels have fewer scaling problems than do point-to-multipoint tunnels, except in situations where a single node begins to build multiple point-to-point tunnels with multiple endpoints. Although a linear scaling problem is introduced at this point, the manageability of point-to-point tunnels lies solely in the administrative overhead and the number of the tunnels themselves. On the other hand, point-to-multipoint tunnels use "cut-through" mechanisms to make greater numbers of endpoints one hop away from one another and subsequently introduce a much more serious scaling problem.

Although the *Multicast Backbone* (Mbone) itself could literally be considered a global VPN, and although DVMRP tunnels are still widely used by organizations to connect to the Mbone, it really is not germane to the central topic of VPNs, because the focus of this article is on unicast traffic.

Traditional Modes of Tunneling

GRE tunnels, as mentioned previously, are generally configured between a source (*ingress*) router and a destination (*egress*) router, such that packets designated to be forwarded across the tunnel (already formatted with an encapsulation of the data with the "normal" protocol-defined packet header) are further encapsulated with a new header (the GRE header), and placed into the tunnel with a destination address of the tunnel endpoint (the new next-hop). When the packet reaches the tunnel endpoint, the GRE header is stripped away, and the packet continues to be forwarded to the destination, as designated in the original IP packet header (Figure 5).

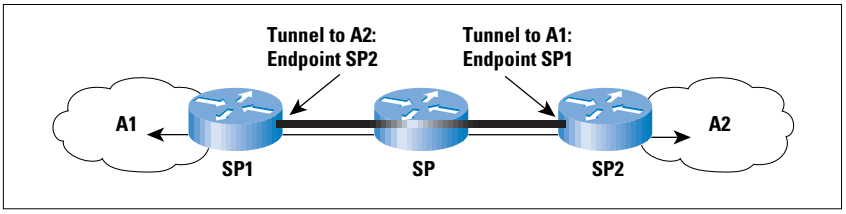


Figure 5: Tunneling across a Service Provider

GRE tunnels are generally point-to-point—that is, there is a single source address for the tunnel and usually only a single destination tunnel endpoint. However, some vendor implementations allow the configuration of point-to-multipoint tunnels—that is, a single source address and multiple destinations. Although this implementation is generally used in conjunction with *Next Hop Resolution Protocol* (NHRP)[10], the effectiveness and utility of NHRP is questionable and should be tested prior to deployment. It is also noteworthy that NHRP is known to produce steady-state forwarding loops when used to establish shortcuts between routers. In the scenario discussed previously, NHRP is used for establishing shortcuts between routers.

Tunnels, however, do have numerous compelling attractions when used to construct VPNs. The architectural concept is to create VPNs as a collection of tunnels across a common host network. Each point of attachment to the common network is configured as a physical link that uses addressing and routing from the common host network, and one or more associated tunnels. Each tunnel endpoint logically links this point of attachment to other remote points from the same VPN. The technique of tunneling uses a tunnel egress address defined within the address space of the common host network, whereas the packets carried within the tunnel use the address space of the VPN, which in turn constrains the tunnel endpoints to be collocated to those points in the network where the VPN and the host network interconnect.

Pros and Cons

The advantage of this approach is that the routing for the VPN is isolated from the routing of the common host network. The VPNs can reuse the same private address space within multiple VPNs without any cross impact, providing considerable independence of the VPN from the host network. This requirement is key for many VPNs in that private VPNs typically may not use globally unique or coordinated address space, and there is often the consequential requirement to support multiple VPNs which independently use the same address block. Such a configuration is not supportable within a controlled route leakage VPN architecture. The tunnel can also encapsulate numerous different protocol families, so that it is possible for a tunnel-based VPN to mimic much of the functionality of dedicated private networks. Again, the need to support multiple protocols in a format which preserves the functionality of the protocol is a critical requirement for many VPN support architectures. This requirement is one in which an IP common network with controlled route leakage cannot provide such services, whereas a tunneling architecture can segment the VPN-private protocol from the common host network. The other significant advantage of the tunneled VPN is the segregation of the common host routing environment with that of the VPN. To the VPN, the common host network assumes the properties of numerous point-to-point circuits, and the VPN can use a routing protocol across the virtual network which matches the administrative requirements of the VPN. Equally, the common host network can use a routing design which matches the administrative requirements of the host network (or collection of host networks), and is not constrained by the routing protocols used by the VPN client networks.

Although it could be said that these advantages indicate that GRE tunneling is the panacea for VPN design, using GRE tunnels as a mechanism for VPNs does have several drawbacks, mostly with regard to administrative overhead, scaling to large numbers of tunnels, and QoS and performance.

Since GRE tunnels must be manually configured, there is a direct relationship to the number of tunnels that must be configured and the amount of administrative overhead required to configure and maintain them—each time the tunnel endpoints must change, and they must be manually reconfigured. Also, although the amount of processing required to encapsulate a packet for GRE handling may appear to be small, there is a direct relationship to the number of configured tunnels and the total amount of processing overhead required for GRE encapsulation. Of course, tunnels can be structured to be triggered automatically, but such an approach has numerous drawbacks that dictate careful consideration of related routing and performance issues. The worst end state of such automatic tunnel generation is that of a configuration loop where the tunnel passes traffic over itself. It is important, once again, to reiterate the impact of a large number of routing peering adjacencies that result from a complete mesh of tunnels; this scenario can result in a negative effect on routing efficiency.

An additional concern with GRE tunneling is the ability of traffic classification mechanisms to identify traffic with a fine enough level of granularity, and not become a hindrance to forwarding performance. If the traffic classification process used to identify packets (that are to be forwarded across the tunnel) interferes with the router's ability to maintain acceptable packet-per-second forwarding rates, then this becomes a performance liability.

Privacy of the network remains an area of concern because the tunnel is still vulnerable—privacy is not absolute. Packets that use GRE formatting can be injected into the VPN from third-party sources. To ensure a greater degree of integrity of privacy of the VPN, it is necessary to deploy ingress filters that are aligned to the configured tunnel structure.

It is also necessary to ensure that the CPE routers are managed by the VPN service provider, because the configuration of the tunnel endpoints is a critical component of the overall architecture of integrity of privacy. However, most VPN service providers are reluctant to add CPE equipment to their asset inventory and undertake remote management of such CPE equipment, due to the high operational overheads and poor capital efficiencies which are typical of CPE deployment. Arguably, one might suggest that having a dedicated CPE router defeats one of the basic premises of constructing a VPN—the use of shared infrastructure as a way to reduce the overall network cost.

It should be noted that VPNs can be constructed using tunnels without the explicit knowledge of the host network provider, and the VPN can span numerous host networks without any related underlying agreements between the network operators to mutually support the overlay VPN. Such an architecture is little different from provider-operated VPN architecture; the major difference lies in the issue of traffic and

performance engineering, and the administrative boundary of the management of the VPN overlay. Independently configured VPN tunnels can result in injection of routes back into the VPN in a remote location, a scenario that can cause traffic to traverse the same link twice, once in an unencapsulated format and again within a tunnel. This situation can then lead to adverse performance impacts.

It is also true that the overlay VPN model has no control over which path is taken in the common host network, nor the stability of that path. This scenario can then lead to adverse performance impacts on the VPN. Aside from the technology aspects of this approach, the major issue is one of whether the VPN management is outsourced to the network provider, or undertaken within administrative functions of the VPN. One of the more serious considerations in building a VPN on tunneling is that there is virtually no way to determine the cost of the route across a tunnel, because the true path is masked by the cut-through nature of the tunnel. This situation could ultimately result in highly suboptimal routing, meaning that a packet could take a path determined by the cut-through mechanism that is excessively suboptimal, while native per-hop routing protocols might find a much more efficient method to forward the packets to their destinations.

Conclusion

So far in our discussion of VPNs, we have introduced a working definition of the term "Virtual Private Network" and discussed the motivations behind the adoption of such networks. We have outlined a framework for describing the various forms of VPNs, and then examined numerous network-layer VPN structures, in particular, that of controlled route leakage and tunneling techniques.

In Part II we will continue this examination of network-layer VPNs, including virtual private dial networks and network-layer encryption. In addition, we will examine link-layer VPNs that use ATM and Frame Relay substrates, and also look at switching and encryption techniques, and issues concerning QoS and non-IP VPNs.

References

[1] Steinberg, Steve G.,"Hype List—Deflating this month's overblown memes." *Wired Magazine*, 6.02, February 1998, p. 80. Ironically, number 1 on the Hype List is virtual private networks, with a life expectancy of 18 months.

[2] Raymond, Eric S., compiler. *The New Hacker's Dictionary, Third Edition*. MIT Press, ISBN 0-262-68092-0, 1996. The Jargon File online: `http://www.ccil.org/jargon/`

[3] *Webster's Revised Unabridged Dictionary* (1913). Hypertext Webster Gateway: `http://work.ucsd.edu:5141/cgi-bin/http_webster`

[4] Aiken, R., R. Carlson, I. Foster, T. Kuhfuss, R. Stevens, and L. Winkler. "Architecture of the Multi-Modal Organizational Research and Production Heterogeneous Network (MORPHnet)," Argonne National Laboratory, ECT and MCS Divisions, January 1997. http://www.anl.gov/ECT/Public/research/morphnt2.htm

[5] Chandra, R., P Traina, and T. Li. RFC 1997, "BGP Communities Attribute." August 1996; E. Chen and T. Bates. RFC 1998, "An Application of the BGP Community Attribute in Multi-home Routing." August 1996.

[6] Hanks, S., T. Li, D. Farinacci, and P. Traina. RFC 1701, "Generic Routing Encapsulation." October 1994; S. Hanks, T. Li, D. Farinacci, and P. Traina. RFC 1702, "Generic Routing Encapsulation over IPv4 networks." October 1994.

[7] Valencia, A., K. Hamzeh, A. Rubens, T. Kolar, M. Littlewood, W. M. Townsley, J. Taarud, G. S. Pall, B. Palter, and W. Verthein. "Layer Two Tunneling Protocol 'L2TP.'" draft-ietf-pppext-l2tp-10.txt, March 1998.

[8] Hamzeh, K., G. Singh Pall, W. Verthein, J. Taarud, and W. A. Little. "Point-to-Point Tunneling Protocol—PPTP." draft-ietf-pppext-pptp-02.txt, July 1997. See also: http://www.microsoft.com/backoffice/communications/morepptp.htm

[9] Waitzman, D., C. Partridge, and S. Deering. RFC 1075, "Distance Vector Multicast Routing Protocol." November 1988. For historical purposes, see also ftp://ftp.isi.edu/mbone/faq.txt

[10] Luciani, J., D. Katz, D. Piscitello, B. Cole, and N. Doraswamy. "NBMA Next Hop Resolution Protocol (NHRP)," draft-ietf-rolc-nhrp-15.txt, February 1998.

PAUL FERGUSON is a consulting engineer at Cisco Systems and an active participant in the Internet Engineering Task Force (IETF). His principal areas of expertise include large-scale network architecture and design, global routing, Quality of Service (QoS) issues, and Internet Service Providers. Prior to his current position at Cisco Systems, he worked in network engineering, analytical, and consulting capacities for Sprint, Computer Sciences Corporation (CSC), and NASA. He is coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, published by John Wiley & Sons, ISBN 0-471-24358-2, a collaboration with Geoff Huston. E-mail: ferguson@cisco.com

GEOFF HUSTON holds a B.Sc and a M.Sc from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Technologist in the Internet area for Telstra. He is also an active member of the IETF, and was an inaugural member of the Internet Society Board of Trustees. He is coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, published by John Wiley & Sons, ISBN 0-471-24358-2, a collaboration with Paul Ferguson. E-mail: gih@telstra.net

SSL: Foundation for Web Security

by William Stallings

Virtually all businesses, most government agencies, and many individuals now have Web sites. The number of individuals and companies with Internet access is expanding rapidly, and all of them have graphical Web browsers. As a result, businesses are enthusiastic about setting up facilities on the Web for electronic commerce. But the reality is that the Internet and the Web are extremely vulnerable to compromises of various sorts. As businesses utilize the Internet for more than information dissemination, they will need to use trusted security mechanisms.

An increasingly popular general-purpose solution is to implement security as a protocol that sits between the underlying transport protocol (TCP) and the application. The foremost example of this approach is the *Secure Sockets Layer* (SSL) and the follow-on Internet standard of SSL known as *Transport Layer Security* (TLS). At this level, there are two implementation choices. For full generality, SSL (or TLS) could be provided as part of the underlying protocol suite and therefore be transparent to applications. Alternatively, SSL can be embedded in specific packages. For example, Netscape and Microsoft Explorer browsers come equipped with SSL, and most Web servers have implemented the protocol. Although it is possible to use SSL for applications other than Web transactions, its use at present is typically as part of Web browsers and servers and hence limited to Web traffic. Most of this article deals with the technical details of SSL; the status of TLS is described at the end.

If you have viewed an HTML source document, you have seen that the links are referenced with `HREF=<URL>` within an anchor (A) tag. In most cases, the reference is to another document through the use of the *Hyper Text Transfer Protocol*, or HTTP. For this, the browser initiates one or more sessions to the destination port of TCP/80 (the well-known port for HTTP) on the server. In some cases, a plug-in can be called, and data specific to that plug-in can be transferred to or from the browser. For that, the browser would initiate a session to the well-known TCP port of the plug-in. SSL is called when the reference starts like the following: `HREF="https://..` By calling "https" within the browser, it is mandating that the data be transferred through the use of SSL. By clicking on this hot link, the browser initiates a session to the server on port TCP/443. SSL attempts to negotiate a secure link and transfers the data across it. If the negotiation fails, no data is transferred. The browser usually indicates that a secure connection has been requested. Netscape Navigator version 3 indicates this with a blue border around the page and a highlighted key in the lower left corner. Netscape Communicator version 4 displays this with a closed padlock in a lower status window. Microsoft's Internet Explorer indicates it

with a padlock in a lower information window. Display of these signs indicates that the information within the browser window has been delivered through the security of SSL.

SSL was originated by Netscape. Version 3 of the protocol was designed with public review and input from industry and was published as an Internet Draft document. Subsequently, when a consensus was reached to submit the protocol for Internet standardization, the TLS working group was formed within the *Internet Engineering Task Force* (IETF) to develop a common standard. The current work on TLS is aimed at producing an initial version as an Internet Standard. This first version of TLS can be viewed as essentially an SSLv3.1, and is very close to SSLv3. TLS includes a mechanism by which a TLS entity can back down to the SSLv3.0 protocol; in that sense, TLS is backward compatible with SSL.

SSL Architecture

SSL is designed to make use of TCP to provide a reliable end-to-end secure service. SSL is not a single protocol but rather two layers of protocols.

The SSL Record Protocol provides basic security services to various higher-layer protocols. In particular, the HTTP, which provides the transfer service for Web client/server interaction, can operate on top of SSL. Three higher-layer protocols are defined as part of SSL: the *Handshake Protocol*, the *Change CipherSpec Protocol*, and the *Alert Protocol*. These SSL-specific protocols are used in the management of SSL exchanges.

Two important SSL concepts are the SSL session and the SSL connection, which are defined in the specification as follows:

- Connection: A logical client/server link that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.

- Session: An association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

Between any pair of parties (applications such as HTTP on client and server), there may be multiple secure connections. In theory, there may also be multiple simultaneous sessions between parties, but this feature is not used in practice.

Several states are associated with each session. When a session is established, there is a current operating state for both read and write (that is, receive and send). In addition, during the Handshake Protocol, pending read and write states are created. Upon successful conclusion of the Handshake Protocol, the pending states become the current states. A session state is defined by the following parameters (definitions taken from the SSL specification):

- Session identifier: An arbitrary byte sequence chosen by the server to identify an active or resumable session state.
- Peer certificate: An X509.v3 certificate of the peer. This element of the state may be null.
- Compression method: The algorithm used to compress data prior to encryption.
- CipherSpec: Specifies the bulk data encryption algorithm (such as DES) and a hash algorithm (such as MD5 or SHA-1). It also defines cryptographic attributes such as the hash size.
- Master secret: 48-byte secret shared between the client and server.
- Is resumable: A flag indicating whether the session can be used to initiate new connections.

A connection state is defined by the following parameters:

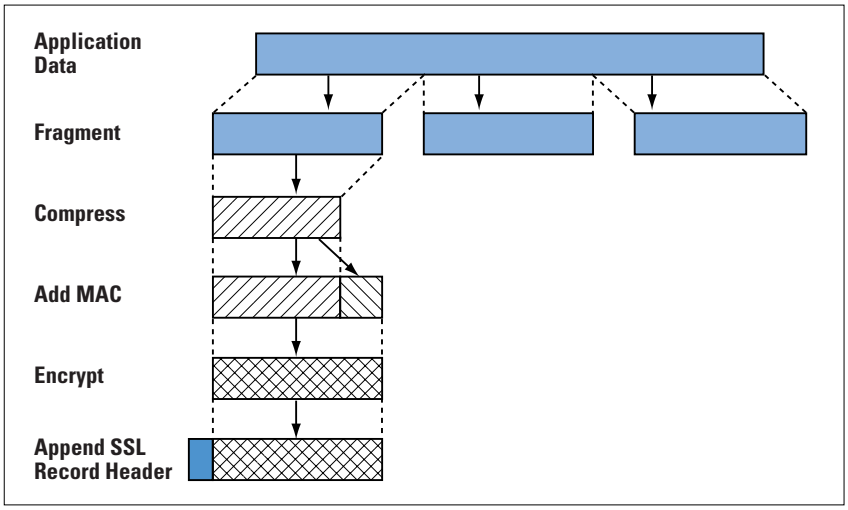
- Server and client random: Byte sequences that are chosen by the server and client for each connection.
- Server write MAC secret: The secret key used in MAC operations on data sent by the server.
- Client write MAC secret: The secret key used in MAC operations on data sent by the client.
- Server write key: The conventional encryption key for data encrypted by the server and decrypted by the client.
- Client write key: The conventional encryption key for data encrypted by the client and decrypted by the server.
- Initialization vectors: When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key. This field is first initialized by the SSL Handshake Protocol. Thereafter the final ciphertext block from each record is preserved for use as the IV for the next record.
- Sequence numbers: Each party maintains separate sequence numbers for transmitted and received messages for each connection. When a party sends or receives a change CipherSpec message, the appropriate sequence number is set to zero.

SSL Record Protocol

The SSL Record Protocol provides two services for SSL connections: confidentiality, by encrypting application data; and message integrity, by using a *message authentication code* (MAC). The Record Protocol is a base protocol that can be utilized by some of the upper-layer protocols of SSL. One of these is the handshake protocol which, as described later, is used to exchange the encryption and authentication keys. It is vital that this key exchange be invisible to anyone who may be watching this session.

Figure 1 indicates the overall operation of the SSL Record Protocol. The Record Protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment. Received data is decrypted, verified, decompressed, and reassembled and then delivered to the calling application, such as the browser.

Figure 1: SSL Record Protocol Operation



The first step is fragmentation. Each upper-layer message is fragmented into blocks of 2^{14} bytes (16,384 bytes) or less. Next, compression is optionally applied. In SLLv3 (as well as the current version of TLS), no compression algorithm is specified, so the default compression algorithm is null. However, specific implementations may include a compression algorithm.

The next step in processing is to compute a message authentication code over the compressed data. For this purpose, a shared secret key is used. In essence, the hash code (for example, MD5) is calculated over a combination of the message, a secret key, and some padding. The receiver performs the same calculation and compares the incoming MAC value with the value it computes. If the two values match, the receiver is assured that the message has not been altered in transit. An attacker would not be able to alter both the message and the MAC, because the attacker does not know the secret key needed to generate the MAC.

Next, the compressed message plus the MAC are encrypted using symmetric encryption. A variety of encryption algorithms may be used, including the Data Encryption Standard (DES) and triple DES.

The final step of SSL Record Protocol processing is to prepend a header, consisting of the following fields:

- Content Type (8 bits): The higher-layer protocol used to process the enclosed fragment.
- Major Version (8 bits): Indicates major version of SSL in use. For SSLv3, the value is 3.
- Minor Version (8 bits): Indicates minor version in use. For SSLv3, the value is 0.
- Compressed Length (16 bits): The length in bytes of the plain-text fragment (or compressed fragment if compression is used).

The content types that have been defined are change_cipher_spec, alert, handshake, and application_data. The first three are the SSL-specific protocols, mentioned previously. The application-data type refers to the payload from any application that would normally use TCP but is now using SSL, which in turn uses TCP. In particular, the HTTP protocol that is used for Web transactions falls into the application-data category. A message from HTTP is passed down to SSL, which then wraps this message into an SSL record.

Change CipherSpec Protocol

The Change CipherSpec Protocol is one of the three SSL-specific protocols that use the SSL Record Protocol, and it is the simplest. This protocol consists of a single message, which consists of a single byte with the value 1. The sole purpose of this message is to cause the pending state to be copied into the current state, which updates the CipherSuite to be used on this connection. This signal is used as a coordination signal. The client must send it to the server and the server must send it to the client. After each side has received it, all of the following messages are sent using the agreed-upon ciphers and keys.

Alert Protocol

The Alert Protocol is used to convey SSL-related alerts to the peer entity. As with other applications that use SSL, alert messages are compressed and encrypted, as specified by the current state.

Each message in this protocol consists of two bytes. The first byte takes the value "warning" (1) or "fatal"(2) to convey the severity of the message. If the level is fatal, SSL immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established. The second byte contains a code that indicates the specific alert. An

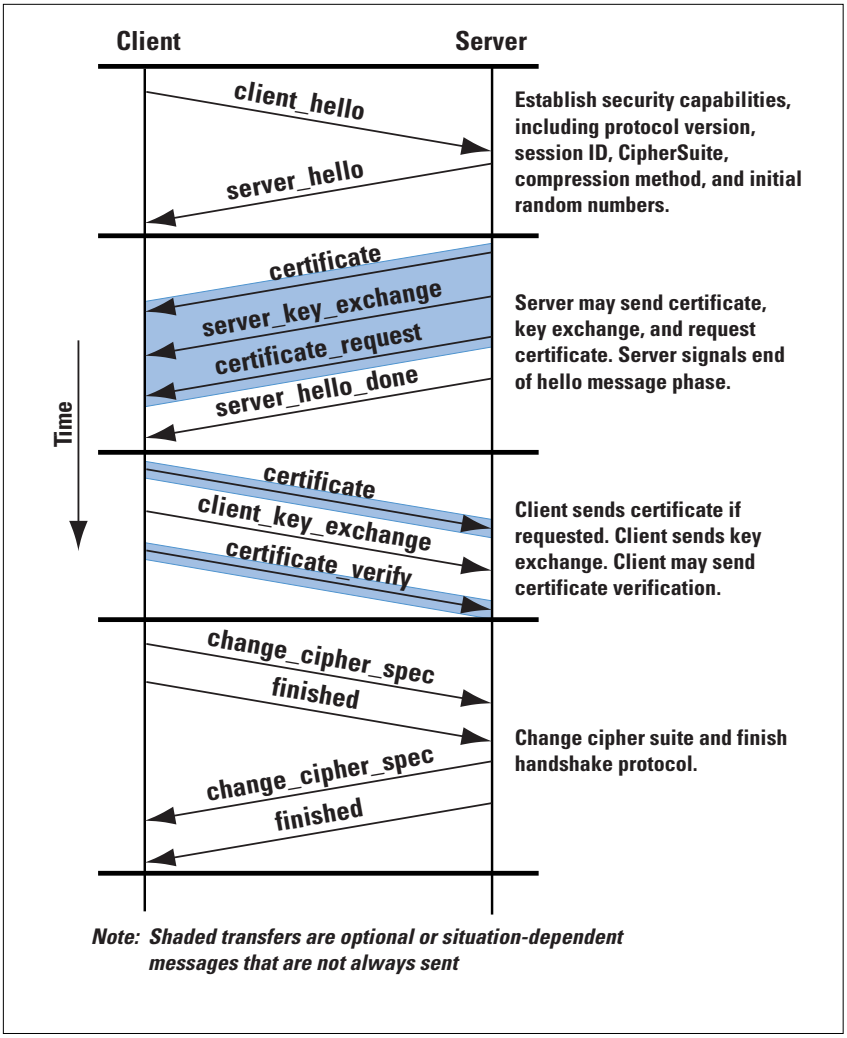
example of a fatal message is illegal_parameter (a field in a handshake message was out of range or inconsistent with other fields). An example of a warning message is close_notify (notifies the recipient that the sender will not send any more messages on this connection; each party is required to send a close_notify alert before closing the write side of a connection).

Handshake Protocol

The most complex part of SSL is the Handshake Protocol. This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record. The Handshake Protocol is used before any application data is transmitted. The Handshake Protocol consists of a series of messages exchanged by the client and the server.

Figure 2 shows the initial exchange needed to establish a logical connection between the client and the server. The exchange can be viewed as having four phases.

Figure 2: Handshake Protocol Action



Phase 1 is used to initiate a logical connection and to establish the security capabilities that will be associated with it. The exchange is initiated by the client, which sends a client_hello message with the following parameters:

- Version: The highest SSL version understood by the client.
- Random: A client-generated random structure, consisting of a 32-bit timestamp and 28 bytes generated by a secure random number generator. These values serve as nonces and are used during key exchange to prevent replay attacks.
- Session ID: A variable-length session identifier. A nonzero value indicates that the client wishes to update the parameters of an existing connection or create a new connection on this session. A zero value indicates that the client wishes to establish a new connection on a new session.
- CipherSuite: A list that contains the combinations of cryptographic algorithms supported by the client, in decreasing order of preference. Each element of the list (each CipherSuite) defines both a key exchange algorithm and a CipherSpec; these are discussed subsequently.
- Compression Method: A list of the compression methods the client supports.

After sending the client_hello message, the client waits for the server_hello message, which contains the same parameters as the client_hello message. For the server_hello message, the following conventions apply. The Version field contains the lower of the version suggested by the client and the highest version supported by the server. The Random field is generated by the server and is independent of the client's Random field. If the SessionID field of the client was nonzero, the same value is used by the server; otherwise the server's SessionID field contains the value for a new session. The CipherSuite field contains the single CipherSuite selected by the server from those proposed by the client. The Compression field contains the compression method selected by the server from those proposed by the client.

The first element of the CipherSuite parameter is the key exchange method (that is, the means by which the cryptographic keys for conventional encryption and MAC are exchanged). The following key exchange methods are supported:

- RSA: The secret key is encrypted with the receiver's RSA public key. A public-key certificate for the receiver's key must be made available.
- Fixed Diffie-Hellman: This a Diffie-Hellman key exchange in which the server's certificate contains the Diffie-Hellman public parameters

signed by the *certificate authority* (CA). That is, the public-key certificate contains the Diffie-Hellman public-key parameters. The client provides its Diffie-Hellman public key parameters either in a certificate, if client authentication is required, or in a key exchange message. This method results in a fixed secret key between two peers, based on the Diffie-Hellman calculation using the fixed public keys.

- Ephemeral Diffie-Hellman: This technique is used to create ephemeral (temporary, one-time) secret keys. In this case, the Diffie-Hellman public keys are exchanged, and signed using the sender's private RSA or DSS key. The receiver can use the corresponding public key to verify the signature. Certificates are used to authenticate the public keys. This option appears to be the most secure of the three Diffie-Hellman options because it results in a temporary, authenticated key.

- Anonymous Diffie-Hellman: The base Diffie-Hellman algorithm is used, with no authentication. That is, each side sends its public Diffie-Hellman parameters to the other, with no authentication. This approach is vulnerable to man-in-the-middle attacks, in which the attacker conducts anonymous Diffie-Hellman exchanges with both parties.

Following the definition of a key exchange method is the CipherSpec, which indicates the encryption and hash algorithms and other related parameters.

The server begins Phase 2 by sending its certificate, if it needs to be authenticated; the message contains one or a chain of X.509 certificates. The certificate message is required for any agreed-on key exchange method except anonymous Diffie-Hellman. Note that if fixed Diffie-Hellman is used, this certificate message functions as the server's key exchange message because it contains the server's public Diffie-Hellman parameters.

Next, a server_key_exchange message may be sent, if it is required. It is not required in two instances: (1) The server has sent a certificate with fixed Diffie-Hellman parameters; or (2) RSA key exchange is to be used.

Next, a nonanonymous server (server not using anonymous Diffie-Hellman) can request a certicate from the client. The certificate_request message includes two parameters: certificate_type and certificate_authorities. The certificate type indicates the type of public-key algorithm. The second parameter in the certificate_request message is a list of the distinguished names of acceptable certificate authorities.

The final message in Phase 2, and one that is always required, is the server_done message, which is sent by the server to indicate the end of the server hello and associated messages. After sending this message, the server waits for a client response. This message has no parameters.

Upon receipt of the server_done message, the client should verify that the server provided a valid certificate, if required, and check that the server hello parameters are acceptable. If all is satisfactory, the client sends one or more messages back to the server in Phase 3. If the server has requested a certificate, the client begins this phase by sending a certificate message. If no suitable certificate is available, the client sends a no_certificate alert instead.

Next is the client_key_exchange message, which must be sent in this phase. The content of the message depends on the type of key exchange.

Finally, in this phase, the client may send a certificate_verify message to provide explicit verification of a client certificate. This message is only sent following any client certificate that has signing capability (that is, all certificates except those containing fixed Diffie-Hellman parameters).

Phase 4 completes the setting up of a secure connection. The client sends a change_cipher_spec message and copies the pending CipherSpec into the current CipherSpec. Note that this message is not considered part of the Handshake Protocol but is sent using the Change CipherSpec Protocol. The client then immediately sends the finished message under the new algorithms, keys, and secrets. The finished message verifies that the key exchange and authentication processes were successful.

In response to these two messages, the server sends its own change_cipher_spec message, transfers the pending to the current CipherSpec, and sends its finished message. At this point the handshake is complete and the client and server may begin to exchange application layer data.

After the records have been transferred, the TCP session is closed. However, since there is no direct link between TCP and SSL, the state of SSL may be maintained. For further communications between the client and the server, many of the negotiated parameters are retained. This may occur if, in the case of Web traffic, the user clicks on another link that also specifies HTTPs on the same server. If the clients or servers wish to resume the transfer of records, they don't have to again negotiate encryption algorithms or totally new keys. The SSL specifications suggest that the state information be cached for no longer than 24 hours. If no sessions are resumed within that time, all information is deleted and any new sessions have to go through the handshake again. The specifications also recommend that neither the client nor the server have to retain this information, and shouldn't if either of them suspects that the encryption keys have been compromised. If either the client or the server does not agree to resume the session, for any reason, then both will have to go through the full handshake.

Transport Layer Security

TLS is an IETF standardization initiative whose goal is to produce an Internet standard version of SSL. In fact, the charter for the TLS working group states:

> "The TLS working group is a focused effort on providing security features at the transport layer, rather than general purpose security and key management mechanisms. The standard track protocol specification will provide methods for implementing privacy, authentication, and integrity above the transport layer."

This means that TLS can be used to provide security services to any application that uses TCP or the *User Datagram Protocol* (UDP). However, the driving force behind this work is to develop a standardized version of SSL. Microsoft has indicated that TLS will go into the next major version of its browser and Web server products, and Netscape has made a similar commitment. With this kind of support, it is likely that TLS will move quickly along the Internet Standards track.

The current draft version of TLS is very similar to SSLv3. TLS uses slightly different cryptographic algorithms for such things as the MAC function generation of secret keys. TLS also includes more alert codes.

SSL is already widely deployed and, under the name TLS, is moving toward Internet standardization. It is the solution of choice for Web transaction security.

References

[1] `http://www.phaos.com/sslresource.html`
(has links to vendors, SSL specifications, and FAQs)

[2] `http://www.netscape.com/newsref/std/SSL.html`
(PostScript versions of the spec are available there)

[3] `http://www.ietf.org/html.charters/tls-charter.html`
(contains latest RFCs and Internet Drafts for TLS)

[4] `http://www.imc.org/ietf-tls/mail-archive/`
(mailing list archive)

[5] `ftp://ftp.ietf.org/internet-drafts/draft-ietf-tls-protocol-05.txt`

[6] `ftp://ftp.ietf.org/internet-drafts/draft-ietf-tls-https-01.txt`

[7] `http://www.consensus.com/security/ssl-talk-faq.html`

WILLIAM STALLINGS is a consultant, lecturer, and author of over a dozen books on data communications and computer networking. He has a PhD in computer science from M.I.T. This article is based on material in the author's latest book, *Cryptography and Network Security, Second Edition* (Prentice-Hall, 1998). His home in cyberspace is `http://www.shore.net/~ws` and he can be reached at `ws@shore.net`

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal will carry tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It will provide readers with technology and standardization updates for all levels of the protocol stack and serve as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and quality of service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

Book Reviews

Groupware

Groupware: Collaborative Strategies for Corporate LANs and Intranets, by David Coleman, ISBN 0-13-727728-8, Prentice-Hall PTR, 1997, `http://www.prenhall.com`.

Some areas of science provide very poor training for dealing with primarily human processes. One might think that packet switching would be an exception because it lives on the stochastic nature of bursty communications. Because our knowledge of human and group activity is, at best, characterized by statistical assessments, those working in networking should do well in understanding and dealing with the unpredictable and human nature of communication, especially when it involves using networks.

So much for theory. In general, the world of lower-level networking has done little for the upper strata of computer-mediated human communication, except to provide a platform for the work of others. An apparent exception in the world of Internet technology is e-mail, yet it actually serves more as proof of the problem than as an exception. The basic facilities in Internet e-mail are the same today as they were 25 years ago. As nice as they are, the word "basic" is essential when characterizing them. Almost none of the Internet's standardized e-mail facilities are really targeted at providing automated or structural support for the work of a group.

Groupware Defined

The collection of products and services designed to help people collaborate via computer, by direct interaction, or by information dissemination is called "groupware." Coleman's book is a revision of *Groupware: Technology and Applications*. Written only 15 months earlier, the world changed more than enough in that time to require the revision. The first book had relatively little to say about the Internet, whereas this new book tries mightily to factor it into the equation. The result is a bit erratic, but the digressions serve to highlight how rapidly things are changing, rather than to suggest looking elsewhere for a better source on the topic.

The new book has an entirely different subtitle, giving a reasonable sense that the content targets more an understanding of system organization and function than detailed technical explanation. That's just fine, because the book really is not particularly technical. It covers the requirements and functions for supporting activity by groups.

Downsizing and working remotely are two very strong driving forces for increased use of groupware. This book is essentially an introduction to concepts, functionality, and use of systems that attempt to help staff members work together. Oddly, that does not only mean working together when physically separated, because there is discussion of meeting room assistance, such as with automated sense-of-the-group tallying devices.

Organization

The first two chapters introduce the topic, emphasizing that human and group process concerns dominate the field and are intimately tied to the aggressive efforts that organizations are making to run more productively and, frequently, with fewer people. The third chapter discusses functionality in terms of the World Wide Web. The book reflects the current enthusiasm for the Web, sometimes to the detriment of the appropriate use of messaging technology, although messaging is more prevalent among groupware than other kinds of commercial Internet systems.

The realm of groupware does not have a firm taxonomy. My own synthesis includes: Message (text and document) Exchange, Forms Exchange, Calendaring & Scheduling, Workflow, Presentations and Interactive Meetings, and Document Development and Sharing. The next six chapters cover the functional pieces of this groupware realm.

The next five chapters cover the major vendors of integrated groupware products: Lotus Notes, Novel GroupWise, TeamWARE, Hewlett-Packard, and Oracle Interoffice. HP's chapter discusses "strategy," suggesting the lack of a well-integrated product suite, but one more survey of the terrain is nonetheless useful. And that, perhaps, is the major reason for reading this book: It constantly emphasizes the human and process-oriented aspect of organizational behavior and the need to attend carefully both to the needs of the humans and the nature of the processes. It is easy to understand that an improper travel authorization, will bring an organization to its knees. It is easy to forget that the system is used by humans who well might not want the added complexity or rigidity of the system and who, therefore, must be part of the design and adoption effort. In my opinion, the book takes a rather more negative view about groupware acceptability than is necessary, but then I like such technology, and the average worker in the average organization does not.

The last six chapters of this book intermix case studies and Hahn, of Collabra and Netscape, points the reader to Chapter 17, "Groupware & Reengineering: The Human Side of Change." Although one of the better considerations of these issues in the book, it is far from the only one.

A Useful Survey

If you have little familiarity with these "upper level application" areas of networking, the functionality, products, or use, then this book is a good one to read. You will not learn much about the underlying technology, nor will you be able to qualify as a "certified groupware support engineer," but you will obtain an extremely useful survey of the field, and you will obtain it from the perspective of human and organization use. As the Internet moves into the mass market, that perspective is a good one.

—Dave Crocker
Brandenburg Consulting
dcrocker@brandenburg.com

High-Speed Networks

High-Speed Networks: TCP/IP and ATM Design Principles, by William Stallings, ISBN 0-13-525965-7 Prentice-Hall, 1997, `http://www.shore.net/~ws/HsNet.html`

High-speed networks now dominate both the WAN and LAN markets. In the WAN market, data networks have evolved from packet-switching networks to ATM networks operating at 155 Mbps or more. In the LAN market, the staple 10-Mbps Ethernet is being replaced with 100-Mbps Fast Ethernet, Gigabit Ethernet, and even Asynchronous Transfer Mode (ATM) LANs. This book provides a survey of high-speed networks and the design issues related to them. Much of the book is devoted to the study of various techniques aimed at reducing network congestion.

Organization

The book is divided into seven sections. The first section deals with the fundamentals: TCP/IP principles; packet switching and Frame Relay networks; and internetworking principles. The second section provides an overview of ATM and Fast and Gigabit Ethernet. These two sections can easily be torn out of the book and serve as an excellent primer on today's modern networks. I am going to recommend to my employer that they be made mandatory reading.

In the third section of the book, Stallings focuses on one treatment of queueing theory, namely, how it is applied to modeling network behavior. Stallings has an undeniable gift for taking large complicated subjects and teaching the fundamentals, and then some, without belittling the subject at hand or the reader. This book is witness to this gift, and this chapter but one fine example. But once the reader has an understanding of queueing theory, Stallings throws a wrench in the gears. The chapter on self-similarity explains why traditional queuing models are inadequate when trying to predict the performance of Ethernet traffic and other self-similar streams. While this section is by far the most theoretical, it is at the same time necessary for the reader's understanding of network performance, and while many readers may not care to devote the time necessary to gain a complete understanding of self-similarity, astute students are urged to invest in more than a simple gloss-over of this section.

Having understood the basics of self-similarity, I hoped the fifth section of the book, on network traffic management, would be addressed with greater emphasis on delivering quality of service and the problems related to self-similarity. Instead, the material is based on traditional queueing models.

The fourth section, flow control, is divided into two categories. The first, link control mechanisms, focuses on some of the performance issues related to the use of *Automatic Repeat Request* (ARQ) link control protocols. The second category, transport control mechanisms,

concentrates on the TCP flow control mechanism. I expected to find references to bugs in some TCP implementations exposed by high-volume WWW servers, but didn't. Stallings goes on to present an overview of some of the performance issues of TCP over ATM. As institutions begin upgrading their networks, this issue is sure to receive a great deal of interest. The section concludes with a look at the *Real-Time Transport Protocol*, another area sure to spark attention as the need to move large multimedia data across WANs, in real time, becomes more relevant.

The sixth section of the book covers Internet routing protocols and opens with a primer on graph theory. Four routing protocols (RIP, OSPF, BGP, and IDRP) are covered. The section concludes with a discussion of multicasting as an introduction to RSVP. This section sparked my curiosity enough to call for a visit to the WWW site for RSVP development.

Stallings shies away from directly addressing application-driven improvements aimed at increasing network performance. In today's Web/CGI-driven world, I would expect this to be a topic of interest to many. Perhaps this is a subject for another book. But the topic is not entirely avoided. The last section of the book focuses on various lossless and lossy compression techniques. The quirkiness of material covered makes this section a darling.

Recommended

This book rates an A+. Unlike most books about computers being published today, this book is neither superficial nor is it insulting to the reader. It is intended for both professional and academic audiences. Stallings' desire to truly educate is apparent. This is not a book about promoting the hype, this is a book about serious learning.

—Neophytos Iacovou,
University of Minnesota
Academic & Distributed Computing Services
iacovou@boombox.micro.umn.edu

Fragments

The Fragments page is intended to provide you with updates and pointers to information related to Internet technology developments.

The Future of the Domain Name System (DNS)

For more than a year, a debate has taken place regarding the future of the DNS. In particular, the issue of competitive name registries, possible addition of new *global Top Level Domains* (gTLDs) and the future of the *Internet Assigned Numbers Authority* (IANA) have been discussed. Information regarding the initial proposal can be found at: `http://www.gtld-mou.org/`. The US Government has issued a so-called *Green Paper* entitled "Technical Management of Internet Names and Addresses." The Green Paper and comments received on this document can be found at:
`http://www.ntia.doc.gov/ntiahome/domainname/`

IETF and Related Links

The *Internet Engineering Task Force* (IETF) is responsible for the development of standards for Internet technology. Membership to the IETF is open and you can participate in person or subscribe to the IETF mailing list. The IETF meets three times per year. For a list of future meetings and other IETF information see: `http://www.ietf.org`. On this website you will also find a number of links to organizations which are related to the IETF in one way or another:

- *The Internet Society* (ISOC) and its annual INET conference.
- *The Internet Architecture Board* (IAB)
- *The Internet Assigned Numbers Authority* (IANA)
- *The Internet Research Task Force* (IRTF)

SIGCOMM

If you want to learn about the latest developments on the research side of networking you should check out SIGCOMM, the Association for Computing Machinery's Special Interest Group on Communications. You can find out more about the group and their annual conference at:
`http://www.acm.org/sigcomm/sigcomm98`

Send Us Your Comments!

We look forward to hearing your comments and suggestions regarding anything you read in this publication. Send e-mail to: `ipj@cisco.com`.

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Cisco News Publications Group, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-J4
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

September 1998 Volume 1, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

Missed the first issue of IPJ? Download your copy in PDF format from: www.cisco.com/ipj

From The Editor

We begin this issue with Part II of "What Is a VPN?" by Paul Ferguson and Geoff Huston. In Part I they introduced a definition of the term "Virtual Private Network" (VPN) and discussed the motivations behind the adoption of such networks. They outlined a framework for describing the various forms of VPNs, and examined numerous network-layer VPN structures, in particular, that of controlled route leakage and tunneling. In Part II the authors conclude their examination of VPNs by describing virtual private dial networks and network-layer encryption. They also examine link-layer VPNs, switching and encryption techniques, and issues concerning Quality of Service and non-IP VPNs.

IP Multicast is an emerging set of technologies and standards that allow many-to-many transmissions such as conferencing, or one-to-many transmissions such as live broadcasts of audio and video over the Internet. Kenneth Miller describes multicast in general, and reliable multicast protocols and applications in particular. Although multicast applications are primarily used in the research community today, this situation is likely to change as the demand for Internet multimedia applications increases and multicast technologies improve.

Successful deployment of networking technologies requires an understanding of a number of technology options ranging from wiring and transmissions systems via switches, routers, bridges and other pure networking components, to networked applications and services. *The Internet Protocol Journal* (IPJ) is designed to look at all aspects of these "building blocks." This time, Thayumanavan Sridhar details some of the issues in the evolution of Layer 2 and Layer 3 switches.

Interest in the first issue of IPJ has exceeded our expectations, and hard copies are almost gone. However, you can still view and print the issue in PDF format on our Web site at **www.cisco.com/ipj**. The current edition is also available on the Web. If you want to receive our next issue, please complete and return the enclosed card.

We welcome your comments, questions and suggestions regarding anything you read in this journal. We are also actively seeking authors for new articles. The Call for Papers and Author Guidelines can be found on our Web page. Please send your comments to **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher

ole@cisco.com

What Is a VPN? — Part II

by Paul Ferguson, Cisco Systems
and Geoff Huston, Telstra

In Part I we introduced a working definition of the term "Virtual Private Network" (VPN), and discussed the motivations behind the adoption of such networks. We outlined a framework for describing the various forms of VPNs, and then examined numerous network-layer VPN structures, in particular, that of controlled route leakage and tunneling techniques. We begin Part II with examining other network-layer VPN techniques, and then look at issues that are concerned with non-IP VPNs and Quality-of-Service (QoS) considerations.

Types of VPNs

This section continues from Part I to look at the various types of VPNs using a taxonomy derived from the layered network architecture model. These types of VPNs segregate the VPN network at the network layer.

Network-Layer VPNs

A network can be segmented at the network layer to create an end-to-end VPN in numerous ways. In Part I we described a controlled route leakage approach that attempts to perform the segregation only at the edge of the network, using route advertisement control to ensure that each connected network received a view of the network (only peer networks). We pick up the description at this point in this second part of the article.

Tunneling

As outlined in Part I, the alternative to a model of segregation at the edge is to attempt segregation throughout the network, maintaining the integrity of the partitioning of the substrate network into VPN components through the network on a hop-by-hop basis. Part I examined numerous tunneling technologies that can achieve this functionality. Tunneling is also useful in servicing VPN requirements for dial access, and we will resume the description of tunnel-based VPNs at this point.

Virtual Private Dial Networks

Although several technologies (vendor-proprietary technologies as well as open, standards-based technologies) are available for constructing a *Virtual Private Dial Network* (VPDN), there are two principal methods of implementing a VPDN that appear to be increasing in popularity—*Layer 2 Tunneling Protocol* (L2TP) and *Point-to-Point Tunneling Protocol* (PPTP) tunnels. From an historical perspective, L2TP is the technical convergence of the earlier Layer 2 Forwarding (L2F)[1] protocol specification and the PPTP protocol. However, one might suggest that because PPTP is now being bundled into the desktop operating system of many of the world's personal computers, it stands to be quite popular within the market.

At this point it is worthwhile to distinguish the difference between "client-initiated" tunnels and "NAS-initiated" (Network Access Server, otherwise known as a Dial Access Server) tunnels. The former is commonly referred to as "voluntary" tunneling, whereas the latter is commonly referred to as "compulsory" tunneling. In voluntary tunneling, the tunnel is created at the request of the user for a specific purpose; in compulsory tunneling, the tunnel is created without any action from the user, and without allowing the user any choice in the matter.

L2TP, as a compulsory tunneling model, is essentially a mechanism to "off-load" a dialup subscriber to another point in the network, or to another network altogether. In this scenario, a subscriber dials into a NAS, and based on a locally configured profile (or a NAS negotiation with a policy server) and successful authentication, a L2TP tunnel is dynamically established to a predetermined endpoint, where the subscriber's *Point-to-Point Protocol* (PPP) session is terminated (Figure 1).

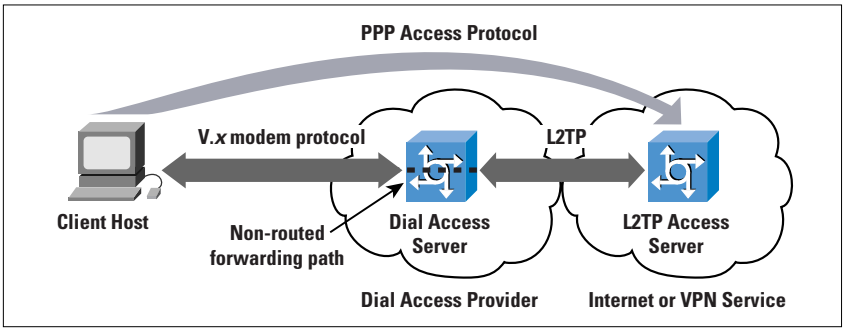


Figure 1: PPP Tunnel Termination Model of L2TP

PPTP, as a voluntary tunneling model, on the other hand, allows end systems (for example, desktop computers) to configure and establish individual discrete point-to-point tunnels to arbitrarily located PPTP servers, without the intermediate NAS participating in the PPTP negotiation and subsequent tunnel establishment. In this scenario, a subscriber dials into a NAS, but the PPP session is terminated on the NAS, as in the traditional Internet access PPP model. The layered PPTP session is then established between the client end system and any upstream PPTP server that the client desires to connect to. The only caveats on PPTP connectivity are that the client can reach the PPTP server via conventional routing processes, and that the user has been granted the appropriate privileges on the PPTP server (Figure 2).

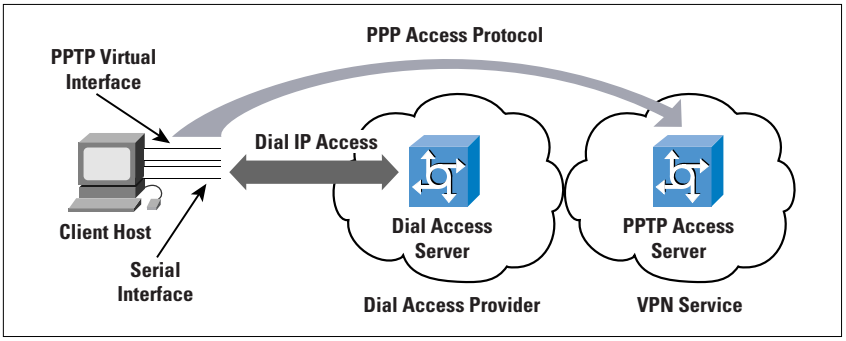


Figure 2: PPP Tunnel Termination Model of PPTP

Although L2TP and PPTP may sound extraordinarily similar, there are subtle differences that deserve further examination. The applicability of both protocols is very much dependent on what problem is being addressed. It is also about control—who has it, and why it is needed. It also depends heavily on how each protocol implementation is deployed—in either the voluntary or the compulsory tunneling models.

With PPTP in a voluntary tunneling implementation, the dial-in user can choose the PPTP tunnel destination (the PPTP server) after the initial PPP negotiation has completed. This feature is important if the tunnel destination changes frequently, because no modifications are needed to the client's view of the base PPP access when there is a change in the server and the transit path to the server. It is also a significant advantage that the PPTP tunnels are transparent to the service provider, and no advance configuration is required between the NAS operator and the overlay dial access VPN. In such a case, the service provider does not house the PPTP server, and simply passes the PPTP traffic along with the same processing and forwarding policies as all other IP traffic. In fact, this feature should be considered a significant benefit of this approach. The configuration and support of a tunneling mechanism within the service provider network would be one less parameter that the service provider has to operationally manage, and the PPTP tunnel can transparently span multiple service providers without any explicit service provider configuration. However, the economic downside to this feature for the service provider, of course, is that a "VPDN-enabled" network service can be marketed to yield an additional source of revenue. Where the client undertakes the VPDN connection, there is no direct service provider involvement and no consequent value added to the base access service.

From the subscriber's perspective, this is a "win-win" situation, because the user is not reliant on the upstream service provider to deliver the VPDN service—at least no more than any user is reliant for basic IP-level connectivity. The other "win" is that the subscriber does not have to pay a higher subscription fee for a VPN service. Of course, the situation changes when the service provider takes an active role in providing the VPDN, such as housing the PPTP servers, or if the subscriber resides within a subnetwork in which the parent organization wants the service provider's network to make the decision concerning where tunnels are terminated. The major characterization of PPTP-based VPDN is one of a roaming client base, where the clients of the VPDN use a local connection to the public Internet data network, and then overlay a private data tunnel from the client's system to the desired remote service point. Another perspective is to view this approach as "on-demand" VPDN virtual circuits.

With L2TP in a "compulsory" tunneling implementation, the service provider controls where the PPP session is terminated. This setup can be extremely important in situations where the service provider to whom

the subscriber is actually dialing into (let's call it the "modem pool provider" network) must transparently hand off the subscriber's PPP session to another network (let's call this network the "content provider"). To the subscriber, it appears as though the local system is directly attached to the content provider's network, when in fact the access path has been passed transparently through the modem pool provider's network to the subscribed content service. Very large content providers, for instance, may outsource the provisioning and maintenance of thousands of modem ports to a third-party access provider, who in turn agrees to transparently pass the subscribers' access sessions back to the content provider. This setup is generally called "wholesale dial." The major motivation for such L2TP-based wholesale dial lies in the typical architecture of the *Public Switched Telephone Network* (PSTN), where the use of wholesale dial facilities can create a more rational PSTN call load pattern with Internet access PSTN calls terminated in the local Central Office.

Of course, if all subscribers who connect to the modem pool provider's network are destined for the same content provider, then there are certainly easier ways to hand this traffic off to the content provider's network—such as simply aggregating all the traffic in the local Central Office and handing the content provider a "big fat pipe" of the aggregated session traffic streams. However, in situations where the modem pool provider is providing a wholesale dial service for multiple upstream "next-hop" networks, the methods of determining how each subscriber's traffic must be forwarded to his/her respective content provider are somewhat limited. Packet forwarding decisions could be made at the NAS, based on the source address of the dialup subscriber's computer. This scenario would allow for traffic to be forwarded along the appropriate path to its ultimate destination, in turn intrinsically providing a virtual connection. However, the use of assigning static IP addresses to dial-in subscribers is highly discouraged because of the inefficiencies in IP address utilization policies, and the critical success of the *Dynamic Host Configuration Protocol* (DHCP).

There are, however, some serious scaling concerns in deploying a large-scale L2TP network; these concerns revolve around the issue of whether large numbers of tunnels can actually be supported with little or no network performance impact. Since there have been no large-scale deployments of this technology to date, there is no empirical evidence to support or invalidate these concerns.

In some cases, however, appearances are everything—some content providers do not wish for their subscribers to know that when they connect to their service, they have instead been connected to another service provider's network, and then passed along ultimately to the service to which they have subscribed. In other cases, it is merely designed to be a matter of convenience, so that subscribers do not need to log into a device more than once.

Regrettably, the L2TP draft does not detail all possible implementations or deployment scenarios for the protocol. The basic deployment scenario is quite brief when compared to the rest of the document, and is arguably biased toward the compulsory tunneling model. Nonetheless, there are implementations of L2TP that follow the voluntary tunneling model. To the best of our knowledge, there has never been any intent to exclude this model of operation. In addition, at various recent interoperability workshops, several different implementations of a voluntary L2TP client have been modeled. Nothing in the L2F protocol would prohibit deploying it in a voluntary tunneling manner, but to date it has not been widely implemented. Further, PPTP has also been deployed using the compulsory model in a couple of specific vendor implementations.

In summary, consideration of whether PPTP or L2TP is more appropriate for deployment in a VPDN depends on whether control needs to lie with the service provider or with the subscriber. Indeed, the difference can be characterized with respect to the client of the VPN, where the L2TP model is one of a "wholesale" access provider who has numerous configured client service providers who appear as VPNs on the common dial access system, whereas the PPTP model is one of distributed private access where the client is an individual end user and the VPN structure is that of end-to-end tunnels. One might also suggest that the difference is also a matter of economics, because the L2TP model allows service providers to actually provide a "value-added" service, beyond basic IP-level connectivity, and charge their subscribers accordingly for the ability to access it, thus creating new revenue streams. By contrast, the PPTP model enables distributed reach of the VPN at a much more basic level, enabling corporate VPNs to extend access capabilities without the need for explicit service contracts with a multitude of network access providers.

Network-Layer Encryption

Encryption technologies are extremely effective in providing the segmentation and virtualization required for VPN connectivity, and they can be deployed at almost any layer of the protocol stack. The evolving standard for network-layer encryption in the Internet is *IP Security* (IPSec)[3, 4]. (IPSec is actually an architecture—a collection of protocols, authentication, and encryption mechanisms. The IPSec security architecture is described in detail in [3].)

While the *Internet Engineering Task Force* (IETF) is finalizing the architecture and the associated protocols of IPSec, there is relatively little network-layer encryption being done in the Internet today. However, some vendor proprietary solutions are currently in use.

Whereas IPSec has yet to be deployed in any significant volume, it is worthwhile to review the two methods in which network-layer encryption is predominantly implemented. The most secure method for network-

layer encryption to be implemented is end-to-end, between participating hosts. End-to-end encryption allows for the highest level of security. The alternative is more commonly referred to as "tunnel mode," in which the encryption is performed only between intermediate devices (routers), and traffic between the end system and the first-hop router is in plaintext. This setup is considerably less secure, because traffic intercepted in transit between the first-hop router and the end system could be compromised.

As a more general observation on this security vulnerability, where a VPN architecture is based on tunnels, the addition of encryption to the tunnel still leaves the tunnel ingress and egress points vulnerable, because these points are logically part of the host network as well as being part of the unencrypted VPN network. Any corruption of the operation, or interception of traffic in the clear, at these points will compromise the privacy of the private network.

In the end-to-end encryption scheme, VPN granularity is to the individual end-system level. In the tunnel mode scheme, the VPN granularity is to the subnetwork level. Traffic that transits the encrypted links between participating routers, however, is considered secure. Network-layer encryption, to include IPSec, is merely a subset of a VPN.

Link-Layer VPNs

One of the most straightforward methods of constructing VPNs is to use the transmission systems and networking platforms for the physical and link-layer connectivity, yet still be able to build discrete networks at the network layer. A link-layer VPN is intended to be a close (or preferably exact) functional analogy to a conventional private data network.

ATM and Frame Relay Virtual Connections

A conventional private data network uses a combination of dedicated circuits from a public carrier, together with an additional private communications infrastructure, to construct a network that is completely self-contained. Where the private data network exists within private premises, the network generally uses a dedicated private wiring plant to carry the VPN. Where the private data network extends outside the private boundary of the dedicated circuits, it is typically provisioned for a larger public communications infrastructure by using some form of time-division or frequency-division multiplexing to create the dedicated circuit. The essential characteristic of such circuits is the synchronization of the data clock, such that the sender and receiver pass data at a clocking rate that is fixed by the capacity of the dedicated circuit.

A link-layer VPN attempts to maintain the critical elements of this self-contained functionality, while achieving economies of scale and operation, by utilizing a common switched public network infrastructure. Thus, a collection of VPNs may share the same infrastructure for connectivity, and share the same switching elements within the interior of

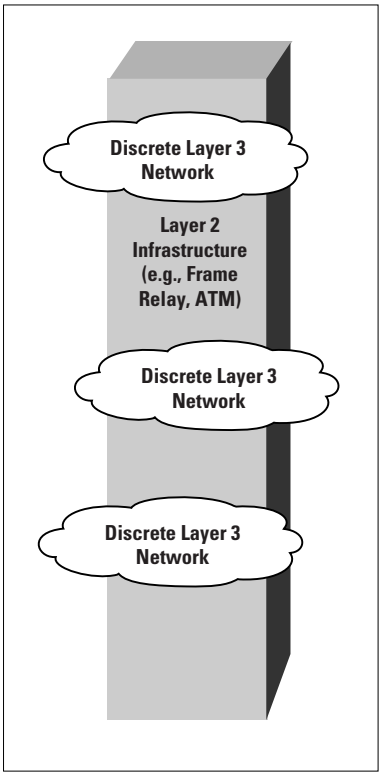


Figure 3: Conceptualization of Discrete Layer 3 Networks on a Common Layer 2 Infrastructure

the network, but explicitly must have no visibility, either direct or inferred, of one another. Generally, these "networks" operate at Layer 3 (the network layer) or higher in the OSI Reference Model, and the "infrastructure" itself commonly consists of either a *Frame Relay* or *Asynchronous Transfer Mode* (ATM) network (Figure 3). The essential difference here between this architecture of virtual circuits and that of dedicated circuits is that there is now no synchronized data clock shared by the sender and receiver, nor necessarily is there a dedicated transmission path that is assigned from the underlying common host network. The sender generally has no a priori knowledge of the available capacity of the virtual circuit, because the capacity varies in response to the total demand placed on it by other simultaneous transmission and switching activity. Instead, the sender and receiver can use adaptive clocking of data, where the sender can adjust the transmission rate to match the requirements of the application and any signaling received from the network and the receiver. It should be noted that a dedicated circuit system using synchronized clocking cannot be oversubscribed, whereas the virtual circuit architecture (where the sender does not have a synchronized end-to-end data clock) can indeed be oversubscribed. It is the behavior of the network when it transitions into this oversubscribed state that is of most interest here.

One of the nice things about a public switched wide-area network that provides virtual circuits is that it can be extraordinarily flexible. Most subscribers to Frame Relay services, for example, have subscribed to the service for economic reasons—it is cheap, and the service provider usually adds a *Service-Level Agreement* (SLA) that "guarantees" some percentage of frame delivery in the Frame Relay network itself.

The remarkable thing about this service offering is that the customer is generally completely unaware of whether the service provider can actually deliver the contracted service at all times and under all possible conditions. The Layer 2 technology is not a synchronized clock blocking technology in which each new service flow is accepted or denied based on the absolute ability to meet the associated resource demands. Each additional service flow is accepted into the network and carried on a best-effort basis. Admission functions provide the network with a simple two-level discard mechanism that allows a graduated response to instances of overload; however, when the point of saturated overload is reached within the network, all services will be affected.

This situation brings up several other important issues: The first concerns the engineering practices of the Frame Relay service provider. If the Frame Relay network is poorly engineered and is constantly congested, then obviously the service quality delivered to the subscribers will be affected. Frame Relay uses a notion of a per-virtual circuit *Committed Information Rate* (CIR), which is an ingress function associated with Frame Relay that checks the ingress traffic rate against the CIR.

Frames that exceed this base rate are still accepted by the Frame Relay network, but they are marked as *discard eligible* (DE). Because the network can be oversubscribed, the data rate within a switch will at times exceed both the egress transmission rate and the local buffer storage. When this situation occurs, the switch will begin to discard data frames, and will do so initially for frames with the DE marker present. This scenario is essentially a two-level discard precedence architecture. It is an administrative decision by the service provider as to the relative levels of provisioning of core transmission and switching capacity, and the ratio of network ingress capacity used by subscribers. The associated CIRs of the virtual circuits against this core capacity are critical determinants of the resultant deliverable quality of performance of the network and the layered VPNs.

For example, at least one successful (and popular) Frame Relay service provider provides an economically attractive Frame Relay service that permits a zero-rate CIR on PVCs, combined with an SLA that ensures that at least 99.8 percent of all frame-level traffic presented to the Frame Relay network will be delivered successfully. If this SLA is not met, then the subscriber's monthly service fee will be appropriately prorated the following month. The Frame Relay service provider provides frame level statistics to each subscriber every month, culled from the Frame Relay switches, to measure the effectiveness of this SLA "guarantee." This particular Frame Relay service provider is remarkably successful in honoring the SLAs because they conduct ongoing network capacity management on a weekly basis, provisioning new trunks between Frame Relay switches when trunk utilization exceeds 50 percent, and ensuring that trunk utilization never exceeds 75 percent. In this fashion, traffic on PVCs with a zero-rate CIR can generally avoid being discarded in the Frame Relay network.

Having said that, the flexibility of PVCs allows discrete VPNs to be constructed across a single Frame Relay network. And in many instances, this scenario lends itself to situations where the Frame Relay network provider also manages each discrete VPN via a telemetry PVC. Several service providers have *Managed Network Services* (MNS) that provide exactly this type of service.

Whereas the previous example revolves around the use of Frame Relay as a link-layer mechanism, essentially the same type of VPN mechanics hold true for ATM. As with Frame Relay, there is no data clock synchronization between the sender, the host network, and the receiver. In addition, the sender's traffic is passed into the ATM network via an ingress function, which can mark cells with a *Cell Loss Priority* (CLP) indication. And, as with Frame Relay, where a switch experiences congestion, the switch will attempt to discard marked (CLP) cells as the primary load shedding mechanism, but if this step is inadequate, the network must shed other cells that are not so marked. Once again, the quality of the service depends on proper capacity engineering of the network, and there is no guarantee of service quality inherently in the technology itself.

The generic observation is that the engineering of Frame Relay and ATM common carriage data networks is typically very conservative. The inherent capabilities of both of these link-layer architectures do not permit a wide set of selective responses to network overload, so that in order for the network to service the broadest spectrum of potential VPN clients, the network must provide high-quality carriage and very limited instances of any form of overload. In this way, such networks are typically positioned as a high-quality alternative to dedicated circuit private network architectures, which are intended to operate in a very similar manner (and, not surprisingly, are generally priced as a premium VPN offering). Technically, the architecture of link-layer VPNs is almost indistinguishable from the dedicated circuit private data network—the network can support multiple protocols, private addressing, and routing schemes, because the essential difference between a dedicated circuit and a virtual link-layer circuit is the absence of synchronized clocking between the sender and the receiver. In all other aspects, the networks are very similar.

These approaches to constructing VPNs certainly involve scaling concerns, especially with regard to configuration management of provisioning new *Virtual Connections* (VCs) and routing issues. Configuration management still tends to be one of the controversial points in VPN management—adding new subscribers and new VPNs to the network requires VC path construction and provisioning, a tedium that requires ongoing administrative attention by the VPN provider. Also, as already mentioned, full mesh networks encounter scaling problems, in turn resulting in construction of VPNs in which partial meshing is done to avoid certain scaling limitations. The liabilities in these cases need to be examined closely, because partial meshing of the underlying link-layer network may contribute to suboptimal routing (for example, extra hops caused by hub-and-spoke issues, or redirects).

These problems apply to all types of VPNs built on the "overlay" model—not just ATM and Frame Relay. Specifically, the problems also apply to *Generic Routing Encapsulation* (GRE) tunnels.

MPOA and the "Virtual Router" Concept

Another unique model of constructing VPNs is the use of *Multiprotocol over ATM* (MPOA)[5], which uses RFC 1483 encapsulation[6]. This VPN approach is similar to other "cut-through" mechanisms in which a particular switched link layer is used to enable all "Layer 3" egress points to be only a single hop away from one another.

In this model, the edge routers determine the forwarding path in the ATM switched network, because they have the ability to determine which egress point packets need to be forwarded to. After a network-layer reachability decision is made, the edge router forwards the packet onto a VC designated for a particular egress router. However, since the egress routers cannot use the *Address Resolution Protocol* (ARP) for destination address across the cloud, they must rely on an external server for address resolution (ATM address to IP address).

The first concern here is a sole reliance on ATM—this particular model does not encompass any other types of data link layer technologies, rendering the technology less than desirable in a hybrid network. Whereas this scenario may have some domain of applicability within a homogenous ATM environment, when looking at a broader VPN environment that may encompass numerous link-layer technologies, this approach offers little benefit to the VPN provider.

Secondly, there are serious scaling concerns regarding full mesh models of connectivity, where suboptimal network-layer routing may result because of cut-through. And the reliance on address resolution servers to support the ARP function within the dynamic circuit framework brings this model to the point of excessive complexity.

The advantage of the MPOA approach is the use of dynamic circuits rather than more cumbersome, statically configured models. The traditional approach to supporting private networks involves extensive manual design and operational support to ensure that the various configurations on each of the bearer switching elements are mutually consistent. The desire within the MPOA environment is to attempt to use MPOA to govern the creation of dynamically controlled, edge-to-edge ATM VCs. Although this setup may offer the carrier operator some advantages in reduced design and operational overhead, it does require the uniform availability of ATM, and in many heterogeneous environments this scenario is not present.

In summary, this model is another overlay model, with some serious concerns regarding the ability of the model to withstand scale.

"Peer" VPN models that allow the egress nodes to maintain separate routing tables have also been introduced—one for each VPN—effectively allowing separate forwarding decisions to be made within each node for each distinctive VPN. Although this is an interesting model, it introduces concerns about approaches in which each edge device runs a separate routing process and maintains a separate *Routing Information Base* (RIB, or routing table) process for each VPN community of interest. It also should be noted that the "virtual router" concept requires some form of packet labeling, either within the header or via some lightweight encapsulation mechanism, in order for the switch to be able to match the packet against the correct VPN routing table. If the label is global, the issue of operational integrity is a relevant concern, whereas if the label is local, the concept of label switching and maintenance of edge-to-edge label switching contexts is also a requirement.

Among the scaling concerns are issues regarding the number of supported VPNs in relation to the computational requirements, and stability of the routing system within each VPN (that is, instability in one VPN affecting the performance of other VPNs served by the same device). The aggregate scaling demands of this model are also significant. Given a change in the underlying physical or link-layer topology, the consequent

requirement to process the routing update on a per-VPN basis becomes a significant challenge. Use of distance vector protocols to manage the routing tables would cause a corresponding sudden surge in traffic load, and the surge grows in direct proportion to the number of supported VPNs. The use of link-state routing protocols would require the consequent link-state calculation to be repeated for each VPN, causing the router to be limited by available CPU capacity.

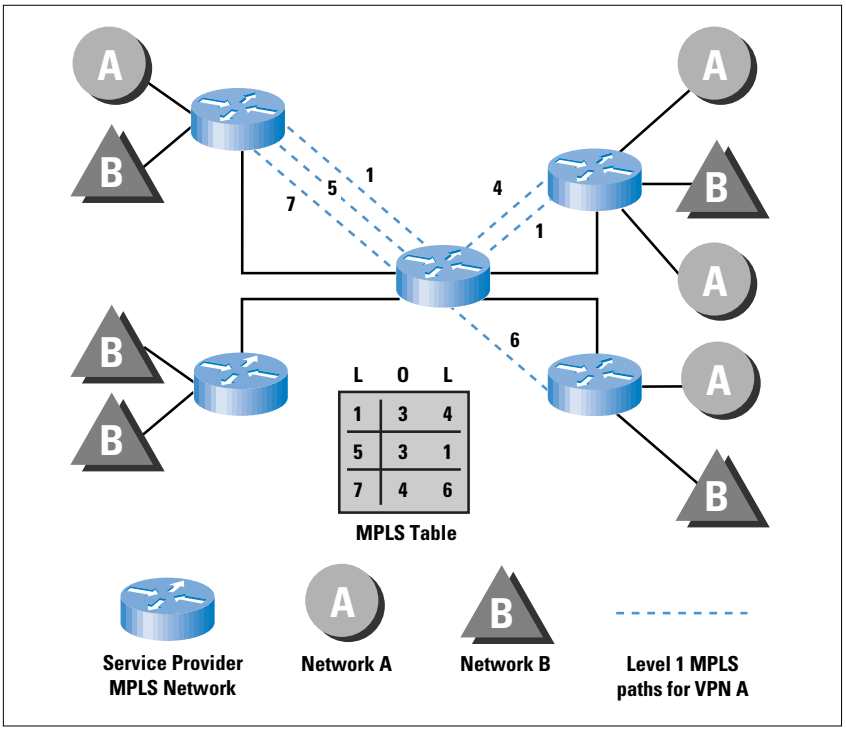
Multiprotocol Label Switching

One method of addressing these scaling issues is to use VPN labels within a single routing environment, in the same way that packet labels are necessary to activate the correct per-VPN routing table. The use of local label switching effectively recreates the architecture of a Multiprotocol Label Switching VPN. It is perhaps no surprise that when presented with two basic approaches to the architecture of the VPN—the use of network-layer routing structures and per-packet switching, and the use of link-layer circuits and per-flow switching—the industry would devise a hybrid architecture that attempts to combine aspects of these two approaches. This hybrid architecture is referred to as *Multiprotocol Label Switching* (MPLS)[7, 8].

The architectural concepts used by MPLS are generic enough to allow it to operate as a peer VPN model for switching technology for a variety of link-layer technologies, and in heterogeneous Layer 2 transmission and switching environments. MPLS requires protocol-based routing functionality in the intermediate devices, and operates by making the interswitch transport infrastructure visible to the routing. In the case of IP over ATM, each ATM bearer link becomes visible as an IP link, and the ATM switches are augmented with IP routing functionality. IP routing is used to select a transit path across the network, and these transit paths are marked with a sequence of labels that can be thought of as locally defined forwarding path indicators. MPLS itself is performed using a label swapping forwarding structure. Packets entering the MPLS environment are assigned a local label and an outbound interface based on a local forwarding decision. The local label is attached to the packet via a lightweight encapsulation mechanism. At the next MPLS switch, the forwarding decision is based on the incoming label value, where the incoming label determines the next hop interface and next hop label, using a local forwarding table indexed by label. This lookup table is generated by a combination of the locally used IP routing protocol, together with a label distribution protocol, which creates end-to-end transit paths through the network for each IP destination. It is not our intention to discuss the MPLS architecture in detail, apart from noting that each MPLS switch uses a label-indexed forwarding table, where the attached label of an incoming packet determines the next-hop interface and the corresponding outgoing label.

The major observation here is that this lightweight encapsulation, together with the associated notion of boundary-determined transit paths, provides many of the necessary mechanisms for the support of VPN structures[9]. MPLS VPNs have not one, but three key ingredients: (1) constrained distribution of routing information as a way to form VPNs and control inter-VPN connectivity; (2) the use of VPN-IDs, and specifically the concatenation of VPN-IDs with IP addresses to turn (potentially) nonunique addresses into unique ones; and (3) the use of label switching (MPLS) to provide forwarding along the routes constructed via (1) and (2). The generic architecture of deployment is that of a label-switched common host network and a collection of VPN environments that use label-defined virtual circuits on an edge-to-edge basis across the MPLS environment. An example is indicated in Figure 4, which shows how MPLS virtual circuits are constructed.

Figure 4: MPLS "Tunnels," or VPNs



Numerous approaches are possible to support VPNs within an MPLS environment. In the base MPLS architecture, the label applied to a packet on ingress to the MPLS environment effectively determines the selection of the egress router, as the sequence of label switches defines an edge-to-edge virtual path. The extension to the MPLS local label hop-by-hop architecture is the notion of a per-VPN global identifier (or *Closed User Group* (CUG) identifier, as defined in [5]), which is used effectively within an edge-to-edge context. This global identifier could be assigned on ingress, and is then used as an index into a per-VPN routing table to determine the initial switch label. On egress from the MPLS environment, the CUG identifier would be used again as an index into a per-VPN global identifier table to undertake next-hop selection.

Routing protocols in such an environment need to carry the CUG identifier to trigger per-VPN routing contexts, and a number of suggestions are noted in [5] as to how this could be achieved.

It should be stressed that MPLS itself, as well as the direction of VPN support using MPLS environments, is still within the area of active research, development, and subsequent standardization within the IETF, so this approach to VPN support is still somewhat speculative in nature.

Link-Layer Encryption

As mentioned previously, encryption technologies are extremely effective in providing the segmentation and virtualization required for VPN connectivity, and can be deployed at almost any layer of the protocol stack. Because there are no intrinsically accepted industry standards for link-layer encryption, all link-layer encryption solutions are generally vendor specific and require special encryption hardware.

Although this scenario can avoid the complexities of having to deal with encryption schemes at higher layers of the protocol stack, it can be economically prohibitive, depending on the solution adopted. In vendor proprietary solutions, multivendor interoperability is certainly a genuine concern.

Transport and Application-Layer VPNs

Although VPNs can certainly be implemented at the transport and application layers of the protocol stack, this setup is not very common. The most prevalent method of providing virtualization at these layers is to use encryption services at either layer; for example, encrypted e-mail transactions, or perhaps authenticated *Domain Name System* (DNS) zone transfers between different administrative name servers, as described in DNSSec (*Domain Name System Security*)[10].

Some interesting, and perhaps extremely significant, work is being done in the IETF to define a *Transport Layer Security* (TLS) protocol[11], which would provide privacy and data integrity between two communicating applications. The TLS protocol, when finalized and deployed, would allow applications to communicate in a fashion that is designed to prevent eavesdropping, tampering, or message forgery. It is unknown at this time, however, how long it may be before this work is finalized, or if it will be embraced by the networking community as a whole after the protocol specification is completed.

The significance of a "standard" transport-layer security protocol, however, is that when implemented, it could provide a highly granular method for virtualizing communications in TCP/IP networks, thus making VPNs a pervasive commodity, and native to all desktop computing platforms.

Non-IP VPNs

Although this article has focused on TCP/IP and VPNs, it is recognized that multiprotocol networks may also have requirements for VPNs. Most of the same techniques previously discussed can also be applied to multiprotocol networks, with a few obvious exceptions—many of the techniques described herein are solely and specifically tailored for TCP/IP protocols.

Controlled route leaking is not suitable for a heterogeneous VPN protocol environment, in that it is necessary to support all protocols within the common host network. GRE tunnels, on the other hand, are constructed at the network layer in the TCP/IP protocol stack, but most routable multiprotocol traffic can be transported across GRE tunnels (for example, IPX and AppleTalk). Similarly, the VPDN architectures of L2TP and PPTP both provide a PPP end-to-end transport mechanism that can allow per-VPN protocols to be supported, with the caveat that it is a PPP-supported protocol in the first place.

The reverse of heterogeneous VPN protocol support is also a VPN requirement in some cases, where a single VPN is to be layered above a heterogeneous collection of host networks. The most pervasive method of constructing VPNs in multiprotocol networks is to rely upon application-layer encryption, and the resulting VPNs are generally vendor proprietary, although some would contend that one of the most pervasive examples of this approach was the mainstay of the emergent Internet in the 1970s and 1980s—that of the UNIX-to-UNIX Copy Program (UUCP) network, which was (and remains) an open technology.

Quality-of-Service Considerations

In addition to creating a segregated address environment to allow private communications, the expectation that the VPN environment will be in a position to support a set of service levels also exists. Such per-VPN service levels may be specified either in terms of a defined service level that the VPN can rely upon at all times, or in terms of a level of differentiation that the VPN can draw upon the common platform resource with some level of priority of resource allocation.

Using dedicated leased circuits, a private network can establish fixed resource levels available to it under all conditions. Using a shared switched infrastructure, such as Frame Relay virtual circuits or ATM virtual connections, a quantified service level can be provided to the VPN through the characteristics of the virtual circuits used to implement the VPN.

When the VPN is moved away from such a circuit-based switching environment to that of a general Internet platform, is it possible for the Internet Service Provider to offer the VPN a comparable service level that attempts to quantify (and possibly guarantee) the level of resources that the VPN can draw upon from the underlying host Internet?

This area is evolving rapidly, and much of it remains within the realm of speculation rather than a more concrete discussion about the relative merits of various Internet QoS mechanisms. Efforts within the *Integrated Services Working Group* of the IETF have resulted in a set of specifications for the support of guaranteed and controlled load end-to-end traffic profiles using a mechanism that loads per-flow state into the switching elements of the network[12, 13]. There are numerous caveats regarding the use of these mechanisms, in particular relating to the ability to support the number of flows that will be encountered on the public Internet[14]. Such caveats tend to suggest that these mechanisms will not be the ones that are ultimately adopted to support service levels for VPNs in very large networking environments.

If the scale of the public Internet environment does not readily support the imposition of per-flow state to support guarantees of service levels for VPN traffic flows, the alternative query is whether this environment could support a more relaxed specification of a differentiated service level for overlay VPN traffic. Here, the story appears to offer more potential, given that differentiated service support does not necessarily imply the requirement for per-flow state, so stateless service differentiation mechanisms can be deployed that offer greater levels of support for scaling the differentiated service[15]. However, the precise nature of these differentiated service mechanisms, and their capability to be translated to specific service levels to support overlay VPN traffic flows, still remain in the area of future activity and research.

Conclusions

So what is a virtual private network? As we have discussed, a VPN can take several forms. A VPN can be between two end systems, or it can be between two or more networks. A VPN can be built using tunnels or encryption (at essentially any layer of the protocol stack), or both, or alternatively constructed using MPLS or one of the "virtual router" methods. A VPN can consist of networks connected to a service provider's network by leased lines, Frame Relay, or ATM, or a VPN can consist of dialup subscribers connecting to centralized services or other dialup subscribers.

The pertinent conclusion here is that although a VPN can take many forms, a VPN is built to solve some basic common problems, which can be listed as virtualization of services and segregation of communications to a closed community of interest, while simultaneously exploiting the financial opportunity of economies of scale of the underlying common host communications system.

To borrow a popular networking axiom, "When all you have is a hammer, everything looks like a nail." Every organization has its own problem that it must solve, and each of the tools mentioned in this article can be used to construct a certain type of VPN to address a particular set of functional objectives. More than a single "hammer" is

available to address these problems, and network engineers should be cognizant of the fact that VPNs are an area in which many people use the term generically—there is a broad problem set with equally as many possible solutions. Each solution has numerous strengths and also numerous weaknesses and vulnerabilities. No single mechanism for VPNs that will supplant all others in the months and years to come exists, but instead a diversity of technology choices in this area of VPN support will continue to emerge.

Acknowledgments

Thanks to Yakov Rekhter, Eric Rosen, and W. Mark Townsley, all of Cisco Systems, for their input and constructive criticism.

References

[1] Valencia, A., M. Littlewood, and T. Kolar. "Layer Two Forwarding (Protocol) 'L2F'." **draft-valencia-l2f-00.txt**, work in progress, October 1997.

[2] Droms, R. "Dynamic Host Configuration Protocol." RFC 2131, March 1997.

[3] Kent, S., and R. Atkinson. "Security Architecture for the Internet Protocol." **draft-ietf-ipsec-arch-sec-04.txt**, work in progress, March 1998.

[4] Additional information on IPSec can be found on the IETF IPSec home page, located at **http://www.ietf.org/html.charters/ipsec-charter.html**

[5] Heinanen, J. "Multiprotocol Encapsulation over ATM Adaptation Layer 5." RFC 1483, July 1993.

[6] The ATM Forum. "Multi-Protocol Over ATM Specification v1.0." **af-mpoa-0087.000**, July 1997.

[7] Callon, R., P. Doolan, N. Feldman, A. Fredette, G. Swallow, and A. Viswanathan. "A Framework for Multiprotocol Label Switching." **draft-ietf-mpls-framework-02.txt**, work in progress, November 1997.

[8] Rosen, E., A. Viswanathan, and R. Callon. "A Proposed Architecture for MPLS." **draft-ietf-mpls-arch-01.txt**, work in progress, March 1998.

[9] Heinanen, J. and E. Rosen. "VPN Support for MPLS." **draft-heinanen-mpls-vpn-01.txt**, work in progress, March 1998.

[10] Eastlake, D. and C. Kaufman. "Domain Name System Security Extensions." RFC 2065, January 1997. For further information regarding DNSSec, see: **http://www.ietf.org/html.charters/dnssec-charter.html**

[11] Dierks, T. and C. Allen. "The TLS Protocol—Version 1.0." **draft-ietf-tls-protocol-05.txt**, work in progress, November 1997. For more information on the IETF TLS working group, see **http://www.ietf.org/html.charters/tls-charter.html**. See also the article on SSL in the *Internet Protocol Journal*, Volume 1, No. 1, June 1998.

[12] Wroclawski, J. "Specification of the Controlled-Load Network Element Service." RFC 2211, September 1997.

[13] Shenker, S., C. Partridge, and R. Guerin. "Specification of Guaranteed Quality of Service." RFC 2212, September 1997.

[14] Mankin, A., F. Baker, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, and L. Zhang. "Resource ReSerVation Protocol (RSVP) Version 1—Applicability Statement, Some Guidelines on Deployment." RFC 2208, September 1997.

[15] "Differentiated Services Operational Model and Definitions." **draft-nichols-dsopdef-00.txt**, work in progress, K. Nichols and S. Blake (editors), February 1998.

PAUL FERGUSON is a consulting engineer at Cisco Systems and an active participant in the Internet Engineering Task Force (IETF). His principal areas of expertise include large-scale network architecture and design, global routing, Quality of Service (QoS) issues, and Internet Service Providers. Prior to his current position at Cisco Systems, he worked in network engineering, analytical, and consulting capacities for Sprint, Computer Sciences Corporation (CSC), and NASA. He is coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, published by John Wiley & Sons, ISBN 0-471-24358-2, a collaboration with Geoff Huston. E-mail: **ferguson@cisco.com**

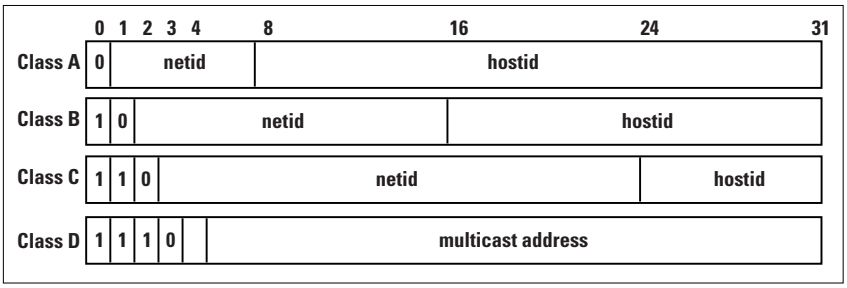
GEOFF HUSTON holds a B.Sc and a M.Sc from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Technologist in the Internet area for Telstra. He is also an active member of the IETF, and was an inaugural member of the Internet Society Board of Trustees. He is coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, published by John Wiley & Sons, ISBN 0-471-24358-2, a collaboration with Paul Ferguson. E-mail: **gih@telstra.net**

Reliable Multicast Protocols and Applications

by C. Kenneth Miller, StarBurst Communications

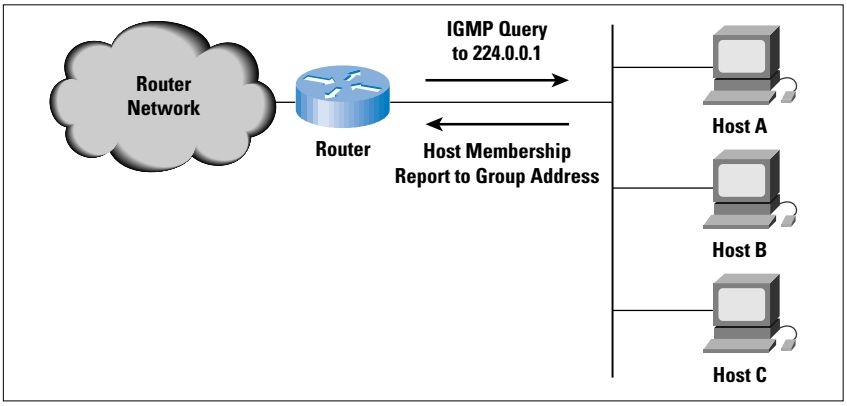
Multicast IP network services offer new opportunities to provide value-added applications that involve many-to-many transmission such as conferencing or network gaming, or one-to-many transmission such as multimedia events, tickertape feeds, and file transfer, where the many could be thousands or even conceivably millions. Multicast IP services use a different kind of IP address, called Class D. In contrast to individual host addresses (Classes A–C), which include a host and a network component and usually are semipermanent, Class D multicast addresses may by design be used only for a particular session, or can be semipermanent, as multicast groups may be set up and torn down relatively quickly, on the order of seconds. The IP address structure is shown in Figure 1.

Figure 1: IP Address Types



Hosts join groups at the receiver's initiation using the *Internet Group Management Protocol* (IGMP). When a host joins a group, it notifies the nearest multicast subnet router of its presence in the group, as shown in Figure 2. First defined in RFC 1112[1], IGMPv1 is still the version of IGMP most widely supported. IGMPv2 has recently been documented as an official RFC (RFC 2236[2]). The main feature that IGMPv2 brings is reduced latency for leaving groups. In IGMPv1, the designated multicast router for the subnet polls for multicast group members; no response between polls indicates that all hosts in a particular multicast group have left the group, and that the routers can prune back the multicast routing tree.

Figure 2: IGMPv1 Dialog



Network infrastructure devices, for example, routers, need to provide a routing protocol to forward multicast packets to group members, in a fashion similar to that performed for unicast routing. Multicast IP packet forwarding is best effort, just as it is with unicast packet forwarding. However, most unicast applications use TCP as a transport layer to provide guaranteed packet ordering and delivery. Some examples of applications that use TCP are the *File Transfer Protocol* (FTP) for file transfer and the *Hypertext Transfer Protocol* (HTTP) for Web access.

However, TCP is a unicast (point-to-point) only transport protocol. Thus, all multicast applications must run on top of the *User Datagram Protocol* (UDP) or alternatively, interface directly to IP via "raw" sockets and provide their own customized transport layer, as shown in Figure 3. UDP provides only minimal transport-layer services, error detection, and port multiplexing. Thus, if any errors or packet loss due to congestion occur, packets are simply lost to the application, and they are not recoverable. Thus, *all multicast applications must have a specific transport-layer service to support that particular application*. When that transport layer operates over UDP, it operates in the application layer with the application. When it interfaces directly to IP using "raw" sockets, the specialized transport layer operates at the transport layer, but is specialized to the particular application that uses it.

It should be noted that TCP supports only data reliability; it is not suited for transport of multimedia streams, which require consistent time delivery at the receiver and only need to be semireliable. Thus, multimedia streaming applications need a specialized transport layer such as the *Real-Time Transport Protocol* (RTP)[3] for unicast as well as multicast transmissions.

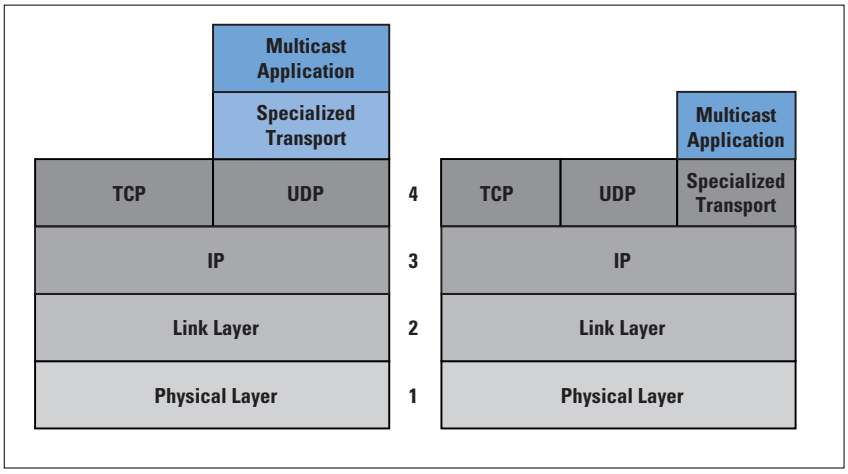


Figure 3: Specialized Multicast Transport Protocols Operate over UDP or IP

Many equate multicast with multimedia, thinking that the Internet and private intranets will become an alternative entertainment media to television by using multicast IP network services and multimedia streaming technology. However, numerous other multicast applications require reliability rather than timeliness; they are multicast applications that are similar to those unicast applications that operate over TCP, except that delivery is to many recipients rather than just one.

Reliable Multicast Application Categories and Requirements

Reliable multicast applications come in three basic categories with differing requirements, as shown in Figure 4.

Figure 4: Reliable Multicast Application Categories

Application Type	Latency Req.	Reliability	Scalability
Collaborative	Low	Semi/Strict	<100
Message Str.	Low/Medium	Semi/Strict	to Millions
Bulk Data	Not Real Time	Strict	to Millions

Collaborative applications such as data conferences (whiteboarding) and network-based games are many-to-many applications with modest scaling requirements of less than 100 participants. This kind of application requires low latency of less than 400 msec so that responses do not cause discomfort to the human participants. Transmission does not always need strict reliability; for example, refresh of background information for a network game could wait for the next refresh.

Message streaming applications such as tickertape and news feeds also often require low latency. Tickertape feeds to brokerage houses need to be very timely because the information loses value greatly with time. Time is very much money in this application, and there is also a need for strict reliability.

Tickertape feeds to consumers are purposely delayed by minutes because they are usually transmitted without charge, but they cannot be so stale as to be viewed as "old" information. This data does not have a strict reliability requirement because the next trade of a particular security refreshes the data. News feeds likewise have only a moderate latency requirement. If the news feeds are sent in a carousel fashion, that is, each news story is repeated, strict reliability may not be needed because it is refreshed in the next transmission of the same story.

Bulk data delivery has no specific latency requirement. Often there is a desire to schedule delivery during the night, when there is less network traffic. At other times, the desire is to receive the data almost

immediately. However, at all times the entire "file" or piece of data needs to be received to be complete. Strict reliability is the rule; for example, if any bit of a software image is lost, the data is worthless.

Message streaming and bulk data application scaling requirements span the gamut from tens to possibly even millions.

Reliable multicast transport protocols, in contrast to multimedia streaming transport protocols, have not yet been standardized. However, numerous reliable multicast protocols exist; some have been used only for research, while others have been commercialized.

The *Reliable Multicast Research Group* (RMRG) in the *Internet Research Task Force* (IRTF) is now studying reliable multicast. It is chartered to recommend techniques for a working group in the *Internet Engineering Task Force* (IETF) to create a set of reliable multicast standards.

Standardization Effort

The standardization effort has been started in an IRTF research group to study the problems and possible solutions by Internet researchers. This effort was first placed in the hands of researchers because the problems were considered very difficult to solve in the global Internet. Some of the concerns about reliable multicast were discussed in an expired Internet Draft published in November 1996 by the Transport Area Directors of IETF.

These concerns formed the basis for the work of the RMRG, which was formed in early 1997. The concerns from that document follow:

"A particular concern for the IETF (and a dominant concern for the Transport Services Area) is the impact of reliable multicast traffic on other traffic in the Internet in times of congestion (more specifically, the effect of reliable multicast traffic on competing TCP traffic). The success of the Internet relies on the fact that best-effort traffic responds to congestion on a link (as currently indicated by packet drops) by reducing the load presented on that link. Congestion collapse in today's Internet is prevented only by the congestion control mechanism in TCP.

There are a number of reasons to be particularly attentive to the congestion-related issues raised by reliable multicast proposals. Multicast applications in general have the potential to do more congestion-related damage to the Internet than do unicast applications. This is because a single multicast flow can be distributed along a large, global multicast tree reaching throughout the entire Internet.

> Further, reliable multicast applications have the potential to do more congestion-related damage than do unreliable multicast applications. First, unreliable multicast applications such as audio and video are, at the moment, usually accompanied by a person at the receiving end, and people typically unsubscribe from a multicast group if congestion is so heavy that the audio or video stream is unintelligible. Reliable multicast applications such as group file transfer applications, on the other hand, are likely to be between computers, with no humans in attendance monitoring congestion levels.
>
> In addition, reliable multicast applications do not necessarily have the natural time limitations typical of current unreliable multicast applications. For a file transfer application, for example, the data transfer might continue until all of the data is transferred to all of the intended receivers, resulting in a potentially-unlimited duration for an individual flow. Reliable multicast applications also have to contend with a potential explosion of control traffic (e.g., ACKs, NAKs, status messages), and with control traffic issues in general that may be more complex than for unreliable multicast traffic.
>
> The design of congestion control mechanisms for reliable multicast for large multicast groups is currently an area of active research. The challenge to the IETF is to encourage research and implementations of reliable multicast, and to enable the needs of applications for reliable multicast to be met as expeditiously as possible, while at the same time protecting the Internet from the congestion disaster or collapse that could result from the widespread use of applications with inappropriate reliable multicast mechanisms. Because of the setbacks and costs that could result from the widespread deployment of reliable multicast with inadequate congestion control, the IETF must exercise care in the standardization of a reliable multicast protocol that might see widespread use."

One of the statements in this document is very specious:

> "First, unreliable multicast applications such as audio and video are, at the moment, usually accompanied by a person at the receiving end, and people typically unsubscribe from a multicast group if congestion is so heavy that the audio or video stream is unintelligible. Reliable multicast applications such as group file transfer applications, on the other hand, are likely to be between computers, with no humans in attendance monitoring congestion levels."

This statement is a very weak argument; it is not reliable to depend on a human to turn off a nonfunctioning event. Do we typically turn off the television when we leave the house? Or leave the room to do something else?

In contrast, some of the reliable multicast protocols such as the *Multicast File Transfer Protocol* (MFTP) have the sense of a finite session, and automatically time out and leave a group, even if all group members did not receive all the content.

Essentially what is desired is a reliable multicast protocol that behaves like TCP in that it backs off in the face of congestion approximately the same way as TCP and shares the bandwidth with TCP traffic "fairly." This feature is of prime importance to Internet researchers who wish to specify protocols that can scale to the global Internet and not cause harm to the traffic already present.

Two additional significant problems need to be solved: scalability and the ability to operate with scalability over many different network infrastructures.

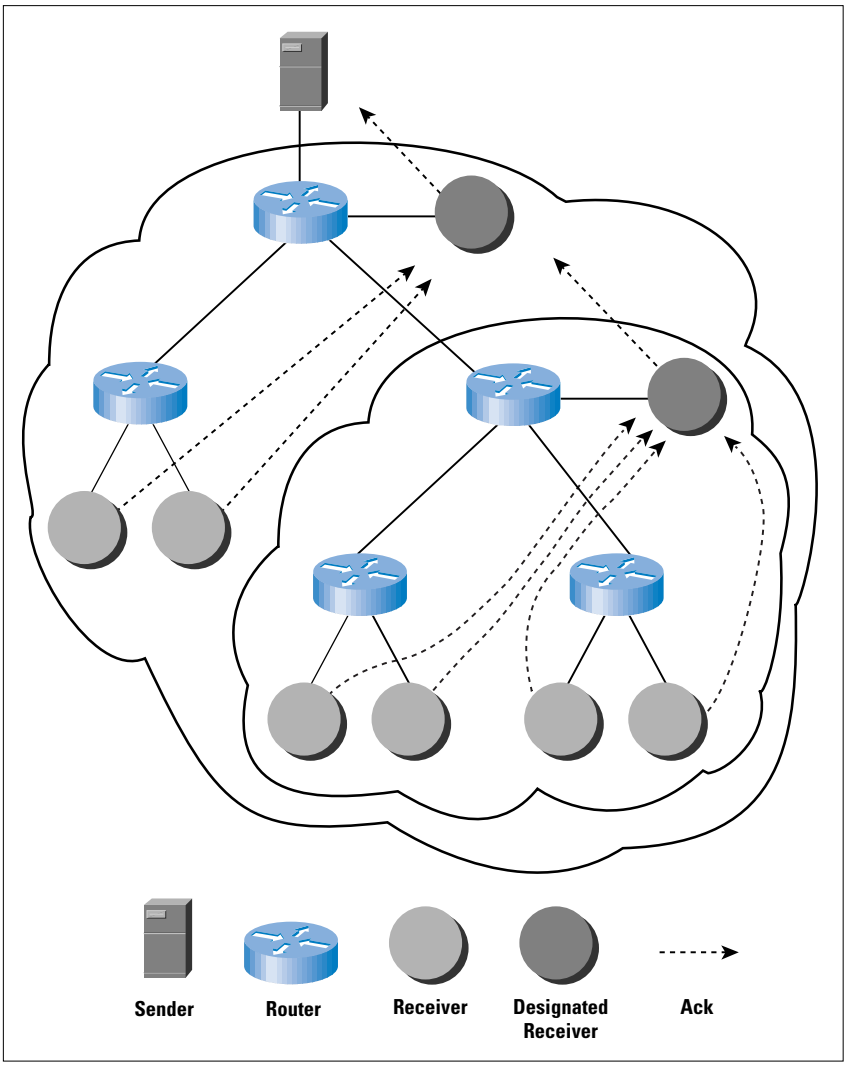
Scaling Issues and How Current Reliable Multicast Protocols Solve Them

Two primary issues are related to scaling, that is, the ability to handle large groups. The first and most significant is widely known as acknowledgment/negative acknowledgment (ACK/NAK) implosion. As the number of receivers grows, the amount of back traffic to the sender eventually overwhelms its capacity to handle them. Additionally, the network at the sender site becomes congested from the cumulative back traffic from the receivers.

The second issue is one of retransmissions (often referred to as "repairs"). If the packet loss is uncorrelated at the receivers, retransmissions grow, so the data may need to be sent multiple times to satisfy all the receivers. Measurements of the *Multicast backbone* (Mbone) have shown that loss consists of both correlated and uncorrelated parts[4]. Satellite networks will also exhibit mostly uncorrelated loss, unless receivers are geographically close.

Various methods have been used to achieve scaling by reducing the amount of ACK/NAK administrative traffic while still retaining reliability. A straightforward approach is to simply deploy repeaters/aggregators in the network, as shown in Figure 5. This approach is provided by the *Reliable Multicast Transport Protocol* (RMTP)[5]. RMTP provides for *designated receivers* (DRs) that collect status messages from nodes in a local RMTP domain and provide repairs (retransmissions of missing data), if available. Receivers direct the administrative messages to the DR by unicast. Thus, the DR provides both local recovery and consolidation of control traffic to the next DR in the hierarchy if the data requested is not available.

Figure 5:
RMTP Designated
Receivers



A second approach is to allow any receiver to provide the repair, biasing the request to the nearest receiver that has the requested data. This approach, called *Scalable Reliable Multicast* (SRM)[6], depends on the concept of repair by any receiver that has the data to gain scalability in reducing administrative back traffic to the source, putting the onus of responsibility on receivers to ensure that they get missed data.

Group members in SRM send low-frequency *session* messages to the group so that their neighbors can learn their status, measure the delay among group members and learn group membership, and detect the last packet in a burst. Session messages are designed to take only about five percent of the traffic in the session.

Receivers with missing data wait a random time period before issuing repair requests, allowing suppression of duplicate requests similar to the mechanism that IGMP uses on its subnet. A similar process occurs for making the actual repairs. The random backoff time for both repair requests made by receivers and repairs made by senders is a

function of "closeness" to the sender and requesting receiver. Thus, those closest to each other time out first and make the repair request or the actual repair in an attempt to keep repairs as local as possible. A receiver that sees the first request and determines that it is the same request that it would have made simply stays silent, reducing potential redundant requests. The requester continues to send repair requests until the repair is received.

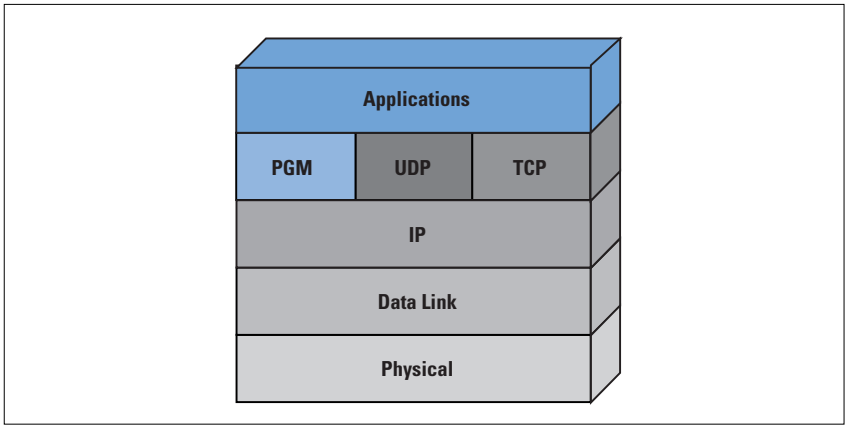
Any receiver may satisfy the repair request, because all receivers are required to cache previously sent data. Any receiver that can satisfy the request is prepared to do so; a random backoff timer is used before a repair is sent, and if it sees the repair being sent by another group member, it stays silent to reduce the probability of sending duplicate repairs.

SRM was first developed to be the reliable multicast protocol to operate with the *wb* whiteboard data conferencing tool developed by Lawrence Berkeley Labs (LBL) researchers, SRM is currently operational over the Mbone, the experimental multicast network of the Internet.

A third approach is to have the network infrastructure, that is, routers, help in providing scaling. This approach, called *Pretty Good Multicast* (PGM)[7], is a new proposal that was first publicly presented to the RMRG meeting held in February 1998.

One design goal of the creators of PGM was simplicity and the ability to optimally leverage routers in the network to provide scalability. PGM is an example of a protocol that bypasses UDP and interfaces directly to IP via "raw" sockets, as shown in Figure 6.

Figure 6: PGM Interfaces Directly to IP



PGM provides no notion of group membership; it simply provides reliability within a source's transmit window from the time a receiver joins a group until it departs.

PGM has only a few data packets that are defined:

ODATA: original content data

NAK: selective negative acknowledgment

NCF: NAK confirmation

RDATA: retransmission (repair)

SPM: source path message

Each PGM packet contains a *Transport Session Identifier* (TSI) to identify the session and source of that data, so multiple sessions may be easily identified by PGM-aware routers and receivers. ODATA, NCF, RDATA, and SPM packets flow downstream in the distribution tree, and NAK packets flow upstream toward the source.

PGM is designed for scalability as well as the ability to serve real-time applications. Thus there is a need for timeliness. This need is handled by the *transmit window*, which defines a sliding window of data such that if no NAKs are received by the sender or a designated local retransmitter by the time the window is up, the data is simply not available for repairs.

PGM is totally NAK based, so the scaling issue is to reduce the number of NAKs sent back to the source, while at the same time protecting against lost NAKs. Enter here the router assist, as shown in Figure 7.

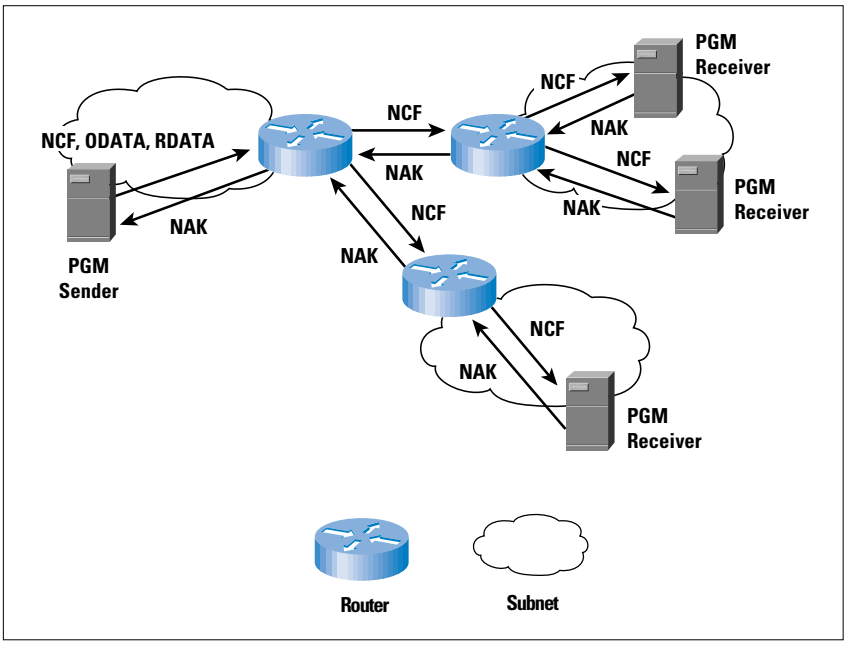


Figure 7: PGM NAK/NCF Dialog

NAKs are unicast from PGM-router to PGM-router, initiated by the receiver that lost data sending a NAK to its nearest PGM-aware router. Each PGM-aware router keeps forwarding NAKs until it sees an NCF or RDATA, which indicates that a repair is being sent. NAK *suppres-*

sion is provided by a receiver's subnet PGM-aware router, and all PGM-aware routers *eliminate* duplicate NAKs all the way upstream to the source.

The unicast path back to the source must be the same path as the downstream multicast tree. SPMs are sent downstream interleaved with ODATA packets to establish a source path state for a given source and session. PGM-aware routers use this information to determine the unicast path back to the source for forwarding NAKs. SPMs also alert receivers that the oldest data in the transmit window is about to be retired from the window and will thus no longer be available for repairs from the source. SPMs are sent by a source at a rate that is at least the rate at which the transmit window is advanced. This rate provokes "last call" NAKs from receivers and updates the receive window state at receivers.

PGM-aware routers also keep state on where the NAKs come from in the distribution tree so that they may constrain the forwarding of RDATA repairs to only those ports from which NAKs requesting that repair were received. This scenario eliminates the transmission of repair data to parts of the distribution tree where the repair is not needed.

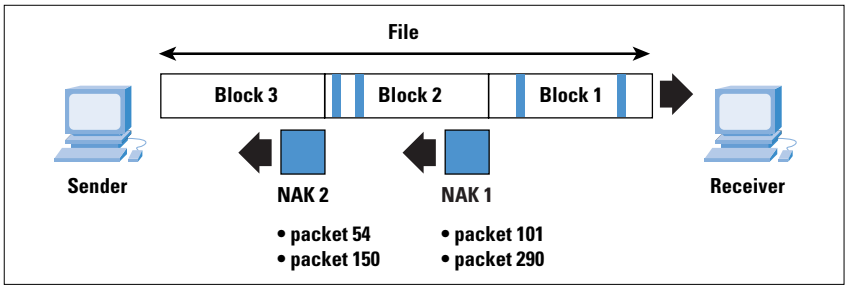
The PGM feature can also optionally redirect NAKs to a *designated local retransmitter* (DLR) rather than the source. A DLR announces its presence to provoke the redirection of NAKs for that session and source.

A fourth approach is to not have a low-latency requirement (that is, only serve "bulk data" delivery applications) and use this feature to advantage to gain scalability. MFTP was first published as an Internet Draft in February 1997, and an update was submitted in April 1998[8].

MFTP also has a provision for sender-based group creation, with different group models, and the group setup protocol to notify receivers to join the group. Group creation is discussed later in this article.

The basic MFTP protocol breaks the data entity to be sent into maximum size "blocks," where a block by default consists of thousands or tens of thousands of packets, depending on packet size used. This setup is shown in Figure 8.

Figure 8: MFTP Blocks



MFTP is a "NAK-only" protocol; that is, if data is received correctly in a block, nothing is sent back to the sender. If one or more packets are in error or missing in a block, receivers respond with a NAK that consists of a bit map of the bad packets in the block. It is thus a *selective reject* mechanism. In this respect, MFTP is similar to RMTP; the main difference is that MFTP explicitly attempts to make the block as large as possible for scaling purposes.

NAKs are normally sent unicast back to the source, unless aggregation to improve scaling using enabled network routers is used. In this case, the NAKs are sent multicast to a special administrative traffic group address.

MFTP does not repair after each block, however; it takes advantage of the non-real time nature of the application for benefit. The data entity, such as a file, is sent initially in its entirety in a *first pass*. The sender collects the NAK packets for a block from all the receivers. One NAK packet from a receiver can represent thousands or even tens of thousands of bad packets, reducing NAK implosion by orders of magnitudes. The collection of NAKs received by the sender from all the receivers is logically OR-ed together to represent the collective need for repairs for the receiving group. These repairs are sent by the sender in a *second pass* to the group. If certain receivers already have the repair, it is simply ignored. This scenario is repeated, if necessary, until all repairs are received by all receivers or until a configurable timeout occurs.

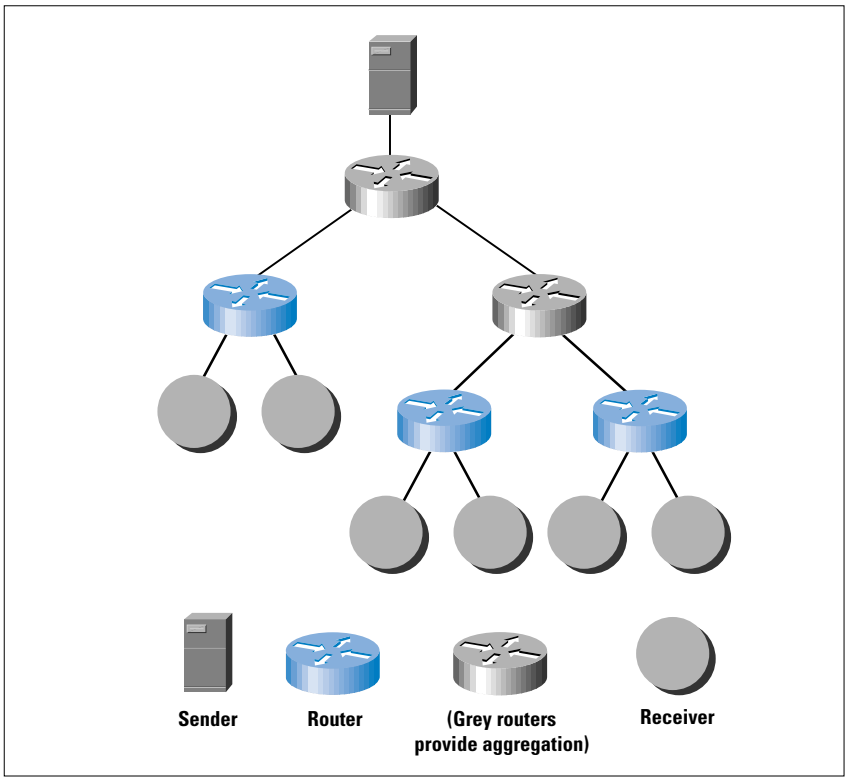
Thus, packet ordering services are not provided, and holes in the data caused by dropped packets or packets in error are filled in as they are received.

The sender is *rate based;* in other words, it transmits at a data rate set by the operator to be less than or equal to what the network can handle. The protocol is thus very efficient with high-latency networks such as satellites, and it is impervious to network asymmetry. It also attempts to be as scalable as possible on one-hop networks such as satellite networks, and it provides for extensions so that network elements may aggregate downstream responses to increase scalability further, depending on the network configuration.

This aggregation capability is shown in Figure 9. The network element, which can be a router, collects MFTP administrative back traffic routers are members. These routers aggregate back traffic from all nodes downstream in the multicast tree from the source, including registrations, NAKs, and dones. Registration and done messages are used by MFTP's group setup protocol, and they are described later in this article.

Depending on the network configuration, this aggregation capability can further improve the scalability of MFTP by orders of magnitudes.

Figure 9: Routers as Network Aggregators



The upper limit to scalability with no network aggregation of administrative traffic is in the tens of thousands of receivers. For example, for a *Maximum Transmission Unit* (MTU) of 1500 bytes (the Ethernet maximum), the default block size is over 11,000 packets. If the number of receivers is 10,000 and each receiver has at least one bad packet per block, then there will be a total of 10,000 NAK packets coming back to the sender from the group about that block, approximately the same number of packets as were sent in the forward direction in that block. MFTP provides for a NAK backoff timer to spread the NAKs out in time to the sender to avoid bursts. If the bandwidth is symmetric at the sender, the sender should be able to handle this maximum NAK. In many situations, the amount of back traffic could exceed forward traffic.

MFTP also has provision for a crude congestion control mechanism. The sender at the beginning of a session sends *announce* messages. These messages are used for many functions, including the setting up of groups. Additionally, it conveys a packet loss parameter to all receivers. This packet loss threshold parameter may be used by receivers to leave the group if the packet loss exceeds the threshold. Leaving the group prunes the distribution tree, relieving the congestion in that section of the tree.

Commercial Usage

The reliable multicast protocols previously discussed are the most prominent ones on the market today. RMTP has been deployed in its message streaming version for a billing record distribution application within a very large telecommunications carrier, but it has had generally limited deployment. It also does not scale over satellite networks, where most of the early multicast deployments reside.

SRM has been used by the research community only over the Mbone, and it is still being refined. Another problem with SRM is that in its current incarnation, it supports neither asymmetric nor satellite networks. Some early Internet Service Provider (ISP) multicast implementations, offer multicast support in only one direction; SRM requires total multicast support.

PGM is new and offers promise, but there is no deployment yet, and it likely will not occur until early 1999. PGM also requires router support in a terrestrial land-line network to gain scaling.

MFTP has the limitation that it supports only bulk transfer applications. However, one trade-off is that it can support all network infrastructures, including satellite infrastructures with scaling. MFTP has also been available commercially in products with the longest application support, dating back to 1995. Thus, MFTP-based products have the largest installed base of any reliable multicast-based product being used over WANs. The largest commercial installation of over 8,500 remote sites in the group is the General Motors[9] dealer network. Several other commercial installations of MFTP-based applications number over 1,000 group members.

Advanced Research Topics Discussed in Reliable Multicast Research Group

A promising technique to reduce the amount of repair data that needs to be retransmitted is called *erasure correction*. This technique can significantly reduce the amount of repairs that need to be resent if the packet loss is largely uncorrelated at the receivers. It uses a *forward error correction* (FEC) code to generate parity packets to be used for repairs only. This setup provides benefit if errors at receivers are uncorrelated. For example, suppose 16 receivers each have one missing packet, but they are all different. Rather than send all 16 original data packets, one FEC packet could be sent that could correct the one missing packet at all 16 receivers, requiring retransmission of only one packet rather than 16.

If the loss is correlated, then many of the receivers lose the same data, and erasure correction is of no benefit. However, there is also no penalty, except for the need for computing power at both the sender and the receivers to perform the FEC correction calculations. Simulations have show[10] that there is a greater than 2:1 reduction in the number of repairs needed to be sent with our example of 10,000 receivers. This benefit will be even larger when group sizes become larger than tens of thousands.

Perhaps a more significant application for FEC is a congestion control technique known as *layering*[11,12]. With layering, numerous groups are set up by the sender, all with different rates. Receivers that can receive at the highest rate join all the "layer" groups. Those receivers that cannot receive at the highest rate simply leave "layers" until congestion is relieved, and they take longer to receive the data. For this to work without sending data redundantly, the number of parity packets created must be very large compared to the number of data packets.

There are some further issues that have been pointed out by the researchers with the Other issues with the layering approaches have been pointed out by the researchers, however. For layering to be effective, the routing tree should be identical for the different groups; otherwise congestion will not be relieved on a part of the tree. This may not always be the case, especially in sparse mode routing protocols, where selection of the rendezvous point or core is based on group address.

Even if the same distribution tree is used for the different layers, it has been pointed out[12] that leaves of hosts downstream from a congested link should be coordinated; otherwise the action of less than all of them has no effect on congestion. Additionally, a receiver could cause congestion by adding a layer that another receiver could interpret as congestion, causing it to drop a layer with no effect.

Thus, layering using FEC techniques is an interesting technique that shows promise for use in congestion control. However, there are issues associated with this type of layering that researchers still need to address.

Another technique that has been proposed for congestion control is bulk feedback to the sender[13]. If the sender receives an excessive number of NAKs from receivers, it drops the sender's transmission rate with an algorithm that attempts to emulate the behavior of TCP. This approach is an obvious one because it is an extension of the process in which TCP falls back in the face of congestion.

This approach, however, has two basic problems. The first is that there is delay, because the sender needs to get feedback from the multitude of receivers before it acts. This delay can be considerably longer than in the case of TCP, which needs feedback from only one receiver.

The second flaw is that one errant receiver can effectively penalize the whole group, because the sender reduces the rate to the total group.

This approach is not viewed as a viable solution for these reasons. In fact, the general consensus is that congestion control decision making will be required at the multiple receivers rather than at the sender for both scaling and timeliness reasons.

Another idea that is now receiving intense study by researchers is that of "subcasting"[14,15,16]. The key idea in subcasting is to optimize local repair to be a retransmitter that may be just above a link congestion point, as shown in Figure 10. The problem is to gain knowledge of the network topology so as to locate a receiving host that is willing to retransmit and that has the repair data.

Then the repairs need to be contained within only the region of the network that lost the original transmission, that is, the "subcast" region.

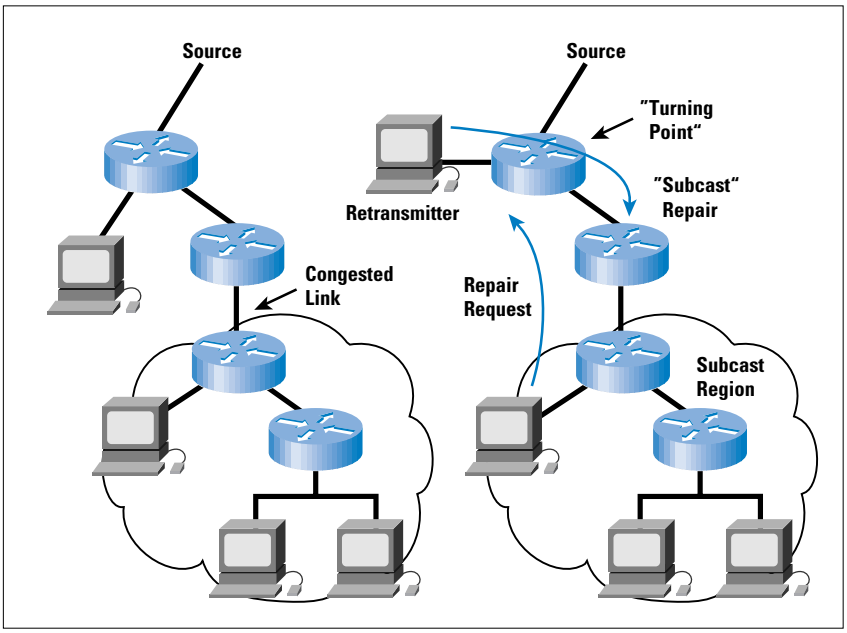


Figure 10: Optimized Local Repair

One proposal is to ask for assistance from the network routers. They know the topology and could be used to find the closest willing retransmitter that has the repair. The router could also direct the repair to only the affected region: the *subcast*.

This technique can be viewed as an extension of concepts originally proposed in SRM to provide local recovery. It assumes that most loss is caused by congested links, and that uncorrelated loss is caused by a series of mildly congested links with few group members. This model is probably the right one for many land-line routed networks; it is problematical with other network infrastructures.

Nevertheless, it is an interesting proposal that merits further research effort. Local repair is destined to be an important tool to meet the goal of improved scalability with minimal traffic overhead.

Group Creation and Destruction

The process of joining a group and leaving a group in IP multicast is left to a potential group member that uses IGMP to notify the nearest multicast router of its membership state. However, mechanisms need to be in place to allow potential members of a group to gain the information needed to decide to join the group.

There are two basic ways to accomplish this scenario for one-to-many sessions. The first and most common is the "broadcast TV" model. The *Multiparty Multimedia Session Control* (MMUSIC) working group of the IETF has developed some protocols that can be used to advertise content. The *Session Announcement Protocol* (SAP)[17] provides the mechanism to send a stream on a "well-known" multicast address to announce content to any potential listeners who may be interested. It uses the *Session Description Protocol* (SDP)[18] to describe the contents that are announced. These two protocols together have been used to create a session directory tool that is available on the Mbone. This setup creates essentially the equivalent of a "preview channel" such as is often available on cable television systems.

SDP is also used to post content on Web sites, which advertise that content to anyone who wishes to receive it.

Although these protocols were originally developed primarily to advertise multimedia streaming applications, they are also applicable for data. They provide a useful tool for "push" vendors to advertise multicast "channels" based on content that any consumer can "tune in" to.

Internet researchers describe this model as providing "loosely coupled" sessions, because the sender does not know who is listening, much like radio or TV broadcasters do not know who tunes in to their stations.

MFTP also includes a group setup protocol. The "closed group" option in MFTP provides a mechanism to create a "tightly coupled" session that is very useful to organizations that wish to deliver critical information from a central site to many remote branch offices. The closed group provides a means for the sender to define a group list centrally and direct those members so defined to join the group. This scenario is somewhat similar to e-mail, except more robust.

These instructions are sent in an "announce" message on a special multicast group address that the superset of possible candidate receivers always listens to. Hosts so directed to join the group notify their designated multicast router of their membership directed to join the group notify their designated multicast router of their membership using IGMP and "register" back to the sender of their presence. Thus, the sender knows group membership before transmission commences, and the sender can then also positively confirm delivery.

This approach has proven very desirable for organizations that have many branches where information is desired to be sent at the discretion and time determined by the and desire to send information at the discretion and time determined by the sender, and usually the information is delivered to a branch office server. Several deployments of applications that use MFTP and the closed group model with group members approaching 10,000 exist.

The MMUSIC group has also created the *Session Invitation Protocol* (SIP)[19], which is used to invite members to a conference of some sort, including possibly a data conference. This protocol is appropriate for use with whiteboard applications, for example.

Summary and Conclusions

Although multicast has often been viewed as synonymous with multimedia, there is a wide spectrum of reliable multicast applications that involve the transfer of data to multiple group members. Because this wide spectrum of applications has many different requirements, as shown in Figure 4, no one reliable multicast protocol can handle all applications and network infrastructures. The result is that numerous reliable multicast protocols are likely to become standardized, and today numerous reliable multicast protocols are either in commercial products/toolkits or due to be available soon.

The reliable multicast standardization effort now resides in the IRTF, because Internet researchers are concerned about congestion control and fairness to TCP for any protocols that might become standardized for general Internet use. This problem is difficult to solve, given the disparate requirements placed on protocols by the wide variety of applications and different network infrastructures.

Nevertheless, a significant number of reliable multicast-based product deployments have already occurred over private networks. These have been shown to save organizations much money and to help create new business opportunities for them.

Stay tuned; reliable multicast-based applications are ready to be mainstreamed. Together with multimedia multicast applications, multicast applications of all forms will become common soon, first in private intranets and extranets and then in the Internet as a whole.

References

[1] Deering, S. "Host Extensions for IP Multicasting." RFC 1112, August 1989.

[2] Fenner, W. "Internet Group Management Protocol, Version 2." RFC 2236, November 1997.

[3] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. "RTP: A Transport Protocol for Real-Time Applications." RFC 1889, January 1996.

[4] Handley, M. "An Examination of Mbone Performance." ISI Report, January 10, 1997.

[5] Paul, S., Sabnani, K. K., Lin, J. C., and Bhattacharyya, S. "Reliable Multicast Transport Protocol (RMTP)." IEEE *Journal on Selected Areas in Communications*, April 1997.

[6] Floyd, S., Jacobson, V., Liu, C., McCanne, S., and Zhang, L. "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing." ACM *Transactions on Networking*, November 1996.

[7] Farinacci, D., Lin, A., Speakman, T., and Tweedly, A. "PGM Reliable Transport Protocol Specification." Work in progress, Internet Draft, **`draft-speakman-pgm-spec-01.txt`**, January 29, 1998.

[8] Miller, K., Robertson, K., Tweedly, A., and White, M. "StarBurst Multicast File Transfer Protocol (MFTP) Specification." Work in progress, Internet Draft, **`draft-miller-mftp-spec-03.txt`**, April 1998.

[9] Miller, K. "Reliable Multicast Protocols: A Practical View." 22nd Conference on Local Computer Networks, November 1997.

[10] Kasera, S. K., Kurose, J., Towsley, D., "Scalable Reliable Multicast Using Multiple Multicast Groups." CMPSCI Technical Report TR 96-73, October 1996.

[11] Nonnenmacher, J., and Biersack, E. W. "Asynchronous Multicast Push: AMP." Proceedings of ICCC '97 International Conference on Computer Communications, Cannes, France, November 1997.

[12] Crowcroft, J., Rizzo, L., and Vicisano, L. "TCP-Like Congestion Control for Layered Multicast Data Transfer." Submitted to INFOCOM '98, August 1997.

[13] Sano, T., Yamanouchi, N., et al. "Flow and Congestion Control for Bulk Reliable Multicast Protocols—toward coexistence with TCP." Submitted to INFOCOM '98, presented at RMRG meeting in Cannes, France, September 1997.

[14] Hofmann, M. "Enabling Group Communication in Global Networks." Proceedings of Global Networking '97, June 1997.

[15] Papadopoulos, C., Parulkar, G., and Varghese, G. "An Error Control Scheme for Large-Scale Multicast Applications." Submitted to INFOCOM '98, presented at RMRG meeting in Cannes, France, September 1997.

[16] Levine, B. N., Paul, S., and Garcia-Luna-Aceves, J. J. "Deterministic Organization of Multicast Receivers Based on Packet Loss Correlation." Presented at RMRG meeting in Orlando, Fla., February 1998, submitted for publication.

[17] Handley, M. "SAP: Session Announcement Protocol." Work in progress, Internet Draft, **draft-ietf-mmusic-sap-00.txt**, November 1996.

[18] Handley, M., and Jacobson, V. "SDP: Session Description Protocol." Work in progress, Internet Draft, **draft-ietf-mmusic-sdp-07.txt**, April 1998.

[19] Handley, M., Schulzrinne, H., and Schooler, E. "SIP: Session Invitation Protocol." Work in progress, Internet Draft, **draft-ietf-mmusic-sip-04.txt**, November 1997.

(This article is based in part on material in the book *Multicast Networking and Applications* written by C. Kenneth Miller to be published by Addison Wesley Longman, Inc. in 1998.
ISBN 0-201-30979-3. Used with permission.)

C. KENNETH MILLER is the founder, Chairman, and Chief Technology Officer of StarBurst Communications. StarBurst Communications provides reliable multicast solutions for commercial applications with such corporate customers as GM, Ford, Chrysler, Toys 'R Us, Thomson Financial, and many others. Miller has been in the data communications industry since 1972. He founded Concord Data Systems in late 1980 and served as its President and CEO until 1986. Concord Data Systems produced high-speed dial modems. He was the author of the IEEE 802.4 LAN standard, which became the lower layer for the Manufacturing Automation Protocol (MAP) factory LAN standard. Miller was a regular columnist in *Data Communications Magazine* from 1992 to 1994. He has also published numerous articles and participated in many panels at trade show and other industry events. He is now writing a book entitled *Multicast Networking and Applications*, to be published in 1998 by Addison-Wesley. Miller received a BEE degree from Rensselaer Polytechnic Institute and a MSEE degree from the University of Pennsylvania, specializing in communications. Miller can be reached at **miller@starburstcom.com**

Layer 2 and Layer 3 Switch Evolution

by Thayumanavan Sridhar, Future Communications Software

Layer 2 switches are frequently installed in the enterprise for high-speed connectivity between end stations at the data link layer. Layer 3 switches are a relatively new phenomenon, made popular by (among others) the trade press. This article details some of the issues in the evolution of Layer 2 and Layer 3 switches. We hypothesize that that the technology is evolutionary and has its origins in earlier products.

Layer 2 Switches

Bridging technology has been around since the 1980s (and maybe even earlier). Bridging involves segmentation of local-area networks (LANs) at the Layer 2 level. A multiport bridge typically learns about the *Media Access Control* (MAC) addresses on each of its ports and transparently passes MAC frames destined to those ports. These bridges also ensure that frames destined for MAC addresses that lie on the same port as the originating station are not forwarded to the other ports. For the sake of this discussion, we consider only Ethernet LANs.

Layer 2 switches effectively provide the same functionality. They are similar to multiport bridges in that they learn and forward frames on each port. The major difference is the involvement of hardware that ensures that multiple switching *paths* inside the switch can be active at the same time. For example, consider Figure 1, which details a four-port switch with stations A on port 1, B on port 2, C on port 3 and D on port 4. Assume that A desires to communicate with B, and C desires to communicate with D. In a single CPU bridge, this forwarding would typically be done in software, where the CPU would pick up frames from each of the ports sequentially and forward them to appropriate output ports. This process is highly inefficient in a scenario like the one indicated previously, where the traffic between A and B has no relation to the traffic between C and D.

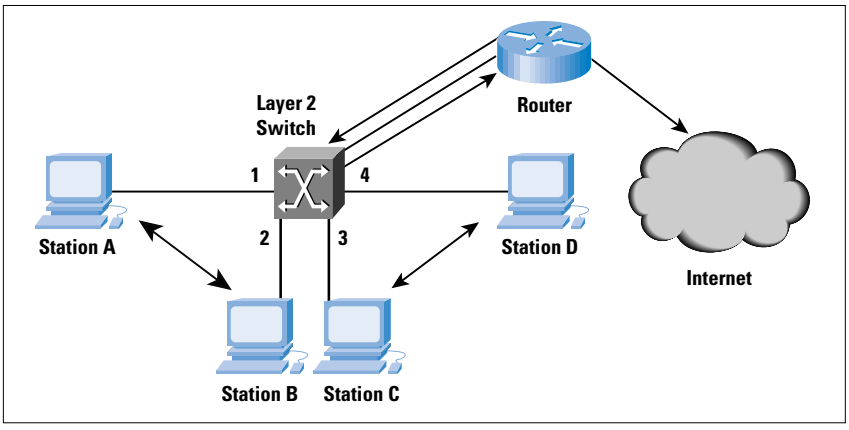


Figure 1: Layer 2 switch with External Router for Inter-VLAN traffic and connecting to the Internet

Enter hardware-based Layer 2 switching. Layer 2 switches with their hardware support are able to forward such frames in parallel so that A and B and C and D can have simultaneous conversations. The parallelism has many advantages. Assume that A and B are NetBIOS stations, while C and D are Internet Protocol (IP) stations. There may be no reason for the communication between A and C and A and D. Layer 2 switching allows this coexistence without sacrificing efficiency.

Virtual LANs

In reality, however, LANs are rarely so *clean*. Assume a situation where A,B,C, and D are all IP stations. A and B belong to the same IP subnet, while C and D belong to a different subnet. Layer 2 switching is fine, as long as only A and B or C and D communicate. If A and C, which are on two different IP subnets, need to communicate, Layer 2 switching is inadequate—the communication requires an IP router. A corollary of this is that A and B and C and D belong to different broadcast domains—that is, A and B should not "see" the MAC layer broadcasts from C and D, and vice versa. However, a Layer 2 switch cannot distinguish between these broadcasts—bridging technology involves forwarding broadcasts to all other ports, and it cannot tell when a broadcast is restricted to the same IP subnet.

Virtual LANs (VLANs) apply in this situation. In short, Layer 2 VLANs are Layer 2 broadcast domains. MAC broadcasts are restricted to the VLANs that stations are configured into. How can the Layer 2 switch make this distinction? By configuration. VLANs involve configuration of ports or MAC addresses. Port-based VLANs indicate that all frames that originate from a port belong to the same VLAN, while MAC address-based VLANs use MAC addresses to determine VLAN membership. In Figure 1, ports 1 and 2 belong to the same VLAN, while ports 3 and 4 belong to a different VLAN. Note that there is an implicit relationship between the VLANs and the IP subnets—however, configuration of Layer 2 VLANs does not involve specifying Layer 3 parameters.

We indicated earlier that stations on two different VLANs can communicate only via a router. The router is typically connected to one of the switch ports (Figure 1). This router is sometimes referred to as a *one-armed router* since it receives and forwards traffic on to the same port. In reality, of course, such routers connect to other switches or to wide-area networks (WANs). Some Layer 2 switches provide this Layer 3 routing functionality within the same box to avoid an external router and to free another switch port. This scenario is reminiscent of the large multiprotocol routers of the early '90s, which offered routing and bridging functions.

A popular classification of Layer 2 switches is "cut-through" versus "store-and-forward." Cut-through switches make the forwarding decision as the frame is being received by just looking at the header of the frame. Store-and-forward switches receive the entire Layer 2 frame

before making the forwarding decision. Hybrid adaptable switches which adapt from cut-through to store-and-forward based on the error rate in the MAC frames are very popular.

Characteristics

Layer 2 switches themselves act as IP end nodes for *Simple Network Management Protocol* (SNMP) management, Telnet, and Web based management. Such management functionality involves the presence of an IP stack on the router along with *User Datagram Protocol* (UDP), *Transmission Control Protocol* (TCP), Telnet, and SNMP functions. The switches themselves have a MAC address so that they can be addressed as a Layer 2 end node while also providing transparent switch functions. Layer 2 switching does not, in general, involve changing the MAC frame. However, there are situations when switches change the MAC frame. The IEEE 802.1Q Committee is working on a VLAN standard that involves "tagging" a MAC frame with the VLAN it belongs to; this tagging process involves changing the MAC frame. Bridging technology also involves the *Spanning-Tree Protocol*. This is required in a multibridge network to avoid loops. The same principles also apply towards Layer 2 switches, and most commercial Layer 2 switches support the Spanning-Tree Protocol.

The previous discussion provides an outline of Layer 2 switching functions. Layer 2 switching is MAC frame based, does not involve altering the MAC frame, in general, and provides transparent switching in parallel with MAC frames. Since these switches operate at Layer 2, they are protocol independent. However, Layer 2 switching does not scale well because of broadcasts. Although VLANs alleviate this problem to some extent, there is definitely a need for machines on different VLANs to communicate. One example is the situation where an organization has multiple intranet servers on separate subnets (and hence VLANs), causing a lot of intersubnet traffic. In such cases, use of a router is unavoidable; Layer 3 switches enter at this point.

Layer 3 Switches

Layer 3 switching is a relatively new term, which has been "extended" by a numerous vendors to describe their products. For example, one school uses this term to describe fast IP routing via hardware, while another school uses it to describe *Multi Protocol Over ATM* (MPOA). For the purpose of this discussion, Layer 3 switches are superfast routers that do Layer 3 forwarding in hardware. In this article, we will mainly discuss Layer 3 switching in the context of fast IP routing, with a brief discussion of the other areas of application.

Evolution

Consider the Layer 2 switching context shown in Figure 1. Layer 2 switches operate well when there is very little traffic between VLANs. Such VLAN traffic would entail a router—either "hanging off" one of the ports as a one-armed router or present internally within the switch. To augment Layer 2 functionality, we need a router—which

leads to loss of performance since routers are typically slower than switches. This scenario leads to the question: Why not implement a router in the switch itself, as discussed in the previous section, and do the forwarding in hardware?

Although this setup is possible, it has one limitation: Layer 2 switches need to operate only on the Ethernet MAC frame. This scenario in turn leads to a well-defined forwarding algorithm which can be implemented in hardware. The algorithm cannot be extended easily to Layer 3 protocols because there are multiple Layer 3 routable protocols such as IP, IPX, AppleTalk, and so on; and second, the forwarding decision in such protocols is typically more complicated than Layer 2 forwarding decisions.

What is the engineering compromise? Because IP is the most common among all Layer 3 protocols today, most of the Layer 3 switches today perform IP switching at the hardware level and forward the other protocols at Layer 2 (that is, bridge them). The second issue of complicated Layer 3 forwarding decisions is best illustrated by IP option processing, which typically causes the length of the IP header to vary, complicating the building of a hardware forwarding engine. However, a large number of IP packets do not include IP options—so, it may be overkill to design this processing into silicon. The compromise is that the most common (fast path) forwarding decision is designed into silicon, whereas the others are handled typically by a CPU on the Layer 3 switch.

To summarize, Layer 3 switches are routers with fast forwarding done via hardware. IP forwarding typically involves a route lookup, decrementing the *Time To Live* (TTL) count and recalculating the checksum, and forwarding the frame with the appropriate MAC header to the correct output port. Lookups can be done in hardware, as can the decrementing of the TTL and the recalculation of the checksum. The routers run routing protocols such as *Open Shortest Path First* (OSPF) or *Routing Information Protocol* (RIP) to communicate with other Layer 3 switches or routers and build their routing tables. These routing tables are looked up to determine the route for an incoming packet.

Combined Layer 2/Layer 3 Switches

We have implicitly assumed that Layer 3 switches also provide Layer 2 switching functionality, but this assumption does not always hold true. Layer 3 switches can act like traditional routers hanging off multiple Layer 2 switches and provide inter-VLAN connectivity. In such cases, there is no Layer 2 functionality required in these switches. This concept can be illustrated by extending the topology in Figure 1—consider placing a pure Layer 3 switch between the Layer 2 Switch and the router. The Layer 3 Switch would off-load the router from inter-VLAN processing.

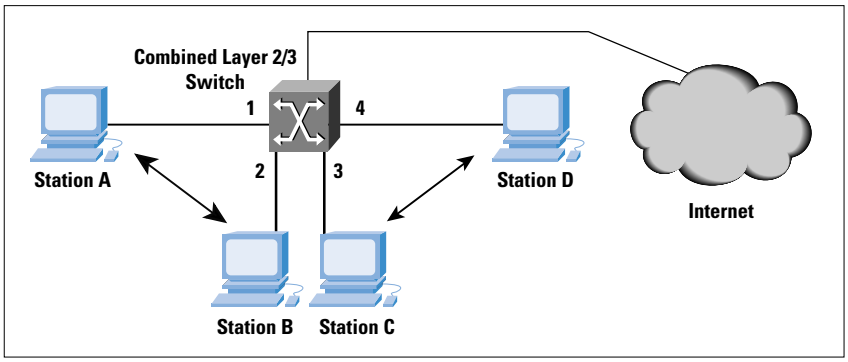


Figure 2: Combined Layer2/ Layer3 Switch connecting directly to the Internet

Figure 2 illustrates the combined Layer 2/Layer 3 switching functionality. The combined Layer 2/Layer 3 switch replaces the traditional router also. A and B belong to IP subnet 1, while C and D belong to IP subnet 2. Since the switch in consideration is a Layer 2 switch also, it switches traffic between A and B at Layer 2. Now consider the situation when A wishes to communicate with C. A sends the IP packet addressed to the MAC address of the Layer 3 switch, but with an IP destination address equal to C's IP address. The Layer 3 switch strips out the MAC header and switches the frame to C after performing the lookup, decrementing the TTL, recalculating the checksum and inserting C's MAC address in the destination MAC address field. All of these steps are done in hardware at very high speeds.

Now how does the switch know that C's IP destination address is Port 3? When it performs learning at Layer 2, it only knows C's MAC address. There are multiple ways to solve this problem. The switch can perform an *Address Resolution Protocol* (ARP) lookup on all the IP subnet 2 ports for C's MAC address and determine C's IP-to-MAC mapping and the port on which C lies. The other method is for the switch to determine C's IP-to-MAC mapping by snooping into the IP header on reception of a MAC frame.

Characteristics

Configuration of the Layer 3 switches is an important issue. When the Layer 3 switches also perform Layer 2 switching, they learn the MAC addresses on the ports—the only configuration required is the VLAN configuration. For Layer 3 switching, the switches can be configured with the ports corresponding to each of the subnets or they can perform IP address learning. This process involves snooping into the IP header of the MAC frames and determining the subnet on that port from the source IP address. When the Layer 3 switch acts like a one-armed router for a Layer 2 switch, the same port may consist of multiple IP subnets.

Management of the Layer 3 switches is typically done via SNMP. Layer 3 switches also have MAC addresses for their ports—this setup can be one per port, or all ports can use the same MAC address. The Layer 3 switches typically use this MAC address for SNMP, Telnet, and Web management communication.

Conceptually, the ATM Forum's *LAN Emulation* (LANE) specification is closer to the Layer 2 switching model, while MPOA is closer to the Layer 3 switching model. Numerous Layer 2 switches are equipped with ATM interfaces and provide a LANE client function on that ATM interface. This scenario allows the bridging of MAC frames across an ATM network from switch to switch. The MPOA is closer to combined Layer2/Layer 3 switching, though the MPOA client does not have any routing protocols running on it. (Routing is left to the MPOA server under the Virtual Router model.)

Do Layer 3 switches completely eliminate need for the traditional router ? No, routers are still needed, especially where connections to the wide area are required. Layer 3 switches may still connect to such routers to learn their tables and route packets to them when these packets need to be sent over the WAN. The switches will be very effective on the workgroup and the backbone within an enterprise, but most likely will not replace the router at the edge of the WAN (read Internet in many cases). Routers perform numerous other functions like filtering with access lists, inter-Autonomous System (AS) routing with protocols such as the *Border Gateway Protocol* (BGP), and so on. Some Layer 3 switches may completely replace the need for a router if they can provide all these functions (see Figure 2).

References

[1] *Computer Networks*, 3rd Edition, Andrew S. Tanenbaum, ISBN 0-13-349945-6, Prentice-Hall, 1996.

[2] *Interconnections: Bridges and Routers*, Radia Perlman, ISBN 0-201-56332-0, Addison-Wesley, 1992.

[3] "MAC Bridges," ISO/IEC 10038, ANSI/IEEE Standard 802.1 D-1993.

[4] "Draft Standard for Virtual Bridged Local Area Networks," IEEE P802.1Q/D6, May 1997.

[5] "Internet Protocol," Jon Postel, RFC 791, 1981.

[6] "Requirements for IP Version 4 Routers," Fred Baker, RFC 1812, June 1995.

[7] "LAN Emulation over ATM Version 1.0," **af-lane-0021.000**, The ATM Forum, January 1995.

[8] "Multiprotocol over ATM (MPOA) Specication Version 1.0" **af-mpoa-0087.000**, The ATM Forum, July 1997.

THAYUMANAVAN SRIDHAR is Director of Engineering at Future Communications Software in Santa Clara, CA. He received his BE in Electronics and Communications Engineering from the College of Engineering, Guindy, Anna University, Madras, India, his Master of Science in Electrical and Computer Engineering from the University of Texas at Austin. He can be reached at **sridhar@futsoft.com**

Book Review

Gigabit Ethernet

Gigabit Ethernet: Technology and Applications for High-Speed LANs, by Rich Seifert, ISBN 0-201-18553-9, Addison-Wesley, 1998, `http://www.awl.com/cseng/titles/0-201-18553-9`.

Gigabit Ethernet is storming its way onto the high-speed LAN scene. From a concept in 1984 to an emerging commercial reality in 1998, Gigabit Ethernet promises to give other high-speed LAN technologies, especially ATM, a serious run for their money. Capitalizing on the basic ease of use and deployment that has made other forms of Ethernet the most popular LAN technology of all, Gigabit Ethernet promises to add major bandwidth to such networks in a straightforward, completely compatible, and relatively affordable way. This book performs an excellent survey of the technologies, algorithms, and design principles that make Gigabit Ethernet possible, and also explains where the tremendous appeal of Gigabit Ethernet really lies. Much of the book is devoted to explaining Ethernet principles and operation in general, as well as exploring recent developments that have enabled gigabit technologies to emerge.

Organization

The book is divided into three parts. Part I explores the foundations that underpin Gigabit Ethernet, starting with a brief but cogent exploration of Ethernet before gigabit versions loomed on the horizon. The rest of Part I covers the trends in LAN usage in general, and Ethernet in particular, that laid the groundwork for Gigabit Ethernet. These trends include the move from shared media to dedicated media on many LANs, and likewise from shared LANs to dedicated LANs, and the concomitant deployment of full-duplex technologies to support bidirectional, high-bandwidth communications. Seifert, an original member of the DIX (Digital-Intel-Xerox) team that developed Ethernet, writes clearly and compellingly about complex issues, such as flow control, medium independence, and automatic configuration, as he explains what made Gigabit Ethernet possible, if not inevitable.

In Part II, Seifert turns his focus onto Gigabit Ethernet itself, beginning with an overview. In the rest of Part II, he explains how *Media Access Control* (MAC) works for half-duplex and full-duplex versions of Gigabit Ethernet, and makes a strong case for the essential irrelevancy of shared-media and half-duplex operation for Gigabit Ethernet. Along the way, Seifert also covers how Gigabit Ethernet networking devices, such as repeaters and switching and routing hubs, must be designed and how they work, and covers the behavior and operation of the physical layer at gigabit speeds.

He concludes this section of the book with a brief overview of the current IEEE Draft 802.3z specification that governs current Gigabit Ethernet operations, and mentions ongoing work in the 802.3ab subcommittee to define a workable implementation for Gigabit Ethernet on twisted-pair media (1000BaseT, as it will probably be known).

In Part III, Seifert tackles some of the most interesting material in this book. He begins with a discussion of how LANs and computers change roles over time in acting as the bottleneck for network use. The point here is that because of its extremely high bandwidth relative to the demands of most applications and end-user requirements, Gigabit Ethernet is likely to remain a backbone or clustering technology for the foreseeable future. He also explores the performance considerations for both networks and applications involved when extreme speeds or excessive bandwidths are available, to point out how bandwidth aggregation is presently Gigabit's most immediate and compelling contribution to networking.

Finally, he explores how Gigabit Ethernet compares to other high-speed networking technologies, including Fast Ethernet, *Fiber Distributed Data Interface* (FDDI), *High-Performance Parallel Interface* (HIPPI), *Fibre Channel*, and ATM. His discussion of why both ATM and Gigabit Ethernet are necessary, and why neither can fully supplant the other, represents a humorous and insightful analysis of why connection-oriented and connectionless communications and applications are both good, and why the two can never truly converge.

An Outstanding Contribution

A rundown of Seifert's layout and content, however, fails to do complete justice to this book. For one thing, Seifert's work includes the funniest and most ingenious footnotes I've seen in recent publications, including some truly horrendous puns and some downright howlers. For example, when discussing how repeaters work, he comments that "A jabbering station causes carrier sense to be continuously asserted and blocks all use of a shared LAN. A repeater looks for this condition and isolates the offending station." To this last sentence, he appends the following footnote: "Research is underway to determine if this mechanism can be extended for use on politicians and university lecturers." And this is just one of dozens of such gems that help to relieve the dryness that deeply technical material can sometimes manifest.

This book is also masterful simply because the author understands his material so well, and does such an outstanding job of explaining and exploring even the most abstruse networking concepts. Although I've been working with Ethernet for 15 years, I learned a great deal of new material from Part I of the book because old concepts were explained in new ways that improved my understanding. I suspect other readers will have one or two "Aha!" experiences from this tome as well.

But it's when making the case for full-duplex Gigabit Ethernet and exploring the requirements for switching and routing behaviors in Gigabit Ethernet networking devices that this material really shines.

Without a doubt, this book is among the very best of any of the literature available on high-speed networking today. I give it an A+ rating, not only because of the breadth and depth of its technical coverage and its compilation of essential concepts and information, but also because the author's deep understanding of networking protocols and communications needs enlivens all of his discussions of matters technical, business, and political. If you want to understand Gigabit Ethernet, this book is the obvious place to begin (and for many, to end) your search for enlightenment.

But even if all you want is a good read about expensive, exotic, and high-performance technology, Seifert's book offers the opportunity for outright enjoyment of the prose, and shared delight at untangling the technical dilemmas that any good design engineer must unravel on the road between a set of requirements and working implementation thereof.

—Ed Tittel
LANWrights, Inc.
etittel@lanw.com

More Book Reviews

We have more book reviews awaiting publication:

- *Internet Cryptography*, by Richard E. Smith, ISBN 0-201-92480-3, Addison-Wesley, 1998. Reviewed by Fred Avolio.
- *Web Security: A Step-by-Step Reference Guide*, by Lincoln D. Stein, ISBN 0-201-63489-9, Addison-Wesley, December 1997. Reviewed by Richard Perlman
- *IP Multicasting: The Complete Guide to Interactive Corporate Networks*, by Dave Kosiur ISBN 0-471-24359-0, Wiley Computer Publishing, 1998. Reviewed by Neophytos Iacovou.

So, make sure you receive the next issue of *The Internet Protocol Journal* due out in December 1998.

Fragments

More on The Future of the Domain Name System (DNS)

Shortly after our first issue went to press, the US Government issued a so-called White Paper as a follow on to the Green Paper. The White Paper, entitled "Management of Internet Names and Addresses," can be found at:

http://www.ntia.doc.gov/ntiahome/domainname/domainhome.htm

In early July, *The International Forum on The White Paper* (IFWP) was formed. The IFWP is "an ad hoc coalition of professional, trade and educational associations representing a diversity of Internet stakeholder groups." The IFWP held a series of meetings in Reston, Brussels, Geneva, Singapore and Buenos Aires to discuss the White Paper, specifically the incorporation of the *Internet Assigned Numbers Authority* (IANA). For more information on the IFWP process, see: **http://www.ifwp.org**

The IANA has posted draft bylaws for its incorporation on the IANA web site at: **http://www.iana.org**, and asked for community input. By the time you read this, the incorporation should already have taken place. We will provide an update in our next issue.

IETF Wins Award

The *Computer Professionals for Social Responsibility* (CPSR) has chosen the *Internet Engineering Task Force* (IETF) to be honored with the Norbert Wiener award for the group's influential role in the evolution of the Internet. In its 12-year history, this is only the second time the CPSR has recognized an organization rather than an individual. The IETF will accept the award at CPSR's annual conference, on Saturday evening, October 10, 1998, in Boston. The IETF is noted for its highly open and democratic processes that have affected the development of the Internet. The CPSR believes that such open processes are both extremely important and seriously threatened, and have accordingly made Internet governance the focus of its 1998 program year. The Norbert Wiener award was established in 1987 by the CPSR in memory of the originator of the field of cybernetics, whose pioneering work was one of the pillars on which the computer technology was created. See: **http://www.cpsr.org** and **http://www.ietf.org**

Send us your comments!

We look forward to hearing your comments and suggestions regarding anything you read in this publication. Send us e-mail at: **ipj@cisco.com**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Cisco News Publications Group, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-J4
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

December 1998 Volume 1, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download previous issues of IPJ in PDF format from:
www.cisco.com/ipj

From The Editor

The *Simple Network Management Protocol* (SNMP) was first standardized in 1988. It quickly became a de facto management standard, not only for Internet technologies, but for a wide range of applications. Like many early Internet protocols, the first two versions of SNMP did not include provisions for security. In 1996, two different proposals for security enhancements to SNMPv2 were put forward, with strong proponents behind each. Everyone agreed that the industry needed just *one* solution, and therefore work proceeded to incorporate the best features of the two security proposals for SNMPv2. The result is SNMPv3, and it is described in this issue by William Stallings.

As the Internet continues to grow, demand for high-speed access for residential users is increasing. Alternatives to traditional dialup service include *Digital Subscriber Line* (DSL) services, wireless solutions, and various television technologies. In this issue, we examine two aspects of Internet access using TV technologies. First, Mark Laubach gives an overview of cable modem technologies and standards, and discusses some deployment issues. In the second article, George Abe looks at the emerging digital television standards and how they could be used to provide Internet access.

The Internet lost one of its most respected pioneers when Jon Postel passed away on October 16, 1998. Jon was well-known as the Director of the *Internet Assigned Numbers Authority* (IANA) and as the editor of the *Request for Comments* (RFC) document series. Included in this issue is "I Remember IANA," a tribute to Jon Postel written by his longtime friend Vint Cerf. The remembrance has also been published as RFC 2468.

With that we have come to the end of 1998 and the end of Volume 1 of *The Internet Protocol Journal*. We wish you a pleasant holiday season and will be back with Volume 2, Number 1 in March 1999. In the meantime, please visit our Web site at **www.cisco.com/ipj**. There you will find back issues in PDF format, our Call for Papers and guidelines for authors of IPJ articles.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Security Comes to SNMP: The New SNMPv3 Proposed Internet Standards

by William Stallings

Data networks typically include bridges, routers, links into WANs, and end-user equipment from multiple vendors. Users need automated tools to help manage such configurations that are easy to install, easy to use, and don't place a great burden on the network.

This accounts for the popularity of the *Simple Network Management Protocol* (SNMP). Introduced in 1988 to provide management capability for TCP/IP-based networks, SNMP rapidly became the most widely used standardized network management tool. Virtually all vendors of network-based equipment provide SNMP.

The appeal of SNMP has indeed been its simplicity because SNMP provides a bare-bones set of functions, and it is indeed easy to implement, install, and use. And, used sensibly, it will not place undue burden on the network. Moreover, because of its simplicity, achievement of interoperability is a relatively straightforward task: SNMP modules from different vendors can be made to work together with minimal effort.

SNMP—Strengths and Weaknesses

SNMP is based on three concepts: *managers, agents,* and the *Management Information Base* (MIB). In any configuration, at least one manager node runs SNMP management software. Network devices to be managed, such as bridges, routers, servers, and workstations, are equipped with an agent software module. The agent is responsible for providing access to a local MIB of objects that reflects the resources and activity at its node. The agent also responds to manager commands to retrieve values from the MIB and to set values in the MIB. An example of an object that can be retrieved is a counter that keeps track of the number of packets sent and received over a link into the node; the manager can track this value to monitor the load at that point in the network. An example of an object that can be set is one that represents the state of a link; the manager could disable the link by setting the value of the corresponding object to the disabled state.

Such capabilities are fine for implementing a basic network-management system. To enhance this basic functionality, a new version of SNMP was introduced in 1993 and revised in 1996. SNMPv2 added bulk transfer capability and other functional extensions. However, neither SNMPv1 nor SNMPv2 offers security features. Specifically, SNMPv1/v2 can neither authenticate the source of a management message nor provide encryption. Without authentication, it is possible for nonauthorized users to exercise SNMP network management functions. It is also possible for nonauthorized users to eavesdrop on

management information as it passes from managed systems to the management system. Because of these deficiencies, many SNMPv1/v2 implementations are limited to simply a read-only capability, reducing their utility to that of a network monitor; no network control applications can be supported.

Enter SNMPv3

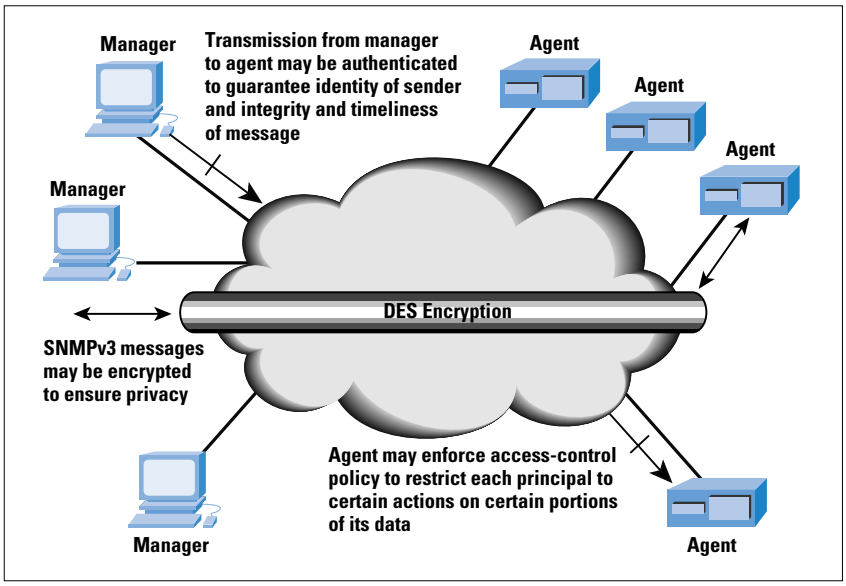
To correct the security deficiencies of SNMPv1/v2, SNMPv3 was issued as a set of Proposed Standards in January 1998 (Table 1). This set of documents does not provide a complete SNMP capability but rather defines an overall SNMP architecture and a set of security capabilities. These are intended to be used with the existing SNMPv2. As one of the SNMPv3 working documents puts it, "SNMPv3 is SNMPv2 plus administration and security."

Table 1: SNMPv3 RFCs

RFC Number	Title
2271	An Architecture for Describing SNMP Management Frameworks
2272	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)
2273	SNMPv3 Applications
2274	User-Based Security Model for SNMPv3
2275	View-Based Access Control Model (VACM) for SNMP

SNMPv3 includes three important services: *authentication, privacy,* and *access control* (Figure 1). To deliver these services in a flexible and efficient manner, SNMPv3 introduces the concept of a *principal*, which is the entity on whose behalf services are provided or processing takes place. A principal can be an individual acting in a particular role; a set of individuals, with each acting in a particular role; an application or set of applications; or combinations thereof. In essence, a principal operates from a management station and issues SNMP commands to agent systems. The identity of the principal and the target agent together determine the security features that will be invoked, including authentication, privacy, and access control. The use of principals allows security policies to be tailored to the specific principal, agent, and information exchange, and gives human security managers considerable flexibility in assigning network authorization to users.

Figure 1: SNMPv3 Security Features



SNMPv3 is defined in a modular fashion, as shown in Figure 2. Each SNMP entity includes a single SNMP *engine*. An SNMP engine implements functions for sending and receiving messages, authenticating and encrypting/decrypting messages, and controlling access to managed objects. These functions are provided as services to one or more applications that are configured with the SNMP engine to form an SNMP *entity*. This modular architecture provides several advantages. First, the role of an SNMP entity is determined by the modules that are implemented in that entity. For example, a certain set of modules is required for an SNMP agent, whereas a different (though overlapping) set of modules is required for an SNMP manager. Second, the modular structure of the specification lends itself to defining different versions of each module. This, in turn, makes it possible to (1) define alternative or enhanced capabilities for certain aspects of SNMP without needing to go to a new version of the entire standard (for example, SNMPv4), and (2) clearly specify coexistence and transition strategies.

Figure 2: SNMP Entity (RFC 2271)

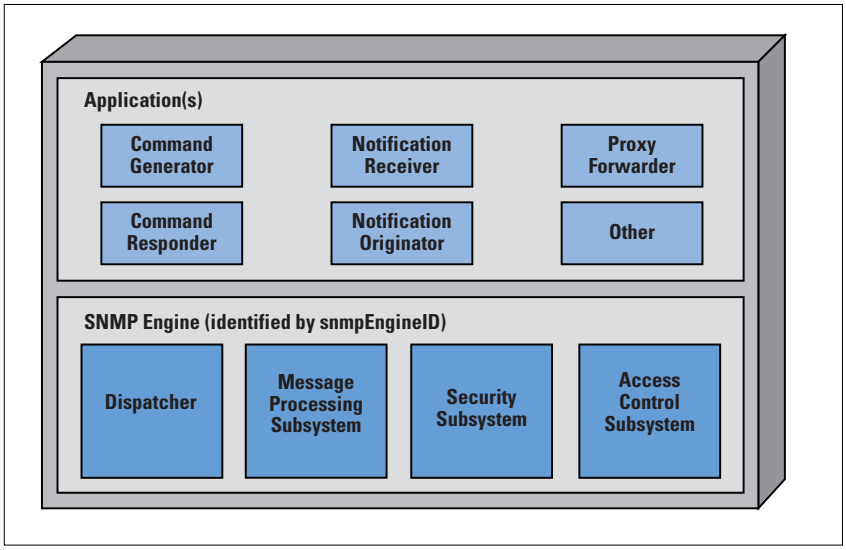


Table 2 provides a brief definition of each module.

Table 2: Components of an SNMP Entity (RFC 2271 and 2273)

Dispatcher	Allows for concurrent support of multiple versions of SNMP messages in the SNMP engine. It is responsible for (1) accepting protocol data units (PDUs) from applications for transmission over the network and delivering incoming PDUs to applications; (2) passing outgoing PDUs to the Message Processing Subsystem to prepare as messages, and passing incoming messages to the Message Processing Subsystem to extract the incoming PDUs; and (3) sending and receiving SNMP messages over the network.
Message Processing Subsystem	Responsible for preparing messages for sending and for extracting data from received messages.
Security Subsystem	Provides security services such as the authentication and privacy of messages. This subsystem potentially contains multiple Security Models.
Access Control Subsystem	Provides a set of authorization services that an application can use for checking access rights. Access control can be invoked for retrieval or modification request operations and for notification generation operations.
Command Generator	Initiates SNMP Get, GetNext, GetBulk, or Set request PDUs and processes the response to a request that it has generated.
Command Responder	Receives SNMP Get, GetNext, GetBulk, or Set request PDUs destined for the local system as indicated by the fact that the contextEngineID in the received request is equal to that of the local engine through which the request was received. The command responder application performs the appropriate protocol operation, using access control, and generates a response message to be sent to the originator of the request.
Notification Originator	Monitors a system for particular events or conditions, and generates Trap or Inform messages based on these events or conditions. A notification originator must have a mechanism for determining where to send messages, and which SNMP version and security parameters to use when sending messages.
Notification Receiver	Listens for notification messages, and generates response messages when a message containing an Inform PDU is received.
Proxy Forwarder	Forwards SNMP messages. Implementation of a proxy forwarder application is optional.

SNMPv3 Message Processing

SNMPv3 relies on the *User Datagram Protocol* (UDP) or some other transport-layer protocol to convey SNMP information. Above the UDP layer, SNMP functionality is organized into two application-level layers: a PDU processing layer and a message processing layer.

The topmost layer is the PDU processing layer. At this layer, management commands (such as Get, Set, Trap, Inform) are realized in a PDU that includes an indication of the command type and a list of variables (management objects) to which the command refers. This PDU is then passed down to the message processing layer, which adds a message header. The message header contains security-related information that may be used for authentication and privacy operations.

Figure 3 illustrates the message structure. The first five fields are generated by the message processing model on outgoing messages and processed by the message processing model on incoming messages. The next six fields show security parameters used by the security model, which is invoked by the message processing model to provide security services. Finally, the PDU, together with the contextEngineID and contextName, constitute a scoped PDU, used for PDU processing.

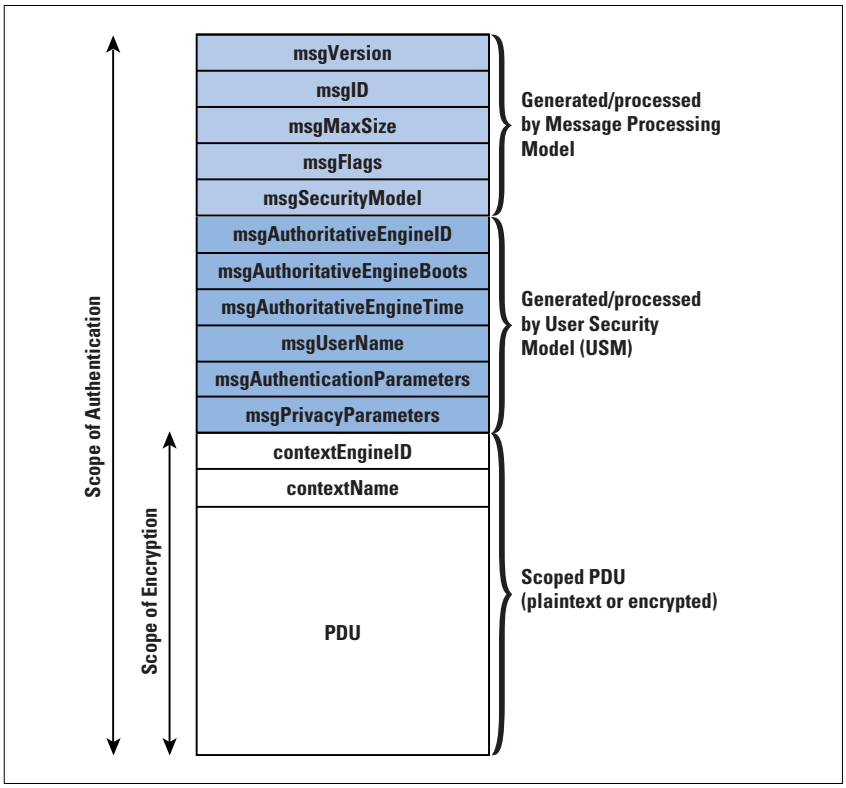


Figure 3: SNMPv3 Message Format with User-Based Security Model

The first five fields follow:

- *msgVersion:* Set to snmpv3(3).
- *msgID:* A unique identifier used between two SNMP entities to coordinate request and response messages, and by the message processor to coordinate the processing of the message by different subsystem models within the architecture. The range of this ID is 0 through 2^{31}–1.

- *msgMaxSize:* Conveys the maximum size of a message in octets supported by the sender of the message, with a range of 484 through 2^{31}–1. This is the maximum segment size that the sender can accept from another SNMP engine (whether a response or some other message type).

- *msgFlags:* An octet string containing three flags in the least significant three bits: reportableFlag, privFlag, authFlag. If reportableFlag = 1, then a Report PDU must be returned to the sender under those conditions that can cause the generation of a Report PDU; when the flag is zero, a Report PDU may not be sent. The reportableFlag is set to 1 by the sender in all messages containing a request (Get, Set) or an Inform, and set to 0 for messages containing a Response, a Trap, or a Report PDU. The reportableFlag is a secondary aid in determining when to send a Report. It is used only in cases in which the PDU portion of the message cannot be decoded (for example, when decryption fails because of incorrect key). The privFlag and authFlag are set by the sender to indicate the security level that was applied to the message. For privFlag = 1, encryption was applied and for privFlag = 0, authentication was applied. All combinations are allowed except (privFlag = 1 AND authFlag = 0); that is, encryption without authentication is not allowed.

- *msgSecurityModel:* An identifier in the range of 0 through 2^{31}–1 that indicates which security model was used by the sender to prepare this message and, therefore, which security model must be used by the receiver to process this message. Reserved values include 1 for SNMPv1, 2 for SNMPv2c, and 3 for SNMPv3.

User-Based Security Model

The *User-Based Security Model* (USM) uses the concept of an authoritative engine. In any message transmission, one of the two entities, transmitter or receiver, is designated as the authoritative SNMP engine, according to the following rules:

- When an SNMP message contains a payload that expects a response (for example, a Get, GetNext, GetBulk, Set, or Inform PDU), then the receiver of such messages is authoritative.

- When an SNMP message contains a payload that does not expect a response (for example, an SNMPv2-Trap, Response, or Report PDU), then the sender of such a message is authoritative.

Thus, for messages sent on behalf of a Command Generator and for Inform messages from a Notification Originator, the receiver is authoritative. For messages sent on behalf of a Command Responder or for Trap messages from a Notification Originator, the sender is authoritative. This designation serves two purposes:

- The timeliness of a message is determined with respect to a clock maintained by the authoritative engine. When an authoritative engine sends a message (Trap, Response, Report), it contains the current value of its clock, so that the nonauthoritative recipient can synchronize on that clock. When a nonauthoritative engine sends a message (Get, GetNext, GetBulk, Set, Inform), it includes its current estimate of the time value at the destination, allowing the destination to assess the timeliness of the message.

- A key localization process, described later, enables a single principal to own keys stored in multiple engines; these keys are localized to the authoritative engine in such a way that the principal is responsible for a single key but avoids the security risk of storing multiple copies of the same key in a distributed network.

When an outgoing message is passed to the USM by the Message Processor, the USM fills in the security-related parameters in the message header. When an incoming message is passed to the USM by the Message Processor, the USM processes the values contained in those fields. The security-related parameters include the following:

- *msgAuthoritativeEngineID:* The snmpEngineID of the authoritative SNMP engine involved in the exchange of this message. Thus, this value refers to the source for a Trap, Response, or Report, and to the destination for a Get, GetNext, GetBulk, Set, or Inform.

- *msgAuthoritativeEngineBoots:* The snmpEngineBoots value of the authoritative SNMP engine involved in the exchange of this message. The object snmpEngineBoots is an integer in the range 0 through $2^{31}-1$ that represents the number of times that this SNMP engine has initialized or reinitialized itself since its initial configuration.

- *msgAuthoritativeEngineTime:* The snmpEngineTime value of the authoritative SNMP engine involved in the exchange of this message. The object snmpEngineTime is an integer in the 0 through $2^{31}-1$ range that represents the number of seconds since this authoritative SNMP engine last incremented the snmpEngineBoots object. Each authoritative SNMP engine is responsible for incrementing its own snmpEngineTime value once per second. A non-authoritative engine is responsible for incrementing its notion of snmpEngineTime for each remote authoritative engine with which it communicates.

- *msgUserName:* The user (principal) on whose behalf the message is being exchanged.

- *msgAuthenticationParameters:* Null if authentication is not being used for this exchange; otherwise, this is an authentication parameter. For the current definition of USM, the authentication parameter is a message authentication code generated using an algorithm referred to as HMAC.

- *msgPrivacyParameters:* Null if privacy is not being used for this exchange; otherwise, this is a privacy parameter. For the current definition of USM, the privacy parameter is a parameter used in the encryption algorithm DES.

Secret-Key Authentication

The authentication mechanism in SNMPv3 assures that a received message was, in fact, transmitted by the principal whose identifier appears as the source in the message header. In addition, this mechanism assures that the message was not altered in transit and that it was not artificially delayed or replayed.

To achieve authentication, each pair of principal and remote SNMP engines that wishes to communicate must share a secret authentication key. The sending entity provides authentication by including a message authentication code with the SNMPv3 message it is sending. This code is a function of the contents of the message, the identity of the principal and engine, the time of transmission, and a secret key that should be known only to the sender and the receiver. The secret key must initially be set up outside of SNMPv3 as a configuration function. That is, the configuration manager or network manager is responsible for distributing initial secret keys to be loaded into the databases of the various SNMP managers and agents. This can be done manually or by using some form of secure data transfer outside of SNMPv3. When the receiving entity gets the message, it uses the same secret key to calculate the message authentication code again. If the receiver's version of the code matches the value appended to the incoming message, then the receiver knows that the message can only have originated from the authorized manager, and that the message was not altered in transit. The shared secret key between sending and receiving parties must be preconfigured.

Another aspect of USM authentication is timeliness verification. USM is responsible for assuring that messages arrive within a reasonable time window to protect against message delay and replay attacks. Two functions support this service: synchronization and time-window checking.

Each authoritative engine maintains two values, snmpEngineBoots and snmpEngineTime, that keep track of the number of boots since initialization and the number of seconds since the last boot. These values are placed in outgoing messages in the fields msgAuthoritativeEngineBoots and msgAuthoritativeEngineTime. A nonauthoritative engine maintains synchronization with an authoritative engine by maintaining local copies of snmpEngineBoots and snmpEngineTime for each remote authoritative engine with which it communicates. These values are updated on receipt of an authentic message from the remote authoritative engine. Between these message updates, the nonauthoritative

engine increments the value of snmpEngineTime for the remote authoritative engine to maintain loose synchronization. These values are inserted in outgoing messages intended for that authoritative engine.

When an authoritative engine receives a message, it compares the incoming boot and time values with its own boot and time values. If the boot values match and if the incoming time value is within 150 seconds of the actual time value, then the message is declared to be within the time window and, therefore, to be a timely message.

Privacy Using Conventional Encryption

The SNMPv3 USM privacy facility enables managers and agents to encrypt messages to prevent eavesdropping by third parties. Again, manager entity and agent entity must share a secret key. When privacy is invoked between a principal and a remote engine, all traffic between them is encrypted using the *Data Encryption Standard* (DES). The sending entity encrypts the entire message using the DES algorithm and its secret key, and sends the message to the receiving entity, which decrypts it using the DES algorithm and the same secret key. Again, the two parties must be configured with the shared key.

The *cipher-block-chaining* (CBC) mode of DES is used by USM. This mode requires that an initial value (IV) be used to start the encryption process. The msgPrivacyParameters field in the message header contains a value from which the IV can be derived by both sender and receiver.

View-Based Access Control Model (VACM)

The access control facility makes it possible to configure agents to provide different levels of access to the agent's MIB to different managers. An agent entity can restrict access to its MIB for a particular manager entity in two ways. First, it can restrict access to a certain portion of its MIB. For example, an agent may restrict most manager principals to viewing performance-related statistics and allow only a single designated manager principal to view and update configuration parameters. Second, the agent can limit the operations that a principal can use on that portion of the MIB. For example, a particular manager principal could be limited to read-only access to a portion of an agent's MIB. The access control policy to be used by an agent for each manager must be preconfigured; it essentially consists of a table that details the access privileges of the various authorized managers. Unlike authentication, which is done by user, access control is done by group, where a group may be a set of multiple users.

Figure 4:
VACM Flowchart

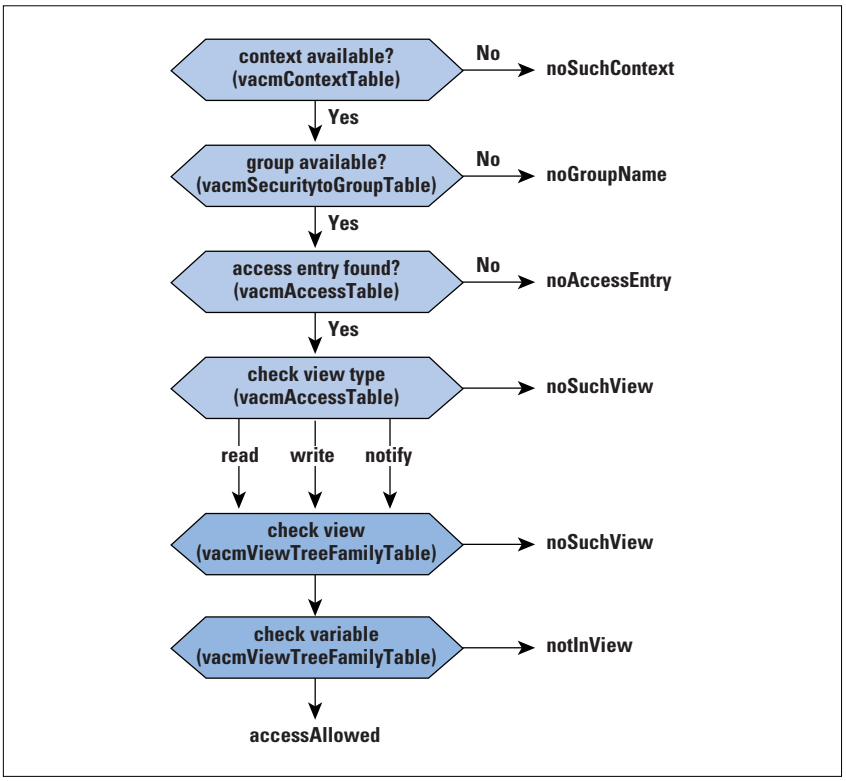


Figure 4 illustrates the overall VACM logic, which proceeds in the following steps:

1. The context name refers to a named subset of the MIB objects at an agent. VACM checks to see if there is an entry in vacmContextTable for the requested contextName. If so, then this context is known to this SNMP engine. If not, then an errorIndication of noSuchContext is returned.

2. Each principal operating under a given security model is assigned to at most one group, and access privileges are configured on a group basis. VACM checks vacmSecurityToGroupTable to determine if there is a group assigned to the requested <securityModel, securityName> pair. If so, then this principal, operating under this securityModel, is a member of a group configured at this SNMP engine. If not, then an errorIndication of noGroupName is returned.

3. VACM next consults the vacmAccessTable with groupName, contextName, securityModel, and securityLevel (indicates authentication, authentication plus privacy, or neither) as indices. If an entry is found, then an access control policy has been defined for this groupName, operating under this securityModel, at this securityLevel, for access to this contextName. If not, then an errorIndication of noAccessEntry is returned.

4. A MIB view is a structure subset of a context; it is essentially a set of managed object instances viewed as a set for access control purposes. VACM determines whether the selected vacmAccessTable entry includes reference to a MIB view of viewType (read, write, notify). If so, then this entry contains a viewName for this combination of groupName, contextName, securityModel, securityLevel, and viewType. If not, then an errorIndication of noSuchView is returned.

5. The viewName from Step 4 is used as an index into vacmViewTreeFamilyTable. If a MIB view is found, then a MIB view has been configured for this viewName. If not, then an errorIndication of noSuchView is returned.

6. VACM checks the variableName against the selected MIB view. If this variable is included in the view, then a statusInformation of accessAllowed is returned. If not, then an errorIndication of notInView is returned.

References

[0] The SNMPv3 RFCs, see Table 1 above.

[1] J. D. Case, M. Fedor, M. L. Schoffstall, and C. Davin, "Simple Network Management Protocol," RFC 1157, May 1990.

[2] Rose, M. T., *The Simple Book: An Introduction to Networking Management,* Revised Second Edition, Prentice-Hall, ISBN 0-13-451659-1, 1996.

[3] Waters, G., Editor, "User-based Security Model for SNMPv2," RFC 1910, February 1996.

[4] *ConneXions—The Interoperability Report,* Volume 10, No. 5, May 1996—Special Issue: "Network Management Today."

WILLIAM STALLINGS is a consultant, lecturer, and author of over a dozen books on data communications and computer networking. He has a PhD in computer science from M.I.T. This article is based on material in the author's latest book: *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2,* Second Edition (Addison Wesley, 1998). His home in cyberspace is **`http://www.shore.net/~ws`** and he can be reached at **`ws@shore.net`**

Residential Area CATV Broadband Internet Technology: Current Status

by Mark Laubach, Com21, Inc.

Cable modem technology has entered commonplace discussion and is in the early stages of widespread deployment throughout the world. The capabilities provided by cable modems promise data bandwidth speeds far in excess of those provided by traditional telephone modem services. In North America the race is on between cable operators deploying services based on standardized cable modems and telephone companies deploying *Digital Subscriber Line* (DSL) services. Internet Service Providers (ISPs) are taking position to promote any method of delivering Internet services to and from the home and are helping to fuel the race. Initially these services will only provide higher-speed Internet access and improved access to major information services (for example, AOL). Cable modem service offerings promote higher than DSL speed to the subscriber and a promise that packet voice services will be available in 1999.

As an introduction to some of the issues surrounding cable modem technology, this article summarizes two of the standardization efforts: the IEEE 802.14 Cable TV Media Access Control and Physical Protocol working group and the North American Data Over Cable Service Interface Specification. Delivering a viable Internet service to a cable TV reached subscriber community has its own set of deployment issues that are briefly reviewed and summarized.

Background

Networks based on packet technology were first presented in 1964[1]. Since then, and through numerous evolutionary steps, the Internet as we know it today was brought into existence. Today, packets are transmitted over most any media. The next economic and technical frontier is the mass deployment of moving packets over cable television (CATV) networks for serving the Internet to every home. There are several link layer approaches for delivering IP datagrams via cable modems. The always present debate of whether to use fixed or variable length packets continues in the cable modem world. This article presents overviews of two variations of cable modem protocols: first, the concept of sending small, fixed-sized packets over the CATV plant using 53-octet *Asynchronous Transfer Mode* (ATM) cells[2], as is being defined in the public standards process of the IEEE 802.14 working group; and secondly, by sending variable-length packets (IP over Ethernet) as defined by the *Multimedia Cable Network System* (MCNS) *Data Over Cable Service Interface Specification* (DOCSIS) for the North American cable industry[3]. As widely accepted standards normally motivate industrial focus and subsequent cost reduction due to vendor competitive pressures, there is an additional drive provided by North American cable operators to get the cost of the cable modem off their books and into retail channels.

The IEEE 802.14 Cable TV MAC and PHY Protocol working group is chartered with providing a single *Media Access Control* (MAC) and multiple physical sublayer (PHY) standard for cable TV networks. The efforts of 802.14 must support IEEE 802 layer services (including Ethernet) and must also be ATM compatible.

The DOCSIS specifications are managed by CableLabs on behalf of its cable television system operator members. The project was initiated by an organization called *Multimedia Cable Network System* (MCNS) Partners, L. P., which consists of Comcast Cable Communications, Cox Communications, Tele-Communications, Inc., and Time Warner Cable. In addition to MCNS, Rogers Cablesystems Limited, MediaOne, and CableLabs have all contributed to the DOCSIS documents, as have several networking and telecommunications vendors. DOCSIS documents describe the internal and external network interfaces for a system that allows bidirectional transfer of IP traffic, between the cable system head-end and customer premises, over a cable television system[4].

The customer network interface in common use today is Ethernet 10BaseT. There is a mandate for a 10 Mbps Ethernet interface in the home. Subscriber access equipment can be a personal computer, X-Terminal, or any such device that supports the TCP/IP protocol suite. Future home interfaces from the cable modem will include the *Universal Serial Bus* (USB) and IEEE 1394 (*FireWire*).

IP Over CATV System Challenges

From an IP perspective, a CATV system almost appears to be another data link layer. However, experience gained thus far has demonstrated that the marriage of IP over CATV radio frequency (RF) channels is not as straightforward as IP over any other high-speed serial point-to-point link.

In the CATV space, the downstream channels in a cable plant (cable head-end to subscribers) is a point-to-multipoint channel. This does have very similar characteristics to transmitting over an Ethernet segment where one transmitter is being listened to by many receivers. The major difference is that baseband modulation has been replaced by a more densely modulated RF carrier with very sophisticated adaptive signal processing and *forward error correction* (FEC).

In the upstream direction (subscriber cable modems transmitting towards the head-end) the environment is many transmitters and one receiver. This introduces the need for precise scheduling of packet transmissions to achieve high utilization and precise power control so as to not overdrive the receiver or other amplifier electronics in the cable system. Since the upstream direction is like a single receiver with many antennas, the channels are much much more susceptible to interfering noise products[5, 6]. In the cable industry, we generally

call this *ingress noise*. As ingress noise is an inherent part of CATV plants, the observable impact is an unfortunate rise in the average noise floor in the upstream channel. To overcome this noise jungle, upstream modulation is not as dense as in the downstream and we have to use more effective FEC as used in the downstream. There is a further complication that there are many upstream "ports" on a fully deployed *Hybrid Fiber-Coaxial* (HFC) plant that requires matching head-end equipment ports for high-speed data[7].

To further the rub on the upstream channel use, the arcane regulations of the FCC from back in the mid 1980s mandated that upstream frequency spectrum be reserved on all cable plants, regardless of whether it was actually used. This was typically the 5–42 MHz region, leaving above 50 MHz for downstream transmissions. (Note that there are other regions available for upstream, but the overwhelming majority of cable plants only use 5–42 MHz.) This leaves precious little spectral bandwidth for upstream communications.

The existing environment for high-speed data protocols therefore provides for relatively clean bandwidth in the downstream direction, allowing for higher-speed data rate channels, while in the upstream, individual channels are of lesser data rate. However, multiple upstream channels can be used per downstream channel to get effective symmetric aggregate bandwidth. Typically, we speak of cable modem systems as providing asymmetric services (higher downstream data rate than upstream). Note though that this asymmetry closely matches what we expect initially for residential high-speed data services. That is, many more subscribers at home pulling things off the Internet via web services, than pushing data back in.

Modern modulation techniques provide for a range of data carrying capability ("baud rate"). A low order modulation rate called *Quadrature Phase Shift Keying* (QPSK) provides for two data bits per symbol encoding. *Quadrature Amplitude Modulation* (QAM) provides a lower order modulation of 16 QAM (four bits per symbol) through higher order rates of 64 QAM (six bits per symbol) and 254 QAM (eight bits per symbol). Low order modulations are more robust in higher average noise environments. Higher order modulations are least robust. Therefore, high order modulations are suitable for downstream channels due to the low noise performance, while the order of upstream channel modulation is heavily effected by noise. Typically, cable modem systems will see QPSK used for upstream channels. When the plant is very clean, noise-wise, 16 QAM may be used.

One additional challenge is that the speed of RF signals in fiber and coaxial cable is much less than the speed of light. For system deployments to be effective, the cable modem protocols must support cable modems out to a wire distance of 50 miles (80 km).

At these distances the round trip propagation delay will be on the order of 800 microseconds; which is several times the length of time it takes to transmit a 64-byte packet on the upstream channel. The IEEE and DOCSIS cable modem protocols have been engineered to overcome these propagation delays in order to increase channel utilization; that is demand-based scheduling of a slotted upstream channel coupled with precise station ranging and timing.

Another challenge is in using an IP-over Ethernet approach to providing a reliable public switched packet service to an abundance of subscribers. Traditional Ethernet networking has always relied on all the Ethernet stations being within the same administrative walls with all users sharing the same common interests. Not so with metropolitan area public access networks. Data communications must now be encrypted such that the privacy of user communications is not invaded by promiscuous neighbors. In addition, users are paying for access in this cable modem world, and any abusive behavior of users must be contained so as to not affect other users. This calls for sophisticated fairness scheduling in the head-end systems and the use of comprehensive cryptological and packet filtering techniques. It is all very complicated both to create, and to manage. Each standard has its own approach for dealing with these issues.

Where IP over CATV appeared to be fundamentally similar to Ethernet when the industry first started out, in reality it is not. High-speed cable data networking, as demonstrated by the work output from various standards activities, is fundamentally a new approach to what at first appeared to be similar old problems. It's not ALOHA anymore[8], nor is it your grandfather's Ethernet[9, 10].

IEEE 802.14 Cable TV MAC and PHY Protocol Working Group

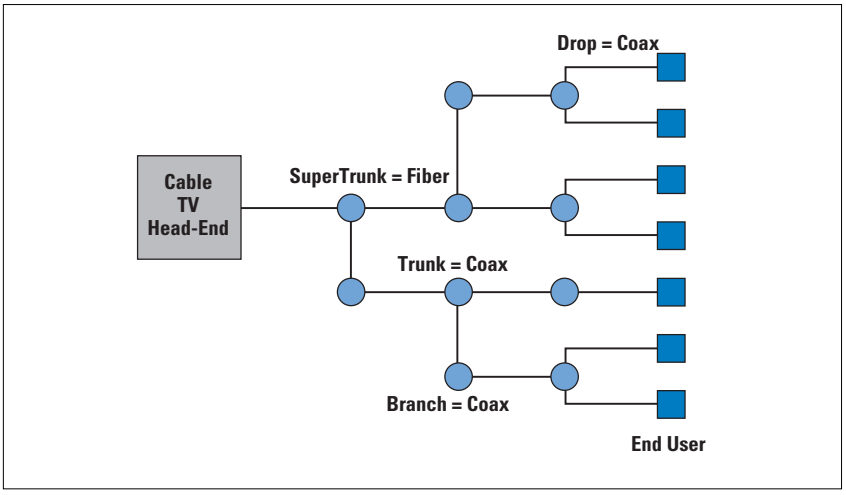
Let's briefly examine the first comprehensive standard activity created to address the current emerging world of high-speed cable data systems. In November 1994, the IEEE 802.14 CATV MAC and PHY Protocol working group met for the first time as an approved project within the 802 standards committee. Previous work had been done in 1993 through 1994 in the 802.catv study group in preparation for formal IEEE 802 project approval. The *Project Authorization Request* (PAR) charter of the group specifies that it will standardize a single MAC layer protocol and multiple PHY layer protocols for two-way HFC networks. Consistent with the IEEE LAN/MAN 802 Reference Model[11], 802.14 is producing a solution that supports the 802 protocol stack while at the same time supporting ATM in an ATM-compatible manner.

The general 802.14 requirements include:

- Communications support for all coaxial and hybrid fiber-coaxial cable TV network tree and branch topologies. (See Figure 1)

- Support of symmetrical and asymmetrical rates
- Support of *Operation, Administrations, and Maintenance* (OAM) functions
- Support of one-way delays on the order of 400 microseconds (round-trip delays to 800 microseconds)
- Support of a large number of users
- Support for moving data from an originating subnetwork to a destination subnetwork, which may be the same or a different one

Figure 1: CATV Tree and Branch Network



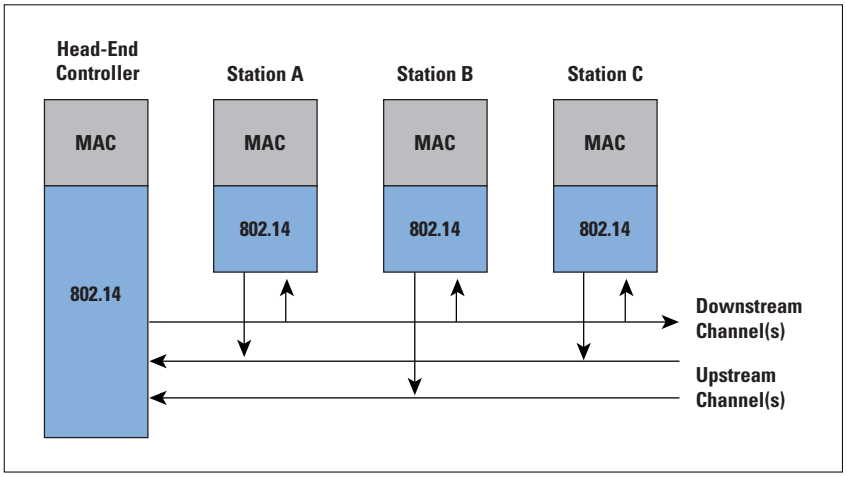
The working group completed a first-release revision of a functional requirements document back in 1995[12], which detailed the 802.14 cable topology model; defined key assumptions, constraints, and parameters; defined key performance metrics and criteria for the selection of multiple PHY protocols and a MAC protocol; and defined the support of *Quality-of-Service* (QoS) parameters. The working group's work plan called for the close of formal proposals in November 1995, with the recommended protocol defined in July 1996. Seventeen MAC protocol proposals were submitted to the working group. Needless to say, it took awhile for the working group to sort through all the issues and opinions. After much consideration, debate, and wrangling of both solutions and personalities, IEEE 802.14 stabilized on a working group draft in September 1998. This working group draft is now being submitted through the IEEE 802 standard approval process.

The 802.14 MAC and PHY specification includes:

- Definition and operational specifications for cable system Head-End Controller and cable modem Stations. (See Figure 2)
- Support of both connectionless and connection-oriented services

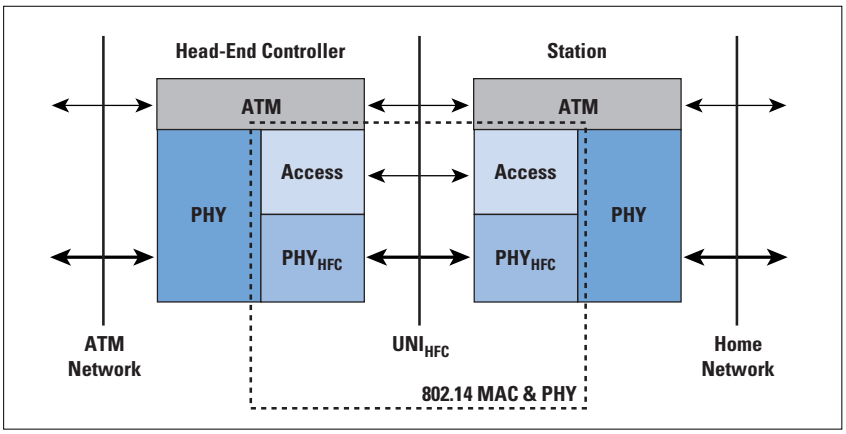
- Support of a formal QoS for connections; support for dynamically allocated bandwidth for different types of traffic, including *Constant Bit Rate* (CBR), *Variable Bit Rate* (VBR), and *Available Bit Rate* (ABR)
- Support for unicast, multicast, and broadcast services; interoperability with ATM
- Predictable low-average access delay without sacrificing network throughput
- Fair arbitration for shared access to the network within any level of service
- Downstream channel support for 64 QAM or 256 QAM modulation
- Compatibility for both international and North American downstream digital video standards
- Upstream channel support for QPSK or 16 QAM modulation

Figure 2: IEEE 802.14 General Model



The selection of ATM cells as the data link layer protocol data unit for IEEE 802.14 networks has the advantage that it provides a suitable integrated multiplexing platform capable of supporting a mix of guaranteed (predictive) traffic flows with best-effort (reactive) traffic flows. See Figure 3. Cable operators can deploy IEEE 802.14 based ATM systems as part of an evolutionary path to a fully integrated multimedia bearer service offering. A residential ATM bearer service easily supports Internet access to the home via the Classical IP over ATM standards of the Internet Engineering Task Force[13] or by providing an IP over Ethernet adaptation overlay service[14]. The development of QoS scheduling support in the Head-End Controller is left for vendors to implement[15, 16, 17].

Figure 3:
IEEE 802.14 ATM
Protocol Model



IEEE 802.14 Status

At the time of this writing, the IEEE 802.14 working group just finalized a working group draft suitable to introduction into the IEEE standards process. The entire IEEE process takes about a year from acceptance of the working group letter ballot to producing a published standard.

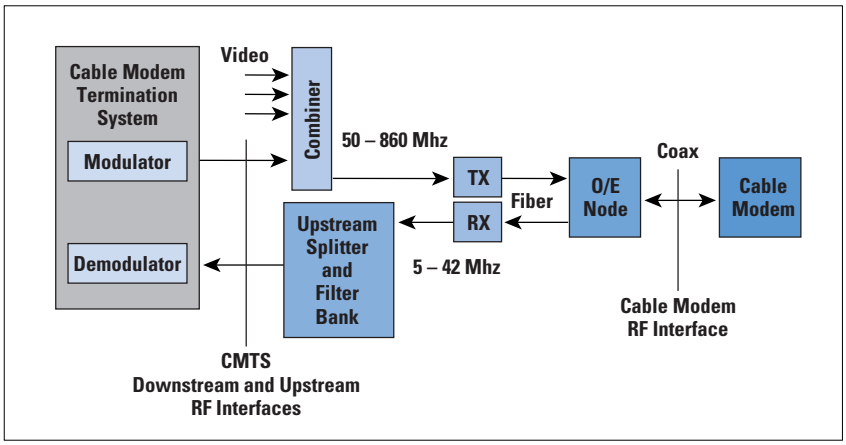
MCNS DOCSIS

The DOCSIS project is an activity of major cable companies and selected vendors to rapidly develop, on behalf of the North American cable industry, the necessary set of communications and operations support interface specifications for cable modems and associated equipment. The activity was triggered by John Malone in December 1995, in response to competition, vendor postures, and unfortunate lack of progress in the public standards process (that is, IEEE 802.14). The target for the specification was to produce a residential, "low-cost," off-the-shelf, Internet access service, with wide-scale vendor interoperability for base functions with sufficient hooks and room for vendor differentiation.

MCNS specifications are intended to be non-vendor specific, allowing cross-manufacturer compatibility for high-speed data communications services over two-way HFC cable television systems. MCNS met its specification release deadline and published versions of the *DOCSIS Radio Frequency (RF) Interface Specification* V1.0. The first draft specification was published in December 1996. The latest specification was published in July 1998[3]. The DOCSIS RFI protocol is based on the original LANCity symmetric 10 Mbps protocol, evolved to an asymmetric system, with multiple upstream and high-speed downstream (for example, 30 Mbps) channel support.

The MCNS system model is very similar to the IEEE 802.14 general model and includes many interfaces to a cable modem system, as shown in Figure 4. The goal of the DOCSIS project is to produce specifications for the CATV RF interfaces, including behavior of the *Cable Modem Termination System* (CMTS) and *Cable Modem* (CM) with respect to delivery of the residential IP over Ethernet service.

Figure 4: Data-Over-Cable RFI Reference Architecture



The DOCSIS RFI system is asymmetric, with one to several downstream channels operating asymmetrically with one to several upstream channels. Specific features of MCNS DOCSIS RFI Version 1.0 include:

- Switched Ethernet service for Internet transport via a variable length MAC packet protocol
- Best-effort service
- Downstream data channel rates from 20 Mbps (16 QAM) to 40 Mbps (256 QAM) with a typical configuration of 30 Mbps (64 QAM) in 6 MHz channels
- Compatibility for North American downstream digital video standards. (See article starting on page 27.)
- Downstream data channel rates selected from 320 Kbps (QPSK) through 10.24 Mbps (16 QAM). Channel spectral widths from 200 KHz to 3.2 MHz
- Software flexibility: ability to download new software to change/update CM behavior
- Many filters and features for controlling packet flow and classification
- Comprehensive MIB specifications for control of the cable modem and cable modem termination system
- A single large LAN segment

Due to the time-to-market push for DOCSIS RFI V1.0 interoperable modems, little to no attention was been given for QoS needs however, vendors will likely include some QoS support in their offerings. (Upstream packet fragmentation was removed from the December 1996 draft release.)

CMs and the CMTSs have basically the same protocol stack: downstream and upstream PHY, the DOCSIS RFI MAC, Ethernet and an Ethernet switching layer with substantial filtering, IP/*Address Resolution Protocol* (ARP), *User Datagram Protocol* (UDP), and *Simple Network Management Protocol/Dynamic Host Configuration Protocol/ Trivial File Transfer Protocol* (SNMP/DHCP/TFTP).

The DOCSIS RFI includes upstream and downstream optional packet encryption using the *Data Encryption Standard* (DES) to provide link privacy. RSA public key exchange is used between the CM and CMTS.

DOCSIS RFI Status

CableLabs is actively driving multiple vendor interoperability with the goal of having "silicon interoperability" as soon as possible for DOCSIS "certified" CMs and CMTSs. CableLabs runs a variety of test and certification laboratories in their facility. Numerous vendors are participating. It was the expectation to have many cable modem vendors certified by the cable industry major trade show, the Western Cable Show, in December, 1998. However, as interoperability does take time to work out, the process is taking longer than expected. There will likely be some certified vendors by December 1998, with many more in first quarter 1999. It is now expected that the first widespread deployments of DOCSIS cable modems will start in late first quarter 1999.

The DOCSIS project is currently updating the RFI Version 1.0 specification to include better support for bandwidth management and QoS support. The changes being studied include support for multiple *Service Identifiers* (SIDs), filters to perform the classification of IP packets to different SIDs for differentiated services (QoS), and the signaling support for dynamic SID creations and deletion. A scheme for packet fragmentation will be included which will give substantially better support for managing jitter for delay sensitive traffic, such as packet voice. The primary motivation for adding these extensions to DOCSIS RFI V1.0 is to provide for better support of packet voice and video over DOCSIS IP services. A major focus of the North American cable industry is to support "near toll quality" voice and video services via DOCSIS systems. The cable industry effort writing specification for packet voice and video is called *PacketCable*[18]. It is expected that the DOCSIS RFI V1.1 and initial PacketCable specifications will appear in December 1998.

DOCSIS RFI Version 1.0 was adopted by the *Society of Cable Television Engineers* (SCTE) Data Standards Subcommittee in July 1997 as the North American residential cable modem system standard.

Substantial work is in progress in the IETF *IP over Cable Data Networks* (ipcdn) working group to standardize the DOCSIS MIBs[19, 20] and to standardize IP over DOCSIS[21].

An IP over Cable Modem Example

This section presents a brief overview of a hypothetical IP over HFC system. It is meant to be an informative example to illustrate the application of the IP technology and some of the issues that surround provision of the service over a residential cable TV network. Moving IP datagrams in and out of the home over the cable plant is the important issue. The specific technology and protocols used by the cable modem vendor are important only in their ability to provide required IP service support.

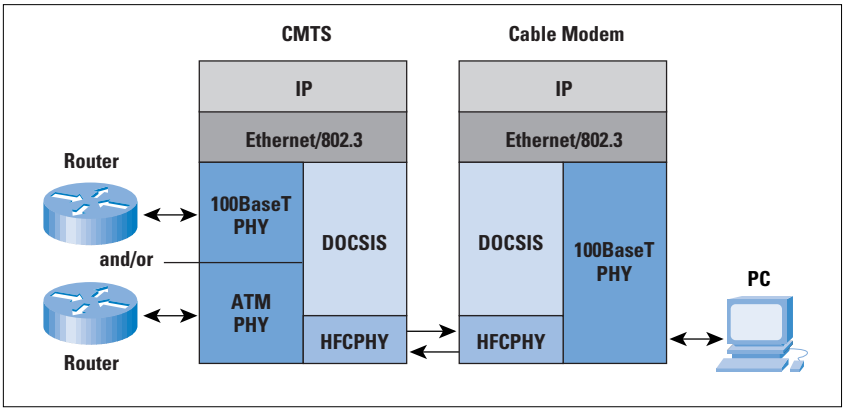
For this example, consider a system that has the following design goals and requirements:

- One-to-many service will be supported in the downstream direction; that is, many cable modems are reachable via the downstream channel
- Many-to-one service will be supported in the upstream direction; that is, the upstream channel bandwidth will be shared. There may be up to several upstream channels
- The protocol used between the Head-End Controller and the head-ends is not significant as long as it meets the needs of the IP service
- The head-end owns the upstream bandwidth and allocates resources to cable modems
- IP over Ethernet 10BaseT is the required interface in the home
- IP over Ethernet or IP over ATM is the required interface at the head-end

This example will rely on the DOCSIS RFI information presented previously in this article. The CMTS can transmit packets to any cable modem on the channel in any order or rate appropriate to the scheduling information it has and controls. The CMTS also participates in the IP multicast group membership (*Internet Group Management Protocol* [IGMP]) and IP *Resource Reservation Protocol* (RVSP) and makes changes in the cable modem resource assignments and allocations as needed. The home cable modem is permitted to use only the upstream channel under direction of the CMTS. Guaranteed and best-effort bandwidth allocations are dynamically assignable by the CMTS. It is assumed that the cable modem protocol has a bandwidth request facility that allows a CM to ask the CMTS for bandwidth. The function of the bandwidth management process is to sort these requests for service and give fair access to the requesting cable modems.

The method for implementation of an Ethernet and 802.3 bridging function over DOCSIS essentially permits the RF channels to act as a serial connection between a half-bridge function in each cable modem with a master in the CMTS. Figure 5 illustrates the protocol stack for this solution. The system presents an Ethernet-like segment to the cable operator. It is well-known how to put together such segments to construct larger internetworks.

Figure 5: Bridged Ethernet via DOCSIS Example



Cable modems provide demarcation between the Internet Service Provider's network and each home network. To help the Internet Service Provider offer fair access service to its residential customers, the cable modem will require sufficient dynamic functionality for multilayer protocol filtering and various forms of rate management (see Figure 6). The goal of this filter is to create a defense perimeter at the first point of entry to the cable network; this perimeter will protect the upstream channel from being saturated or abused by misbehaving home networks. Some examples of this filtering functionality include, but are not limited to:

- Filtering on Ethertype for permitting only certain protocols to pass upstream; for example, IP and ARP only
- Filtering on IP source or destination address to permit/deny access from the home network
- IP and Ethernet broadcast rate limiting; that is, keep any home network broadcast storms confined to the home network
- IP Multicast group address filtering; that is, explicitly permit participation of the home network in an IP multicast group

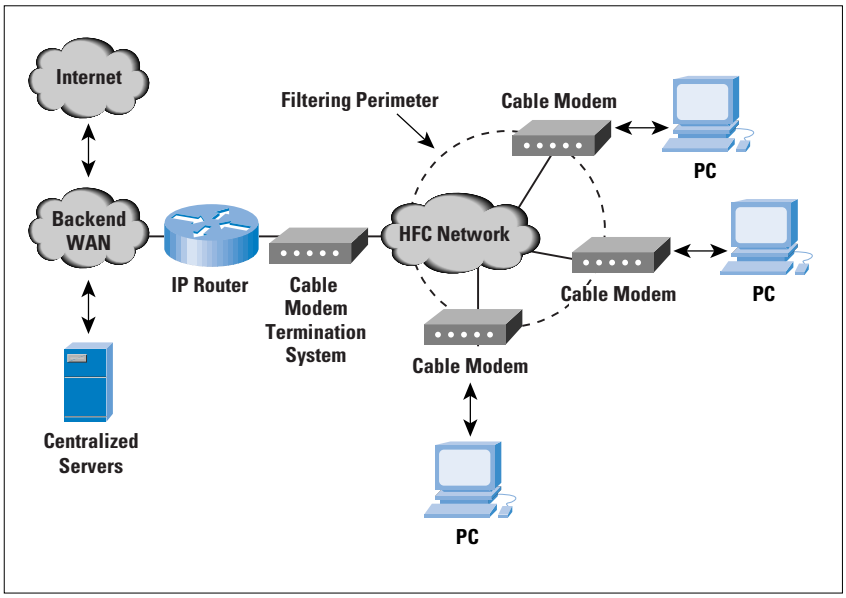


Figure 6: Internet Services via Cable Modem Deployment Model

It should be noted that these filtering functions are under consideration by numerous cable modem manufacturers, and they are being discussed in the IETF ipcdn working group.

A brief overview of IP over cable TV networks has been presented. From an engineering and deployment viewpoint, making the Internet move over cable modems is deceptively straightforward. Many issues are beyond the scope of this article: address allocation methods, back-end network design, configuration services, server placement, home customer support services, installation, firewalls, and troubleshooting.

Summary

This article has presented an overview of the work in progress of the IEEE 802.14 Cable TV MAC and PHY Protocol Standards working group and the MCNS DOCSIS effort. Initial review of these works is positive; indications are that data over HFC systems are viable. The IEEE 802.14 effort began as a study group in late 1993 and has yet to produce a standard. The MCNS DOCSIS process started in early 1996, moved rapidly, and has produced an accepted international standard specification for North American cable operators for residential cable modem service. The IEEE 802.14 standard appears to be destined for some international use and in systems where ATM over CATV is preferred by cable operators.

The cable network environment will provide a very usable and scaleable bandwidth platform for delivering Internet services to and from the home[22]. A hypothetical example was provided that illustrates a general equipment deployment model. Actual deployment of Internet to the home will occur in many areas of North America in 1998 with increasing and substantial deployment in 1999.

For More Information

Information on the IEEE's 802.14 working group can be found on the World Wide Web at: **http://www.walkingdog.com/**

Information the Internet Engineering Task Force's IP over Cable Data Networks working group can be found at: **http://www.ietf.org/**

Information on the North American MCNS DOCSIS effort can be found at: **http://www.cablemodem.com/**

Information on the North American PacketCable effort can be found at: **http://www.packetcable.com/**

Information on the SCTE Data Standards Subcommittee can be found at: **http://www.cablenet.org/scte/scte_dcs.html**

References

[1] Baran, Paul, "On Distributed Communication Networks." IEEE *Transactions on Communication Systems*, Vol. CS-12, pp. 1–9, March 1964.

[2] ATM Forum, "ATM User-Network Interface Signaling 4.0," Specification number af-sig-0061.000, **www.atmforum.com**, July, 1996.

[3] MCNS, "Data-Over-Cable Service Interface Specification—Radio Frequency Interface." SP-RFI-I02-981008, **www.cablemodem.com**, July, 1998.

[4] MCNS, **www.cablemodem.com**, main page, April 1998.

[5] Kim, Albert. "Two-Way Plant Characterization." Technical Session 23, National Cable Television Association Show and Conference, Dallas, Texas, May 9, 1995.

[6] Chelehemal, M., Prodan, R., et al., "Field Evaluation of Reverse-Band Channel Impairments." Society of Cable Telecommunications Engineers, Emerging Technologies Conference, San Francisco, California, January 9–12, 1996.

[7] Laubach, Mark, "Avoiding Gridlock on the Data Infobahn: Port Mismatches Pose Challenges." *CED Magazine,* March 1998

[8] Abramson, Norman, "Development of the ALOHANET." IEEE *Transactions on Information Theory,* Vol. IT-31, pp. 119–123, March 1985.

[9] XEROX, "The Ethernet, A Local Area Network: Data Link Layer and Physical Layer Specification." X3T51/80-50, Xerox Corporation, Stamford, Connecticut, October 1980.

[10] IEEE, "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications." Standard 802.3-1985 (ISO DIS 8802/3), IEEE, New York, ISBN 0-471-82749-5, 1985.

[11] IEEE, "IEEE Standards for Local Area Networks: Logical Link Control, ANSI/IEEE Std 802.2-1985." Fifth printing, February 1988.

[12] IEEE 802.14 Working Group, "Cable-TV Functional Requirements and Evaluation Criteria." Work in progress, IEEE802.14/94-002R2, IEEE 802 Committee, February 1995.

[13] Laubach, Mark. "Classical IP and ARP over ATM." RFC 1577, January 1994.

[14] Laubach, Mark, "Logical IP Subnetworks over IEEE 802.14 Services." Work in progress, **draft-ietf-ipcdn-ipover-802d14-01.txt**, November 1997.

[15] Laubach, Mark, "Serving Up Quality of Service." *CED Magazine,* April 1997.

[16] Laubach, Mark, "Deploying ATM Residential Broadband Networks." NCTA Cable 96 Conference, Los Angeles, California, April 30, 1996.

[17] Nichols, Kathleen, and Laubach, Mark, "On Quality of Service in an ATM-based HFC Architecture." IEEE ATM Workshop 96, San Francisco, California, August 27, 1996.

[18] PacketCable, "What is PacketCable?" **http://www.packetcable.com**, April 1998.

[19] Roeck, Guenter, "Cable Device Management Information Base for MCNS compliant Cable Modems and Cable Modem Termination Systems." Work in progress, **draft-ietf-ipcdn-cable-device-mib-05.txt**, October 1998.

[20] Roeck, Guenter, "Radio Frequency (RF) Interface Management Information Base for MCNS compliant RF interfaces." Work in progress, **draft-ietf-ipcdn-rf-interface-mib-05.txt**, October 1998.

[21] White, Gerry, "Logical IP Subnetworks over MCNS Data Link Services." Work in progress, **draft-ietf-ipcdn-ip-over-mcns-00.txt**, August 1997.

[22] Lucien Rhodes, "The Race for More Bandwidth." (Interview with Milo Medin of @Home), *Wired Magazine,* Vol. 4.01, January 1996

Internet Drafts are *works in progress* and can be retrieved from:
ftp://ds.internic.net/internet-drafts

MARK LAUBACH holds a B.E.E. and M.Sc. from the University of Delaware. He is Vice President and Chief Technical Officer at Com21, Inc. in Milpitas, California, and is responsible for the end-to-end systems architecture and ATM over HFC protocol specification of the Com21 product family. Prior to Com21, he was with the Hewlett-Packard Company for 14.5 years. Laubach is a member of the IETF, and is past chair of the IP over ATM working group. He is the author of RFC 1577, "Classical IP and ARP over ATM." He regularly attends IETF, IEEE, and SCTE working group meetings. He is a Senior member of the IEEE and a member of the SCTE. E-mail: **laubach@com21.com**

Digital Television: A New Venue for the Internet

by George Abe, Cisco Systems

The digitization of television is of interest to the Internet community in that it opens the possibility of a new mode of delivering IP packets to the home. IP services can be delivered over television broadcast distribution networks, whether over the air, cable, or satellite. This article introduces the basic concepts of *digital television* (DTV) and provides a point of departure for further reading.

Why Is Digital TV Happening?

The original motivation for the research into advanced TV (we avoid the term DTV for a moment) was to prop up sagging TV sales. It was mostly vendor push.

By the late 1970s, Japan and Korea had achieved domination in the production of TV sets worldwide. They were so successful that the market had become saturated, particularly in the developed world. Everyone had one or, more likely, three or four TVs at home. Further, a TV lasts over 10 years, so the replacement market is low. TV production had ceased to be a growth market. Margins were and are poor and few innovations were on the horizon.

So in the early 1980s Japan had begun research into new high-definition televisions that would stimulate new demand and enable them to keep their market leadership. Their system is called *Multiple Subnyquist* (MUSE). MUSE was an analog system, but it had better-quality pictures.

Not to be outdone, the U.S. decided it needed to try to recapture the TV market, so began its own development, under the aegis of the Federal Government. A partnership called the *Grand Alliance* was formed, and it began working in 1984. Pioneering work was done by the partnership members, particularly Zenith, MIT, and General Instruments. They created a digital specification after more than a decade of research and development. Along the way, the computer industry made contributions (or some would say interferences) of its own until the FCC announced a final specification in December 1996. The basic elements are found at **`www.atsc.org`** and referenced later in this article.

Benefits of DTV

The movement toward widespread DTV gained momentum among government officials, broadcasters, and hardware vendors when some of the benefits became clear.

First, because of improvements in technology, it is possible to transmit pictures and sound of significantly higher quality in the same 6 MHz spectrum that analog TV occupies. The 6 MHz spectrum is wasteful of bandwidth, and the government would like to recover the excess so it can be auctioned or used to support other public services (police, fire,

deep space probes, and so on), which could operate at the relatively low frequencies of VHF TV.

Second, digitally encoded TV could provide new services, such as Web access via TV or interactive TV. These have long been dreams of the consumer electronics (CE) industry, but hope springs eternal.

Third, digital TV offers greater security to the programmer and the network. There is a cottage industry in hacking analog set-top boxes. Digital techniques, such as the *Data Encryption Standard* (DES), double DES, and triple DES give operators hope that they can secure their pay-per-view content.

Finally and most interestingly, since digital TV occupies less bandwidth per program, broadcasters, satellite operators, and cable operators have the opportunity to offer more channels. Instead of a mere 10–13 channels available over the air in a single metropolitan area, it is possible to have perhaps 60 or more over the air channels. Cable operators, with their greater bandwidth underground, could have many more channels. Although technically cable could offer 500 channels, it is hard to imagine where the scripts would come from.

What Is DTV?

By our definition, digital television is the capture, production, distribution, and broadcast of programming in a digitally encoded format. Whereas today's analog TV transmits in amplitude modulation, DTV would use *Quadrature Phase Shift Keying* (QPSK), *Quadrature Amplitude Modulation* (QAM), or *Vestigal Side Band* (VSB) modulation techniques. We won't detail these techniques here except to mention that they are mutually incompatible.

When DTV standards were discussed in the 1980s, the industry could not agree on a single display. The deliberations became more protracted with the entry of the computer industry into the discussions, long after the broadcasters and consumer electronics people began their work. Would there be interlaced or progressive scanning? Would there be the existing aspect ratio or would there be a wide-screen display? Square pixels or not? How many lines of resolution would be displayed?

With the broadcasters and consumer electronics vendors arguing for interlacing, oval pixels, and wide screens and the computer people arguing for progressive scanning, square pixels, and a more square display, the disagreements could not be bridged.

Therefore, the FCC had no choice but to declare that the "market should decide" which display format would prevail. Accordingly, the FCC announced in December 1996 that 18 different display formats would be permissible for over-the-air digital TV. A broadcaster could elect to transmit in any of the approved formats. The approved formats are shown in Tables 1 and 2.

Table 1: Progressive Video Scanning Formats for Digital TV

Vertical Lines	Horizontal Pixels	Aspect Ratio	Frame Rate per Second
1080	1920	16:9	24, 30
720	1280	16:9	24, 30, 60
480	704	16:9	24, 30, 60
480	704	4:3	24, 30, 60
480	640	4:3	24, 30, 60

Table 2: Interlaced Video Scanning Formats for Digital TV

Vertical Lines	Horizontal Pixels	Aspect Ratio	Frame Rate per Second
1080	1920	16:9	30
480	704	16:9	30
480	704	4:3	30
480	640	4:3	30

The vernacular to describe the formats typically indicates the number of vertical lines and the scanning format. For example, "1080i" refers to 1080 lines, interlaced scanning; "720p" refers to 720 lines in progressive format.

In practice, only a few of the 18 approved formats are under consideration by the nation's broadcasters. NBC and CBS have declared they will support 1080i. ABC is opting for 720p, and Fox has opted for 480p.

Apart from the controversy over display, most of the other elements were quickly resolved. Modulation scheme, transport multiplexing, compression, timing, and an overall systems and testing procedure were agreed to. The apparatus for DTV was in place, almost. The time was January 1997.

High Definition or Standard Definition

Some view DTV as synonymous with high-definition television. It is not. DTV encompasses both *High-Definition TV* (HDTV) and *Standard-Definition TV* (SDTV). Hence HDTV is a proper subset of DTV. The difference between HD and SDTV is not standardized, but our definition of HD includes the display formats that have 720 or 1080 lines. Formats with fewer lines are standard definition.

The key point of difference between HD and SD is that with HD and current compression techniques (MPEG-2), only one program is accommodated in one 6-MHz channel. With SD, it is possible for the broadcaster to transmit two or more programs simultaneously, in a single 6-MHz chunk of bandwidth.

This has tremendous implications. If broadcasters can transmit multiple channels at once, it would be possible (technically) for Disney to broadcast ABC, the Disney Channel, ESPN, and A&E over the air in the same bandwidth they use to show ABC today. (Of course they won't do this for commercial and contractual reasons, but the technology makes it doable).

For Internet Service Providers, a broadcast could transmit SD programming simultaneously with datacasting, and go into the push-mode data service business. For example, Disney/ABC could download software updates for Disney Interactive, or perhaps contract with Microsoft to deliver Windows updates. Whereas most Internet folk view MPEG being transported inside IP packets on the Internet, broadcasters intend to insert IP packets into MPEG-2 transport streams. The consumer's digital set-top box would tune to the data "channel," extract the data from its MPEG capsule, and divert the data packet to an Ethernet or ATM port on the set-top.

There are nearly 1,600 broadcasters in the U.S. Each could, in theory, transmit 19.3 megabits per second. Of course, most of these bits will be used for television, but certainly 1 or 2 megabits can be accommodated by each broadcaster for data service.

Given the dearth of programming to fill multiple SD channels, broadcasters are strongly motivated to consider data services and compete for a slice of the Internet service market.

Digital TV—End to End

Whereas one easily thinks of DTV as a distribution and display technology, in fact there are major changes required to capture, edit, and distribute digital content. Thus there is the need for new cameras, post-production editors, sound mixers, and the like.

Digital TV can be transmitted over the air, through cable networks, or via *Direct Broadcast Satellite* (DBS). Today, only DBS has achieved large-scale distribution of digital TV, with over 7 million subscribers in the U.S. and 15 million worldwide.

Content is created either through a digital camera or by converting existing analog content, such as 35mm film, into digital format. Within the production environment, editing changes are made, typically using *Nonlinear Editors* (NLEs) that connect to a local-area network.

Original production is normally done in the high definition. The highest form of resolution is 1.492 Gbps. (See Table 3.) Equipment to do this is not widely available, but it will be eventually. Panasonic is shipping a digital camera capable of 1.5-Gbps output, but rumor has it they cost almost $500,000, if you can even get one. Nonetheless, 41 stations began HD programming in November, highlighted by an NFL game on CBS between the Buffalo Bills and the New York Jets on November 8.

Some compression is applied within the postproduction and editing environment. The TV industry, through the *Society of Motion Picture and TV Engineers* (**www.smpte.org**), developed a series of digital transmission standards. Chief among these is SMPTE 305M, which defines a protocol called *Serial Data Transport Interface* (SDTI), which calls for a 270- or 360-Mbps service to link various pieces of production equipment such as NLEs in a postproduction facility. SMPTE 305M is a networking scheme complete with an addressing specification.

(Interesting point about 305M: It is the first and only protocol known to this author that specifies use of IPv6 addressing.)

Another important protocol is SMPTE 259M, which is a link-layer protocol underneath 305M.

A competing protocol to SDTI is the *Digital Video Broadcasters Asynchronous Serial Interface* (DVB-ASI). Information on DVB-ASI is found at **www.dvb.org**.

From the editing environment, content is distributed via satellite or land lines to local affiliates (for local over-the-air broadcast), cable head-ends (for cable TV distribution) and satellite hubs (for direct-to-home satellite service). The distribution from national feeds to local facilities is normally at T3/E3 speeds because of the availability of T3/E3 services by telephone companies and satellite transponders for affiliate and direct-to-home distribution.

Cable providers, local broadcasters, and satellite services add their own content and make certain changes to the national feeds. Among these changes are assignment of the programming to specific frequencies or channels, insertion of local advertising, local programming, and emergency broadcasts.

After adding their own content, the local services distribute the final programming to consumers. Over-the-air broadcasters will transmit 19.3 Mbps per 6 MHz, cable will transmit 27 Mbps per 6 MHz, and satellite uses variable channelization, kept closely under wraps.

So there is the progression downward from 1492 Mbps of original encoding, to 270 Mbps for editing, to 34/45 Mbps for affiliate distribution, to 27 Mbps or less for distribution to the end user.

Table 3: Bit Rate Requirements for Various Display Formats

Format	Pixels per Line	Lines per Frame	Pixels per Frame	Frames per Second	Millions of Pixels per Second	Bits per Pixel	Mbps
SVGA	800	600	480,000	72	34.6	8	276.5
NTSC	640	480	307,200	30	9.2	24	221.2
PAL	580	575	333,500	50	16.7	24	400.2
SECAM	580	575	333,500	50	16.7	24	400.2
HDTV	1920	1080	2,073,600	30	62.2	24	1492.8
Film	2000	1700	3,400,000	24	81.6	32	2611.2

Note: Film display formats vary, depending on content and directorial prerogative.

Over the Air and Cable

All the huffing and puffing by the FCC, the consumer electronics industry, the computer industry, and the broadcasters pertains to over-the-air transmission. However, about two-thirds of the American viewing public views TV through cable. So if most Americans are to receive DTV, they must receive it through cable.

This raises important technical and regulatory questions. The technical question is: How are the digital signals produced by the broadcasters and their affiliates to be sent through wires, and what is the allocation of functions between the digital set-top and the digital receiver? This question seems simple but it is not, as we shall see.

The regulatory question pertains to whether the cable operators are to be compelled to carry DTV from broadcasters. This problem is referred to as the digital *Must Carry Problem,* now under consideration by the FCC. It certainly will be litigated, whatever the outcome of the FCC's decision.

Technical Question

Among the key provisions agreed to by the Grand Alliance is the use of a modulation technique called 8-VSB for over-the-air digital transmission. The particulars of 8-VSB are not significant here, but we will mention that this particular decision was arrived at in the mid-1980s, before the cable industry had much impact on the viewing public or on the broadcasting industry.

When the cable industry began to think about digital, in the mid-1990s, they settled on a modulation scheme called 64 QAM. 64 QAM is able to produce 27 Mbps in 6 MHz, whereas 8-VSB produces about 19.3 Mbps. The difference occurs because over-the-air broadcasting requires a more robust encoding scheme to combat the more hostile nature of over-the-air transmission, as opposed to the safer environment of coaxial cables. Thus the cable modulation technique can be more aggressive than over-the-air techniques.

(We should add that satellites use an even more robust modulation technique called QPSK, which gets fewer bits per Hertz than VSB or QAM. But robustness is needed because satellite signals must travel far greater distances than cable or local broadcast.)

Thus for cable to carry a digital over-the-air broadcast, some conversion of 8-VSB encoding to 64 QAM encoding is necessary. This necessity does not present a major technical problem, but agreement is needed on where the conversion is done and at what cost. For example, Broadcom and Sony are collaborating on the development of a chip, to be embedded in a TV, that can decode VSB and QAM. It sounds simple, but the cable industry is not interested. They want to carry QAM and QAM only on their networks.

One option is to convert the format of the digital bitstream coming out of the cable box to the IEEE 1394 *FireWire* format. Since DTVs are likely to have FireWire input, this conversion can provide a ubiquitous connection. However, this scenario raises the problem of copy protection, a sore point in Hollywood. Since digital copies are pristine, the content providers (studios and record companies) are firm in their resolve that unless there is strong copy protection, none of their content will be available over FireWire.

Another option is to build a set-top box that takes baseband signals and modulates them to look like 8-VSB broadcast signals on channel 3, similar to how VCRs work in the analog world now. This scenario is clearly rather ugly, but understood by consumers.

Finally, it could be up to the cable operators to transmodulate the 8-VSB into QAM at the cable head-end. Better yet, they can accept broadcasters' feeds in baseband, and then QAM-modulate the baseband signals for their consumers. The cable set-top box would be sending bit maps to a dumb digital monitor, like a computer monitor, which doesn't know or care that it is receiving QAM or VSB programming.

Apart from modulation, there is the issue of display format. NBC and CBS have declared they will transmit in 1080i. ABC has chosen 720p and Fox has chosen 480p, with some vague pledge for higher definition later. After all, it does not seem necessary to show *The Simpsons* in HD.

On the other hand, John Malone, Chairman of TCI, went public in May 1998 with his declaration that TCI would not voluntarily carry 1080i because it (1080i) was wasteful of bandwidth. Implied in his comment is the fact that cable operators do need to be restricted to 6-MHz channelization for digital. In fact, the entire DTV spectrum on cable could be considered a gigantic pool of bandwidth that the cable operator could allocate to individual channels, much as direct satellite does. This setup gives the cable operators incentive to downconvert the broadcasters' DTV signals. For example, when NBC sends 1080i, the cable operator may elect to transmit 720p, or less, to its customers.

Should the cable operators be required to carry the HDTV pictures from the broadcasters in the broadcasters' chosen format? Would they be allowed to downconvert the HD into standard definition? What happens when a broadcaster, say NBC, elects to transmit in SDTV and thereby has the capability of multiplexing several channels onto a single chunk of 6 MHz? What is the duty of the cable operator to carry Internet datacasting offered by the broadcasters over the cable network, in competition with services such as @Home and Roadrunner?

The complexities of multiplexing go further. Let's say ABC elects to broadcast SD. If one of the subprograms in the multiplex is a pay-per-view channel, should the authentication procedures of the cable operator be superceded? Should the electronic program guide of the cable operator be superceded?

Questions like these have technical and regulatory aspects and are being worked in industry, the FCC, and state regulatory agencies. It is possible that Congress will get involved as well. When John Malone made his statement, both sides of the aisle in Congress were not amused. They want DTV to happen so that spectrum can be freed. If the cable operators stand in the way, the conversion to digital is stopped dead in its tracks.

The Open Cable Initiative

The cable industry does not want to be a bottleneck to broadcasters. On the other hand, it needs to make quick progress into DTV to compete against satellite. Therefore, the industry has embarked on a process called *Open Cable,* which seeks to define a digital set-top box that can be available at retail. Available at retail means a nonproprietary, open design. Open Cable strives to make the DTV set-top box independent of processor platform (that is, not an Intel Pentium necessarily) and operating system independent (that is, not a Microsoft Windows CE necessarily).

The Open Cable set-top box will allow for data services through a specification written by the *Digital Audio Visual Council* (DAVIC—**www.davic.org**) and therefore, is not compatible with the current *Data-over-Cable Service Interface* (DOCSIS) specification supported by the U.S. cable industry. (See article starting on page 13.) However, it is possible for DOCSIS capabilities to be added on to an Open Cable set-top box. We mention Open Cable because it will be the key customer premises device for cable and digital TV and much hinges on its interoperability with broadcasters transmissions.

Digital TV via Satellite

In addition to over-the-air and cable, DTV can be received by satellite. As of this writing, it is the only way to receive DTV. The digital satellite industry has nearly 7 million subscribers who received DTV today. Its role in all the discussions of HD vs. SD and the provision of data services is relatively low key because it is believed that satellite will continue to be a niche provider because of its technical and legal problems in distributing locally originated TV stations.

But satellites bear watching because if they are able to deliver local channels and obtain 15–20 million homes in the U.S., then the financial consequences on cable and over the air could be crucial.

The New Digital Studio

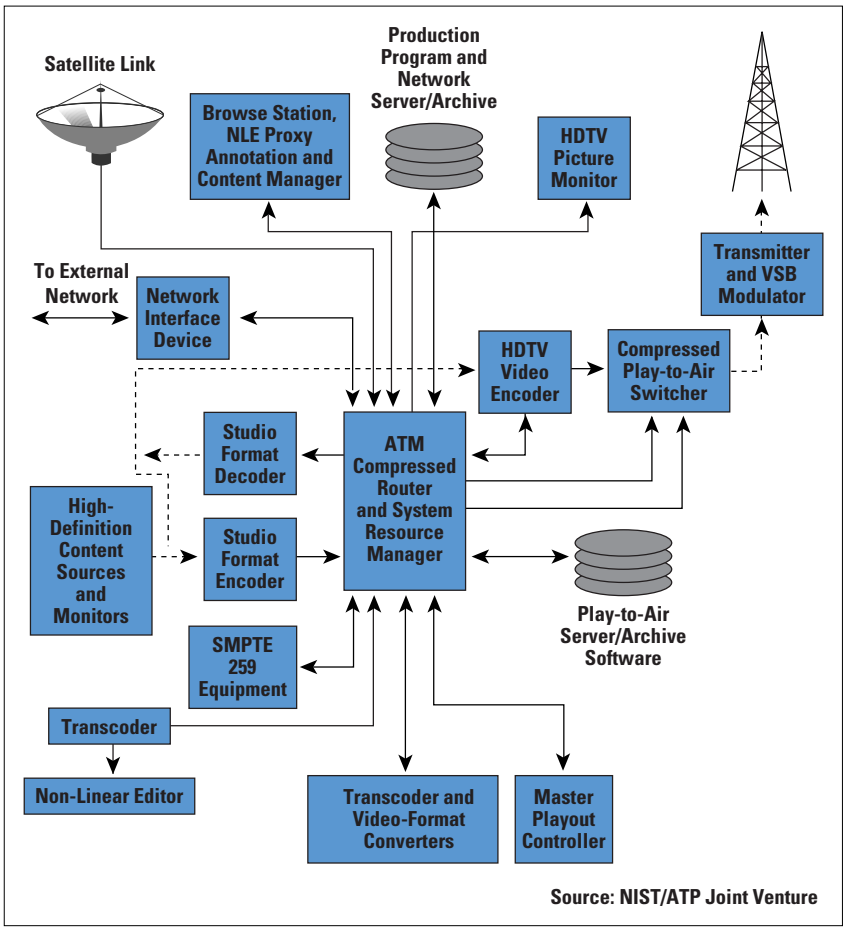
The figure shows a schematic of the elements of a DTV broadcast studio described recently by the U.S. *National Institute of Standard and Technology* (NIST). At the heart of the studio is an ATM switch with new interfaces that connect to DVB or ATSC infrastructures via DVB-ASI or SDTI interfaces.

Connection for wide-area distribution will likely be over ATM. Converters exist for DVB-ASI to ATM. For example, Cellware (**www.cellware.de**) in Germany markets such a converter, but there is no SDTI-to-ATM interface known to this author at this time.

The digital studio provides a new a marketing opportunity for the LAN industry. Broadcast digital production demands higher speeds than most other LAN applications.

Thus vendors of data communications equipment have two opportunities: to provide equipment to broadcasters who want to enter the Internet service business and to production houses that use ATM or other LANs to support editing and production applications.

Figure 1: Prototype of HDTV Broadcast Studio



Web Sites

www.atsc.org: *Advanced TV Standards Committee*. S13 and S16 are subgroups working on datacasting; S13 focuses primarily on the downstream path, whereas S16 focuses primarily on the reverse communication from the receiver. Since over-the-air is one way, this work is limited to the communications between the S13 forward channels and a telephone or Internet return path.

www.dvb.org: The *Digital Video Broadcasting Project* (DVB) has taken the lead in defining DTV specifications as well as defining datacasting interfaces over DTV infrastructures.

www.smpte.org: *Society of Motion Picture and Television Engineers.*

www.sbe.org: *Society of Broadcast Engineers.*

www.scte.org: *Society of Cable TV Engineers.*

www.mpeg.org: *Motion Picture Experts Group*. The word on MPEG compression, controls, and transmission.

References

[1] ISO/IEC IS 13818-1, International Standard, MPEG-2 Systems.

[2] ISO/IEC IS 13818-2, International Standard, MPEG-2 Video.

[3] ISO/IEC 13818-6, International Standard, Digital Storage Media Command and Control (DSM-CC).

[4] ATSC Standard A/52 (1995), Digital Audio Compression (AC-3).

[5] ATSC Standard A/53 (1995), ATSC Digital Television Standard.

[6] ATSC Standard A/55 (1996), Program Guide for Digital Television.

[7] ATSC Standard A/56 (1996), System Information for Digital Television.

[8] ATSC Standard A/57 (1996), Program/Episode/Version Identification.

[9] ATSC Standard A/63 (1997), Standard for coding 25/50-Hz Video.

[10] ATSC Standard A/64 (1997), Transmission Measurement and Compliance For Digital Television.

[11] ATSC Standard A/65 (1998), Program and System Information Protocol for Terrestrial Broadcast and Cable.

[12] ATSC T3/S13 Doc. 010 DVS-yyy Rev z Draft, ATSC Data Broadcast Specification for Terrestrial Broadcast and Cable.

[13] ETR XXX: Digital Video Broadcasting (DVB); Guidelines for the Use of the DVB Specification: Network Independent Protocols for Interactive Services (ETS 300 802).

[14] SCTE DVS-nn: SCTE Digital Video Subcommittee (DVS) standard for Cable Headend and Distribution Systems (spec not released—under development)

GEORGE ABE holds an A.B. in Mathematics and an M.S. in Operations Research from UCLA. He currently is a Consulting Engineer at Cisco Systems, where he has dabbled in various areas of residential broadband networking since 1994. He is the author of *Residential Broadband*, Cisco Press (imprint of Macmillan Press). He expects to be a early adopter of digital TV and, when not watching TV, he can be reached at **georgea@acm.org**

I Remember IANA

by Vint Cerf, MCI WorldCom
October 17, 1998

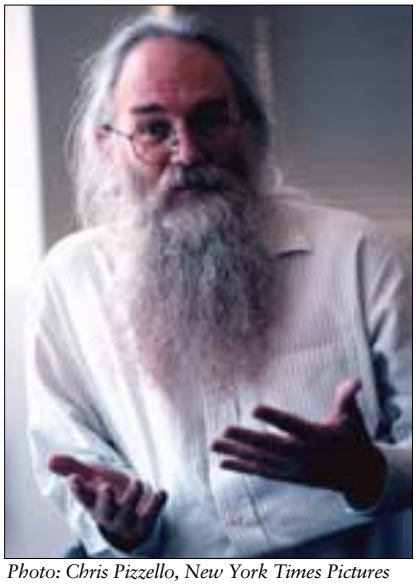


Photo: Chris Pizzello, New York Times Pictures

A long time ago, in a network, far far away, a great adventure took place! Out of the chaos of new ideas for communication, the experiments, the tentative designs, and crucible of testing, there emerged a cornucopia of networks. Beginning with the ARPANET, an endless stream of networks evolved, and ultimately were interlinked to become the Internet. Someone had to keep track of all the protocols, the identifiers, networks and addresses and ultimately the names of all the things in the networked universe. And someone had to keep track of all the information that erupted with volcanic force from the intensity of the debates and discussions and endless invention that has continued unabated for 30 years. That someone was Jonathan B. Postel, our *Internet Assigned Numbers Authority* (IANA), friend, engineer, confidant, leader, icon, and now, first of the giants to depart from our midst.

Jon, our beloved IANA, is gone. Even as I write these words I cannot quite grasp this stark fact. We had almost lost him once before in 1991. Surely we knew he was at risk as are we all. But he had been our rock, the foundation on which our every Web search and e-mail was built, always there to mediate the random dispute, to remind us when our documentation did not do justice to its subject, to make difficult decisions with apparent ease, and to consult when careful consideration was needed. We will survive our loss and we will remember. He has left a monumental legacy for all Internauts to contemplate. Steadfast service for decades, moving when others seemed paralyzed, always finding the right course in a complex minefield of technical and sometimes political obstacles.

Jon and I went to the same high school, Van Nuys High, in the San Fernando Valley north of Los Angeles. But we were in different classes and I really didn't know him then. Our real meeting came at UCLA when we became a part of a group of graduate students working for Professor Leonard Kleinrock on the ARPANET project. Steve Crocker was another of the Van Nuys crowd who was part of the team and led the development of the first host-to-host protocols for the ARPANET. When Steve invented the idea of the *Request for Comments* (RFC) series, Jon became the instant editor. When we needed to keep track of all the hosts and protocol identifiers, Jon volunteered to be the Numbers Czar and later the IANA once the Internet was in place. Jon was a founding member of the *Internet Architecture Board* (IAB) and served continuously from its founding to the present. He was the *first* individual member of the Internet Society—I know, because he and Steve Wolff raced to see who could fill out the application forms and make payment first and Jon won. He served as a trustee of the Internet Society.

He was the custodian of the .us domain, a founder of the Los Nettos Internet service, and, by the way, managed the networking research division of USC Information Sciences Institute.

Jon loved the outdoors. I know he used to enjoy backpacking in the high Sierras around Yosemite. Bearded and sandaled, Jon was our resident hippie-patriarch at UCLA. He was a private person but fully capable of engaging photon torpedoes and going to battle stations in a good engineering argument. And he could be stubborn beyond all expectation. He could have outwaited the Sphinx in a staring contest, I think.

Jon inspired loyalty and steadfast devotion among his friends and his colleagues. For me, he personified the words "selfless service." For nearly 30 years, Jon has served us all, taken little in return, indeed sometimes receiving abuse when he should have received our deepest appreciation. It was particularly gratifying at the last Internet Society meeting in Geneva to see Jon receive the Silver Medal of the International Telecommunications Union. It is an award generally reserved for Heads of State, but I can think of no one more deserving of global recognition for his contributions.

While it seems almost impossible to avoid feeling an enormous sense of loss, as if a yawning gap in our networked universe had opened up and swallowed our friend, I must tell you that I am comforted as I contemplate what Jon has wrought. He leaves a legacy of edited documents that tell our collective Internet story, including not only the technical but also the poetic and whimsical as well. He completed the incorporation of a successor to his service as IANA and leaves a lasting legacy of service to the community in that role. His memory is rich and vibrant and will not fade from our collective consciousness. "What would Jon have done?" we will think, as we wrestle in the days ahead with the problems Jon kept so well tamed for so many years.

There will almost surely be many memorials to Jon's monumental service to the Internet Community. As current chairman of the Internet Society, I pledge to establish an award in Jon's name to recognize long-standing service to the community, the *Jonathan B. Postel Service Award*, which will be awarded to Jon posthumously as its first recipient.

If Jon were here, I am sure he would urge us not to mourn his passing but to celebrate his life and his contributions. He would remind us that there is still much work to be done and that we now have the responsibility and the opportunity to do our part. I doubt that anyone could possibly duplicate his record, but it stands as a measure of one man's astonishing contribution to a community he knew and loved.

VINTON G. CERF is senior vice president of Internet Architecture and Technology for MCI WorldCom. Widely known as a "Father of the Internet," he is the co-designer of the TCP/IP protocol. Cerf served as founding president of the Internet Society from 1992–1995 and is currently chairman of the Board. Cerf holds a Bachelor of Science degree in Mathematics from Stanford University and Master of Science and Ph.D. degrees in Computer Science from UCLA. E-mail: **vcerf@mci.net**

Book Reviews

Internet Messaging

Internet Messaging: From the Desktop to the Enterprise, by Marshall T. Rose and David Strom ISBN 0-13-978610-4, Prentice-Hall PTR, 1998, **http://www.prenhall.com**

Very few Internet voices hold a status equivalent to E.F. Hutton's advertising campaign: "When they speak, we should listen." Marshall Rose and David Strom are two such voices, making any product of their combined efforts a serious matter, indeed. Rose has typically written about basic technology, Strom about the pragmatics of use, especially trials and tribulations of fitting networked pieces together. *Internet Messaging* is in the latter category, with a strong added introduction of e-mail and security technology. Anyone who has professional contact with e-mail should get a copy of this book. If commercial use of Internet mail were more advanced and stable, we probably would not need an effort like this. However, e-mail professionals must constantly deal with problems in using interesting functions and in troubleshooting interoperability. *Internet Messaging* helps with the planning, use and debugging of complex, or otherwise "interesting," e-mail services.

Updated Information

The book provides a superb survey of the relevant technology, the popular user mail software, and the rather interesting range of mail and messaging operations issues, including styles of use by organizations. The comparisons of different mail systems leave the reader with a solid understanding of functional and usage requirements for modern systems, as well as the choices available at the time of publication. Mary Houten-Kemp's Web site at **http://www.everythingemail.net** is being used to provide updated information.

E-mail includes a wide range of technical and operations issues, and *Internet Messaging* touches all of them. Its introductions cover user environment, mail transfer, mailing list services, unsolicited bulk e-mail ("spam"), encryption-based security, remote user access, virtual private networks, and directory services. Providing a single discussion, which integrates the use of these disparate technologies, is enough to justify the book.

Organization

Internet Messaging attempts very regular organization and states that the goal is to permit use as a problem/solution reference work. It primarily distinguishes between sending and receiving functions and between desktop and enterprise requirements. This creates a two-by-two matrix, defining the core four chapters. The other chapters include philosophical opening and closing discussions, a separate, very informative chapter on security, and another on general enterprise operations issues.

Most of the chapters are organized into Introduction, Problems, Standards, and Solutions. Unfortunately that regularization is all that is shown in the Table of Contents, so the reader gets little help finding specifics by reading the Table. Similarly, the organization of the chapter contents did not seem compelling for use in problem solving. The additional "How Can I" matrix (on page 10) and its associated discussion text is intended as the primary means for locating relevant discussions.

Comparisons

User software comparisons are given throughout the book, for Microsoft Outlook 4.01, Netscape Messenger 4.04, Qualcomm Eudora Pro 4.0, Lotus cc:Mail 8.1, CompuServe WinCIM 3.02, and America Online 3.0. Specific mailing lists, security, remote access, and directory software and services are also reviewed. Oddly, the discussion of remote access mentions only global, single-provider services—and their favorite is currently having financial problems—but did not mention the "association" style of service that integrates many independent providers, notably GRIC and iPass. (Full disclosure: iPass is a client.)

Most products are undergoing aggressive enhancement so that no printed text can be entirely up-to-date. Hence the Web site. For the software and services I know well, the book looked reasonable. Of course it is not entirely error free, but the errors are small and perfect detail is not required. I believe there are two major benefits to these comparisons. One is that the reader is given a very solid sense of the general capabilities and limitations of modern e-mail software. The second is to make a reasonable, first-pass filtering of candidate packages to be used in an organization. It would *not* be appropriate to attempt selecting among these packages according to subtle differences reported in the book.

Benefits

As one would expect of these authors, a very large, long-term benefit of their efforts is in their many excellent criticisms and suggestions. Unfortunately, many of them are in notes located at the end of each chapter. It's hard to imagine a less-convenient place to put them, since I found myself constantly shifting back and forth between the main text and the notes. It would not have been so irritating if the comments were less interesting; they should have been true footnotes, with easy access on each page. The stellar example of direct utility from these comments is Figure 2.1 on page 38. It shows a systems structure for user software processing of incoming mail. Every vendor should study this discussion carefully and implement it immediately. Please!

—Dave Crocker
Brandenburg Consulting
dcrocker@brandenburg.com

Web Security

Web Security: A Step-by-Step Reference Guide, by Lincoln D. Stein, ISBN 0-201-63489-9, Addison-Wesley, December 1997,
http://www.awl.com/cseng/titles/0-201-63489-9

Whenever the topic of the World Wide Web comes up, you can be sure that some mention of "security" will soon follow. Web users, Web creators, and even Web technology developers are all keenly aware of the security concerns. But what do we mean by "security?" The safety to use a credit card? Keeping a Web site safe from break-ins? Keeping the kids away from online erotica? And whose security are we concerned with, the user's or the Web site operator's?

This book covers most of what we might expect to find under the umbrella of security. In addition to dealing with the broad scope of Web security, the author also tries to cover the topic with sufficient simplicity for the novice and enough detail for the engineer. The good news is that this book succeeds in delivering a single volume that covers all we could possibly expect on the topic, and at levels suited for a broad audience range.

Organization

The author begins by making the distinction between security for the browser, the Web site, and the network between them. This division of the topic forms the basis for the organization of the book. Moving through each of the three parts, the author proceeds from the simple to the complex in a logical, additive order. He discusses topics introduced early in the book from a functional standpoint—how they affect the user. He may cover the same technology in later chapters, but in greater depth, detailing server and network configuration and discussing the underlying technology.

In the first part of the book, the author covers document confidentiality, including standard "text" documents as well as electronic commerce. A major theme in this section is cryptography. The author presents symmetric and public key encryption technologies from a functional standpoint. He presents various encryption standards, with a discussion of their strengths and weaknesses. In another chapter he provides a good primer on the *Secure Electronic Transaction* (SET) protocol handling, as well as other options (*Common Gateway Interface* [CGI] scripts and *Secure Sockets Layer* [SSL]) for credit card order processing.

In Part 2 we are introduced to issues of client-side security. The author devotes a full chapter to an in-depth explanation of SSL services. He also looks at issues associated with active content, and presents technologies such as Java, ActiveX, and other options, along with notes on their respective security implications. Finally, he covers issues of privacy—in this case, the personal privacy of the user. Throughout these chapters, the author emphasizes user-controllable settings such as browser configuration options.

Whereas the author focuses on user involvement in the first two parts, with an appropriate level of technical content, in part 3, targeted to Web masters and system administrators, he introduces the engineering side with an in-depth coverage of server-side security. He covers the two prominent Web-serving operating systems: UNIX and Windows NT, with good attention to various versions of each. Topics include basic system security, access control, and activity monitoring. Other chapters include an excellent discussion of encryption and certificate technology, safe CGI scripting, remote authoring of Web data, and firewalls.

Presentation and Style

The author illustrates his points with good examples. He also presents appropriate sidebar discussions and illustrations, which not only clarify the information, but also provide interest and variety in what could be a very dry volume. Each chapter ends with a listing of resources, both print and "online." Where appropriate, the author includes checklists to help the reader apply the material just covered.

As a result of the practical, well-grounded presentation of material, we are continually able to see practical applicability to our own situation. For example, the author presents us with information about dangers to our privacy, and why that might be important to us. This is immediately followed by clear instruction on changing privacy-affecting settings in various versions of both Netscape and Internet Explorer. The author uses this technique throughout the book, and it is as useful with password management, CGI scripting, or firewall configuration as it is with privacy.

Recommended

Although experts in encryption and other specific security-related technologies will find this book too simple for their personal area of expertise, the strength of the book is not in its coverage of any one area, but in its well-integrated and cohesive coverage of a broad range of interrelated topics. The ability for any reader, first-time surfer or Web guru, to find practical, easily applied information makes this book a required item on any webmaster's bookshelf, and a must-read for anyone who spends any serious time on the Web.

—Richard Perlman
Berkeley Internet Group
perl@berkinet.com

Internet Cryptography

Internet Cryptography, by Richard E. Smith, ISBN 0-201-92480-3, Addison-Wesley, 1998, **www.awl.com/cseng/titles/0-201-92480-3**

The 1990s might easily be known as the decade of the Internet. The Internet came into the mainstream during this decade, a global frontier with frontier problems and rules. Seemingly overnight, everyone from government agencies to Chinese restaurants had a Web presence. Young children exchanged e-mail with their grandparents and friends, a big change from just a few years ago when it was the domain of technologies and a place where everybody knew your name.

The 1990s could also be known as the decade when cryptography became mainstream. Perhaps because of the change in the Internet community, people became more aware of the need to protect the privacy of internetwork communications. Certainly, the U.S. government's attempt to push government control of cryptographic keys in the Clipper controversy helped to move cryptography and its related issues from science journals to the front pages of our newspapers. Today, while not mainstream, terms such as *Virtual Private Networks* (VPNs), *Secure Sockets Layer* (SSL), *IP Security* (IPSec), *Pretty Good Privacy* (PGP), *Secure Multipurpose Internet Mail Extensions* (S/MIME), and related technologies are known among IT professionals, and cryptography is no longer a tool used only by spies and military communication officers.

The Author

Richard E. Smith is well-known to members of various security-related forums on the Internet, as well as to security conference attendees. A security consultant with Secure Computing Corporation, Smith's background is in military-grade security. His experience on the lecture circuit, explaining issues of firewalls, cryptography, and other computer and network security topics, has directly contributed to production of a book on a lofty subject that is reachable by the nonscientist.

Organization

The chapters of this book fall into three groupings: an introduction to the basics of cryptography, its terms, methods, and mechanisms; network encryption and a discussion of VPNs, focusing on IPSec; and finally public key cryptography as it is used with message and file encryption and "Web" transactions.

The discussion in the opening chapter on basics may scare some off; Smith tends to oscillate between various levels of complexity. Consequently, some members of the intended audience of (quoting from the Preface) "people who know very little about cryptography but need to make technical decisions about cryptographic security," may, for example, zone out during the discussion of IP protocols. My suggestion would be to press on, and not worry about the random item that might go over your head. Everything there has a purpose, and the important information will fall into place by the end of each chapter.

If this book ended with Chapter 4, it would still be a useful book. The complex basics of cryptography and the issues that should be of concern to an information security officer are clearly presented and explained. The only area that is given less than adequate coverage is that of key recovery. Smith makes no mention of legitimate business reasons for the recovery of encrypted data if the originator is unavailable (the proverbial question, "What if you got hit by a truck?"), nor does he mention any mechanism other than the escrow of secret keys, although there are other, safer, methods. Of particular use are Smith's explanations of the various cryptographic algorithms and his discussions of safe key lengths and risks.

In the sections on VPNs and IPSec, Smith covers everything from mobile users and remote access, to point-to-point encryption, and the issues of key distribution, exchange, and the mechanisms used to automate encrypted communication. Everyone seems to know that IPSec will save the world and is the answer to all our security problems (and I have my tongue firmly planted in my cheek), but few know what IPSec really does, from a "features and benefits" point of view. Of particular use and interest are the sections labeled "Deployment Example." These are small case studies that show the technology in action and discuss some of the decisions and processes that came before deployment.

The section covering public key cryptography along with file and message encryption is perhaps shorter than it should be, although much of the groundwork is done earlier in the book. Missing is a "how to" on setting up a public key infrastructure (PKI) for a corporation to use. There are "Product Examples" in this section, but not "Deployment Examples." Perhaps those will have to wait for a second edition, for although this is a lack in the book, there are not many real-life examples from which to choose. Although discussed in theory for years, this is still "leading edge" in the real world. The chapter on Web servers should prove informative and useful to any organization thinking of deploying (or having already deployed) a Web server.

In the chapter entitled "Secure Electronic Mail," the fact that Smith covers *Privacy Enhanced Mail* (PEM) as a technology more than he covers S/MIME is puzzling, but the basics of PEM are useful for discussion, even if PEM as a technology seems to be dead.

Cryptography Is Necessary

The advertisement on the back of the book (not written by the author, of course) states "Here, in one comprehensive, soup-to-nuts book, is the soution for Internet security: modern-day cryptography." Obviously the claim that cryptography is *the* solution for Internet security is way overinflated; modern-day cryptography is not *the* solution, but, cryptography is an important part of a "balanced" security solution. Smith does an admirable job of making this heretofore…well, cryptic… subject, understandable, interesting, and even enjoyable.

—*Frederick M. Avolio, Avolio Consulting,* **fred@avolio.com**

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and quality of service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

Fragments

ICANN

The *Internet Corporation for Assigned Names and Numbers* (ICANN) was incorporated in late October. ICANN is a private, non-profit corporation, managed by an international board, formed to coordinate and administer policies and technical protocols relating to the domain name and address system that permits Internet communications to be routed to the correct person or entity. Its proposed duties include those now performed under U.S. Government contract by the *Internet Assigned Numbers Authority* (IANA), whose Director, Internet pioneer Jon Postel, died on October 16th. ICANN has elected its Initial Board and chosen Michael M. Roberts as its Interim President and Chief Executive Officer. In addition, the Board chose Esther Dyson as its Interim Chairman, and appointed an Executive Committee consisting of Dyson, Gregory L. Crew, Hans Kraaijenbrink and Roberts. The other Initial Board members include Geraldine Capdeboscq (France), George H. Conrades (United States), Gregory L. Crew (Australia), Frank Fitzsimmons (United States), Hans Kraaijenbrink (The Netherlands), Jun Murai (Japan), Eugenio Triana (Spain), and Linda S. Wilson (United States). ICANN was originally proposed by Postel on behalf of a broad coalition of Internet stakeholders in response to the request by the U. S. Government last June that the Internet community create a global consensus non-profit corporation to which the U.S. could transition the responsibility for overseeing and funding those coordination activities. For more information, see:
`http://www.iana.org/index2.html`

APRICOT '99

The *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will be held in Singapore, March 1–5, 1999. APRICOT provides a forum for key Internet builders in the region to learn from their peers and other leaders in the Internet community from around the world. The week-long summit consists of seminars, workshops, tutorials, conference sessions, birds-of-a-feather sessions, and other forums—all with the goal of spreading and sharing the knowledge required to operate the Internet within the Asia Pacific region. For more information, see: **`http://www.apricot.net`**

Send us your comments!

We look forward to hearing your comments and suggestions regarding anything you read in this publication. Send us e-mail at: **`ipj@cisco.com`**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Cisco News Publications Group, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-J4
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

March 1999 Volume 2, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download previous issues of IPJ in PDF format from: www.cisco.com/ipj

From The Editor

Today's Internet is comprised of numerous interconnected *Internet Service Providers* (ISPs), each serving many constituent networks and end users. Just as individual regional and national telephone companies interconnect and exchange traffic and form a global telephone network, the ISPs must arrange for points of interconnection to provide global Internet service. This interconnection mechanism is generally called "peering," and it is the subject of a two-part article by Geoff Huston. In Part I, which is included in this issue, he discusses the technical aspects of peering. In Part II, which will follow in our next issue, Mr. Huston continues the examination with a look at the business arrangements (called "settlements") that exist between ISPs, and discusses the future of this rapidly evolving marketplace.

In the early 1990s, concern grew regarding the possible depletion of the IP version 4 address space because of the rapid growth of the Internet. Predictions for when we would literally run out of IP addresses were published. Several proposals for a new version of IP were put forward in the IETF, eventually resulting in IP version 6 or IPv6. At the same time, new technologies were developed that effectively slowed address depletion, most notably *Classless Inter-Domain Routing* (CIDR) and *Network Address Translators* (NATs). Today there is still debate as to if and when IPv6 will be deployed in the global Internet, but experimentation and development continues on this protocol. We asked Robert Fink to give us a status report on IPv6.

We've already discussed the historical lack of security in Internet technologies and how security enhancements are being developed for every layer of the protocol stack. This time, Marshall Rose and David Strom examine the state of electronic mail security. We clearly have a way to go before we see "seamless integration" of security systems with today's e-mail clients.

Our first Letter to the Editor is included on page 46. As always, we would love to hear your comments and questions regarding anything you read in this journal. Please contact us at ipj@cisco.com

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Interconnection, Peering and Settlements—Part I

by Geoff Huston, Telstra

Technology and business models share a common evolution within the Internet. To enable deployment of the technology within a service environment, a robust and stable business model also needs to be created. This tied destiny of technology and business factors is perhaps most apparent within the area of the interconnection of *Internet Service Providers* (ISPs). Here there is an interaction at a level of technology, in terms of routing signaling and traffic flows, and also an interaction of business models, in terms of a negotiation of benefit and cost in undertaking the interconnection. This article examines this environment in some detail, looking closely at the interaction between the capabilities of the technical protocols, their translation into engineering deployment, and the consequent business imperatives that such environments create.

It is necessary to commence this examination of the public Internet with the observation that the Internet is not, and never has been, a single network. The Internet is a collection of interconnected component networks that share a common addressing structure, a common view of routing and traffic flow, and a common view of a naming system. This interconnection environment spans a highly diverse set of more than 50,000 component networks, and this number continues, inexorably, to grow and grow. One of the significant aspects of this environment is the competitive Internet service industry, where many thousands of enterprises, both small and large, compete for market share at a regional, national, and international level.

Underneath the veneer of a highly competitive Internet service market is a somewhat different environment, in which every ISP network must interoperate with neighboring Internet networks in order to produce a delivered service outcome of comprehensive connectivity and end-to-end service. No ISP can operate in complete isolation from others while still offering public Internet services, and therefore, every ISP not only must coexist with other ISPs but also must operate in cooperation with other ISPs.

This article examines both the technical and business aspects that surround this ISP interaction, commonly referred to as "interconnection, peering, and settlements." It examines the business motivation for interconnection structures, and then the technical architectures of such environments. The second part looks at the business relationships that arise between ISPs in the public Internet space, and then examines numerous broader issues that will shape the near-term future of this environment.

[This article is based in part on material in *The ISP Survival Guide*, by Geoff Huston, ISBN 201-3-45567-9, published by Wiley. Used with permission.]

Interconnection: Retailing, Reselling, and Wholesaling

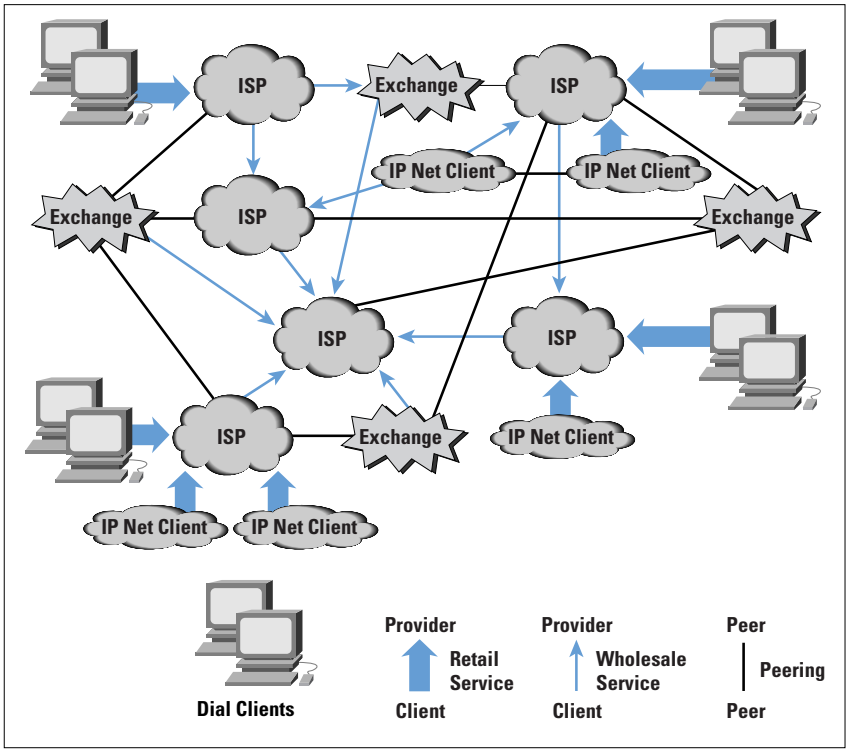
To provide some motivation for this issue of ISP interconnection, it is first appropriate to look at the nature of the environment. The regulatory framework that defined the traditional structure of other communications enterprises such as telephony or postal services was largely absent in the evolution of the Internet service industry. The resultant service industry for the Internet is most accurately characterized as an outcome of business and technology interaction, rather than a planned outcome of some regulatory process. This section examines this interaction between business and technology within the ISP environment.

A natural outcome of the Internet model is that the effective control of the retail service environment rests with a network client of an access service rather than with the access service provider. As such, a client of an ISP access service has the discretionary ability to resell the access service to third-party clients. In this environment, reselling and wholesaling are very natural developments within the ISP activity sector, with or without the explicit concurrence of the provider ISP. The provider ISP may see this reselling as an additional channel to market for its own Internet carriage services, and may adopt a positive stance by actively encouraging resellers into the market as a means of overall market stimulus, while tapping into the marketing, sales, and support resources of these reselling entities to continue to drive the volumes of the underlying Internet carriage service portfolio. The low barriers to entry to the wholesale market provide a means of increasing the scope of the operation, because to lift business cash-flow levels, the business enters into wholesale agreements that effectively resell the carriage components of the operation without the bundling of other services normally associated with the retail operation. This process allows the ISP to gain higher volumes of carriage capacity that in turn allow the ISP to gain access to lower unit costs of carriage.

Given that a retail operation can readily become a wholesale provider to third-party resellers at the effective discretion of the original retail client, is a wholesale transit ISP restricted from undertaking retail operations? Again, there is no such natural restriction from a technical or business perspective. An Internet carriage service is a commodity service that does not allow for a significant level of intrinsic product discrimination. The relatively low level of value added by a wholesale service operation implies a low unit rate of financial return for that operation. This low unit rate of financial return, together with an inability to competitively discriminate the wholesale product effectively, induces a wholesale provider into the retail sector as a means of improving the financial performance of the service operation. The overall result is that many ISPs operate both as clients and as providers. Few, if any, reasonable technical-based characterizations draw a clear and unambiguous distinction between a client and service provider when access services to networks are considered. A campus network may be a client of one or more ser-

vice providers, while the network is also a service provider to campus users. Indeed most networks in a similar situation take on the dual role of client and provider, and the ability to resell an access service can extend to almost arbitrary depths of the reselling hierarchy. From this technical perspective, very few natural divisions of the market support a stable segmentation into exclusively wholesale and exclusively retail market sectors. The overall structure of roles is indicated in Figure 1.

Figure 1: ISP Roles and Relationships



The resultant business environment is one characterized by a reasonable degree of fluidity, in which no clear delineation of relative roles or markets exists. The ISP market environment is, therefore, one of competitive market forces in which each ISP tends to create a retail market presence. However, no ISP can operate in isolation. Each client has the expectation of universal and comprehensive reachability, such that any client of any other ISP can reach the client, and the client can reach a client of any other ISP. The client of an ISP is not undertaking a service contract that limits connectivity only to other clients of the same ISP. Because no provider can claim ubiquity of access, every provider relies on every other provider to complete the user-provided picture of comprehensive connectivity. Because of this dependent relationship, an individual provider's effort to provide substantially superior service quality may have little overall impact on the totality of client-delivered service quality. In a best-effort public Internet, the service quality becomes something that can be impacted negatively by poor local engineering but cannot be uniformly improved beyond the quality provided by the network's peers, and their peers in turn. Internet wholesale carriage services in such an environment are constrained to be a com-

modity service, in which scant opportunity exists for service-based differentiation. In the absence of service quality as an effective service discriminator, the wholesale activity becomes a price-based service with low levels of added value, or in other words a commodity market.

The implication in terms of ISP positioning is that the retail operation, rather than the wholesale activity, is the major area in which the ISP can provide discriminating service quality. Within the retail operation, the ISP can offer a wide variety of services with a set of associated service levels, and base a market positioning on factors other than commodity carriage pricing.

Accordingly, the environment of interconnection between ISPs does not break down into a well-ordered model of a set of wholesale carriage providers and associated retail service providers. The environment currently is one with a wide diversity of retail-oriented providers, where each provider may operate both as a retail service operator, and a wholesale carriage provider to other retailers.

Peer or Client?

One of the significant issues that arises here is: Can an objective determination be made of whether an ISP is a peer to, or a client of, another ISP? This is a critical question, because if a completely objective determination cannot be readily made, the question then becomes one of who is responsible for making a subjective determination, and on what basis.

This question is an inevitable outcome of the reselling environment, where the reseller starts to make multiple upstream service contracts, with a growing number of downstream clients of the reselling service. At this point, the business profile of the original reseller is little distinguished from that of the original provider. The original reseller sees no unique value being offered by the original upstream provider and may conclude that it is, in fact, adding value to the original upstream provider by offering the upstream provider high-volume carriage and close access to the reseller's client base. From the perspective of the original reseller, the roles have changed, and the reseller now perceives itself as a peer ISP to the original upstream ISP provider.

This assertion of role reversal is perhaps most significant when the generic interconnection environment is one of "zero-sum" financial settlement, in which the successful assertion by a client of a change from client to peer status results in the dropping of client service revenue without any net change in the cost base of the provider's operation. The party making the successful assertion of peer interconnection sees the opposite, with an immediate drop in the cost of the ISP operation with no net revenue change.

The traditional public regulatory resolution of such matters has been through an administrative process of "licensed" communications service providers, who become peer entities through a process of

administrative fiat. In this model, an ISP becomes a licensed service provider through the payment of license fees to a communications regulatory body. The license then allows the service enterprise access to interconnection arrangements with other licensed providers. The determination of peer or client is now quite simple: A *client* is an entity that operates without such a carrier license, and a *peer* is one that has been granted such an instrument. However, such regulated environments are quite artificial in their delineation of the entities that operate within a market, and this regulatory process often acts as a strong disincentive to large-scale private investment, thereby placing the burden of underwriting the funding of service industries into the public sector. The regulatory environment is changing worldwide to shift the burden of communications infrastructure investment from the public sector, or from a uniquely positioned small segment of the private sector, to an environment that encourages widespread private investment. The Internet industry is at the leading edge of this trend, and the ISP domain typically operates within a deregulated valued-added communications service provider regulatory environment. Individual licenses are replaced with generic class licenses or similar deregulated structures in which formal applications or payments of license fees to operate in this domain are unnecessary. In such deregulated environments, no authoritative external entity makes the decision as to whether the relationship between two ISPs is that of a provider and client or that of peers.

If no public regulatory body wants to make such a determination, is there a comparable industry body that can undertake such a role? The early attempts of the *Commercial Internet eXchange* (CIX) arrangements in the United States in the early 1990s were based on a description of the infrastructure of each party, in which acknowledgments of peer capability were based on the operation of a national transit infrastructure of a minimum specified capability. This specification of peering within the CIX was subsequently modified so that CIX peer status for an ISP was simply based on payment of the CIX Association membership fee.

This CIX model was not one that intrinsically admitted bilateral peer relationships. The relationship was a multilateral one, in which each ISP executed a single agreement with the CIX Association and then effectively had the ability to peer with all other association member networks. The consequence of this multilateral arrangement is that the peering settlements can be regarded as an instance of "zero-sum" financial settlement peering, using a single-threshold pricing structure.

Other industry models use a functional peer specification. For example, if the ISP attaches to a nominated physical exchange structure, then the ISP is in a position to open bilateral negotiations with any other ISP also directly attached to the exchange structure. This model is inherently more flexible, as the bilateral exchange structure enables each represented ISP to make its own determination of whether to agree to a peer

relationship or not with any other colocated ISP. This model also enables each bilateral peer arrangement to be executed individually, admitting the possibility of a wider diversity of financial settlement arrangements.

The bottom line is that a true peer relationship is based on the supposition that either party can terminate the interconnection relationship and that the other party does not consider such an action a competitively hostile act. If one party has a high reliance on the interconnection arrangement and the other does not, then the most stable business outcome is that this reliance is expressed in terms of a service contract with the other party, and a provider/client relationship is established. If a balance of mutual requirement exists between both parties, then a stable basis for a peer interconnection relationship also exists. Such a statement has no intrinsic metrics that allow the requirements to be quantified. Peering in such an environment is best expressed as the balance of perceptions, in which each party perceives an acceptable approximation of equal benefit in the interconnection relationship in its own terms.

This conclusion leads to the various tiers of accepted peering that are evident in the Internet today. Local ISPs see a rationale to viewing local competing ISPs as peers, and they still admit the need to purchase trunk transit services from one or more upstream ISPs under terms of a client contract with the trunk provider ISP. Trunk ISPs see an acceptable rationale in peering with ISPs with a similar role profile in trunk transit but perceive an inequality of relationship with local ISPs. The conclusion drawn here is that the structure of the Internet is one in which there is a strong business pressure to create a rich mesh of interconnection at various levels, and the architecture of interconnection structures is an important feature of the overall architecture of the public Internet.

Physical Interconnection Architectures: Exchanges and NAPs

One of the physical properties of electromagnetic propagation is that the power required to transmit an electromagnetic pulse over a distance varies in accordance with this distance. The shorter the distance between the transmitter and the receiver, the lower the transmission power budget required; *closer is cheaper*.

This statement holds true not only for electrical power budgets but also for data protocol efficiency. Minimizing the delay between the sender and receiver allows the protocol to operate faster and operate more efficiently as well; *closer is faster,* and *closer is more efficient*.

These observations imply that distinct and measurable advantages are gained by localizing data traffic; that is, by ensuring that the physical path traversed by the packets passed between the sender and the receiver is kept as physically short as possible. These advantages are realizable in terms of service performance, efficiency, and service cost.

How then are such considerations of locality factored into the structure of the Internet?

The Exchange Model

A strictly hierarchical model of Internet structure is one in which a small number of global ISP transit operators is at the "top;" a second tier is of national ISP operators; and a third tier consists of local ISPs. At each tier, the ISPs are clients of the tier above, as shown in Figure 2. If this hierarchical model is strictly adhered to, traffic between two local ISPs is forced to transit a national ISP, and traffic between two national ISPs transits a global ISP—even if both national ISPs operate within the same country. In the worst case, traffic between two local ISPs needs to transit a national ISP, then a global ISP from one hierarchy, then a second global ISP, and a second national ISP from an adjacent hierarchy in order to reach the other local ISP. If the two global providers interconnect at a remote location, the transit path of the traffic between these two local ISPs could be very long indeed.

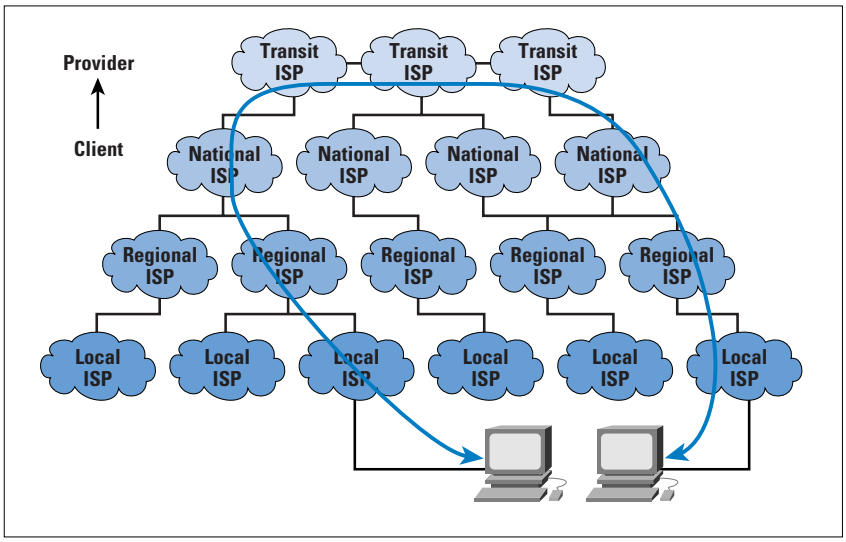
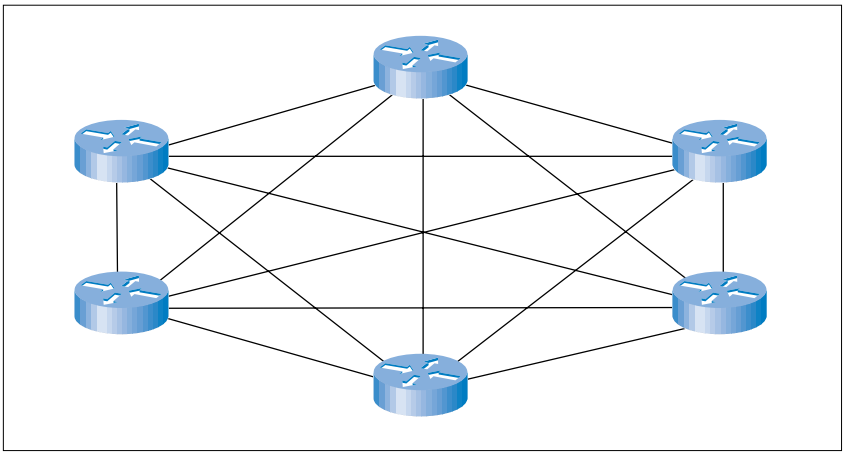


Figure 2: A Purely Hierarchical Structure for the Internet

As noted above, such extended paths are inefficient and costly, and such costs are ultimately part of the cost component of the price of Internet access. In an open, competitive market, strong pressure always is applied to reduce costs. Within a hierarchical ISP environment, strong pressure is applied for the two national providers, who operate within the same market domain, to modify this strict hierarchy and directly interconnect their networks. Such a local interconnection allows the two networks to service their mutual connectivity requirements without payment of transit costs to their respective global transit ISP providers. At the local level is a similar incentive for the local ISPs to reduce their cost base, and a local interconnection with other local ISPs would allow local traffic to be exchanged without the payment of transit costs to the respective transit providers.

Although constructing a general interconnection regime based on point-to-point bilateral connections is possible, this approach does not exhibit good scaling properties. Between N providers who want to interconnect, the outcome of such a model of single interconnecting circuits is $(N^2 - N) / 2$ circuits and $(N^2 - N) / 2$ routing interconnections, as indicated in Figure 3. Given that interconnections exhibit the greatest leverage within geographical local situations, simplifying this picture within the structure of a local exchange is possible. In this scenario, each provider draws a single circuit to the local exchange and then executes interconnections at this exchange location. Between N providers who want to interconnect, the same functionality of complete interconnection can be constructed using only N point-to-point circuits.

Figure 3: Fully Meshed Peering



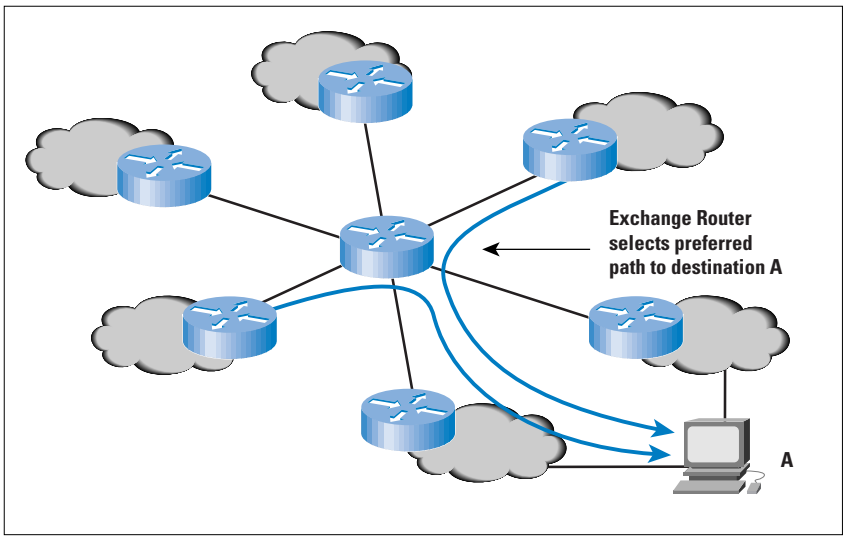
The Exchange Router

One model of an exchange is to build the exchange itself as a router, as indicated in Figure 4. Each provider's circuit terminates on the exchange router, and each provider's routing system peers with the routing process on the exchange router. This structure also simplifies the routing configuration, so that full interconnection of N providers is effected with N routing peer sessions. This simplification does allow greater levels of scaling in the interconnection architecture.

However, the exchange router model becomes an active component of the interconnect peering policy environment. In effect, each provider must execute a multilateral interconnection peering with all of the other connected providers. Selectively interconnecting with a subset of the providers present at such a router-based exchange is not easily achieved. In addition, this type of exchange must execute its own routing policy. When two or more providers are advertising a route to the same destination, the exchange router must execute a policy decision as to which provider's route is loaded in the router's forwarding table, making a policy choice of transit provider on behalf of all other exchange-connected providers.

Because the exchange is now an active policy element in the interconnection environment, the exchange is no longer completely neutral to all participants. This imposition on the providers may be seen as unacceptable, in that some of their ability to devise and execute an external transit policy is usurped by the exchange operator's policies.

Figure 4: An Exchange Router



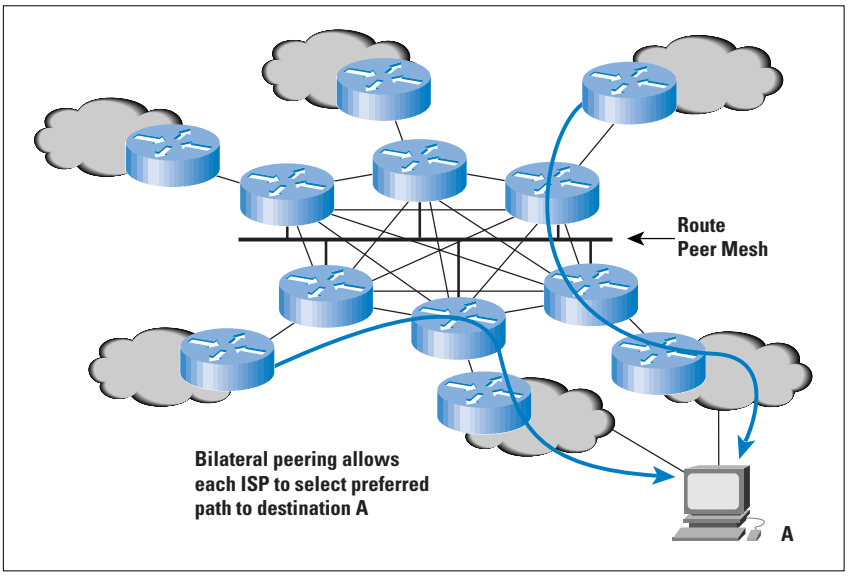
Typically, providers have a higher expectation of flexibility of policy determination from exchange structures than the base level of functionality that is provided by an exchange router. Providers want the flexibility to execute interconnections on a bilateral basis at the exchange, and to make policy decisions as to which provider to prefer when the same destination is advertised by multiple providers. They require the exchange to be neutral with respect to such individual routing policy decisions.

The Exchange Switch

The modification to the interprovider exchange structure is to use a local Layer 2 switch (or LAN) as the exchange element. In this model, a participating provider draws a circuit to the exchange and locates a dedicated router on the exchange LAN, as shown in Figure 5. Each provider executes a bilateral peering agreement with another provider by initiating a router peering session with the other party's router. When the same network destination is advertised by multiple peers, the provider can execute a policy-based preference as to which peer's route will be loaded in the local forwarding table. Such a structure preserves the cost efficiency of using N circuits to effect interconnection at the N provider exchange, while admitting the important policy flexibility provided by up to $(N^2 - N) / 2$ potential routing peer sessions.

Early interprovider exchanges were based on an Ethernet LAN as the common interconnection element. This physical structure was simple, and not all that robust under the pressures of growth as the LAN became congested.

Figure 5:
An Exchange LAN



Subsequent refinements to the model have included the use of Ethernet switches as a higher capacity LAN, and the use of *Fiber Distributed Data Interface* (FDDI) rings, switched FDDI hubs, Fast Ethernet hubs, and switched Fast Ethernet hubs. Exchanges are very-high-traffic concentration points, and the desire to manage ever-higher traffic volumes has led to the adoption of Gigabit Ethernet switches as the current evolutionary technology step within such exchanges.

The model of the exchange colocation accommodates a model of diversity of access media, in which the provider's colocated router undertakes the media translation between the access link protocol and the common exchange protocol.

The local traffic exchange hub does represent a critical point of failure within the local Internet topology. Accordingly, the exchange should be engineered in the most resilient fashion possible, using standards associated with a premium quality data center. This structure may include multiple power utility connections, uninterruptible power supplies, multiple trunk fiber connections, and excellent site security measures.

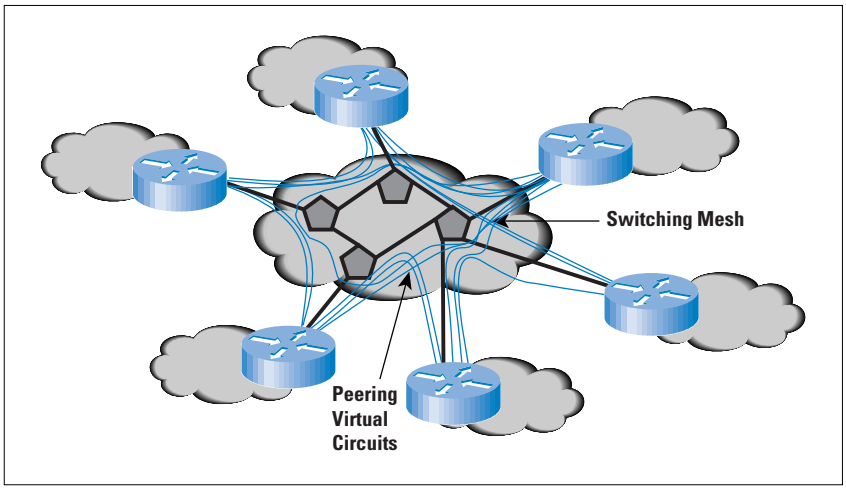
The exchange should operate neutrally with respect to every participating ISP, with the interests of all the exchange clients in mind. Thus, exchange facilities, which are operated by an entity that is not also a local or trunk ISP, enjoy higher levels of trust from the clients of the exchange.

There are also some drawbacks to an exchange, and a commonly cited example is that of imposed transit. If an exchange participant directs a default route to another exchange router, then in the absence of defensive mechanisms, the target router carries the imposed transit traffic even when there is no routing peering or business agreement between the two ISPs. Exchange-located routers do require careful configuration management to ensure that route peering and associated transit traffic matches the currently executed interconnection agreements.

Distributed Exchanges

Distributed exchange models also have been deployed in various locations. This deployment can be as simple as a metropolitan FDDI extension, in which the exchange comes to the provider's location rather than the reverse, as indicated in Figure 6. Other models that use an ATM-based switching fabric also have been deployed using *LAN Emulation* (LANE) to mimic the Layer 2 exchange switch functionality. Distributed exchange models attempt to address the significant cost of operating a single colocation environment with a high degree of resilience and security, but do so at a cost of enforcing the use of a uniform access technology between every distributed exchange participant.

Figure 6: A Distributed Exchange



However, the major challenge of such distributed models is that of switching speed. Switching requires some element of contention resolution, in which two ingress data elements that are addressed to a common egress path require the switch to detect the resource contention and then resolve it by serializing the egress. Switching, therefore, requires signaling, in which the switching element must inform the ingress element of switch contention. To increase the throughput of the switch, the latency of this signaling must be reduced. The dictates of increased switching speed have the corollary of requiring the switch to exist within the confines of a single location, if exchange performance is a paramount concern.

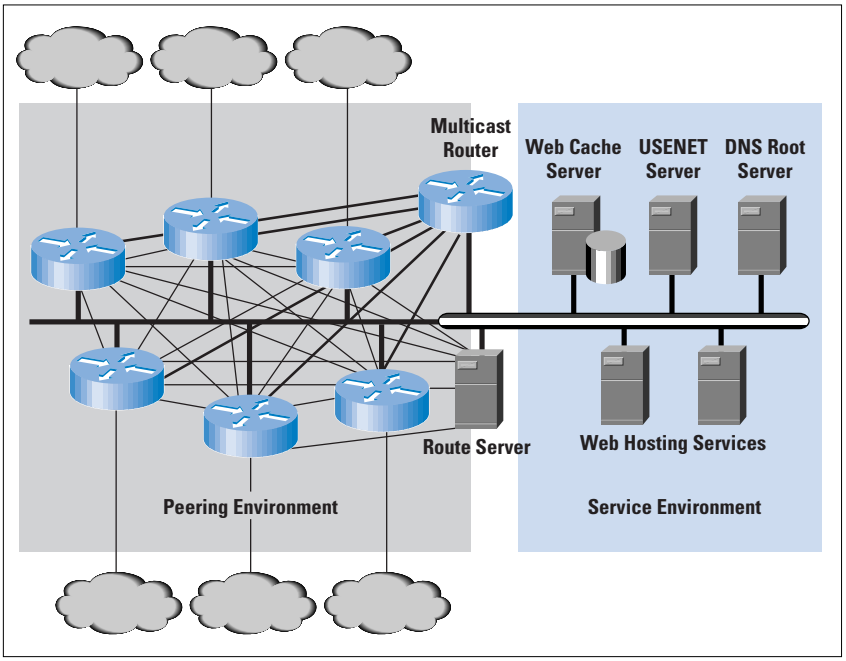
In addition to speed, the cost shift must be considered. In a distributed exchange model, the exchange operator operates the set of access circuits that form the distributed exchange. This process increases costs to providers, while it prevents the providers from using a specific access technology that matches their business requirements of cost and supportable traffic volume. Not surprisingly, to date the most prevalent form of exchange remains the third-party hosted colocation model. This model admits a high degree of diversity in access technologies, while still providing the substrate of an interconnection environment that can operate at high speed and therefore manage high traffic volumes.

Other Exchange-Located Services

The colocation environment is often broadened to include other functions, in addition to a pure routing and traffic exchange role. For a high-volume content provider, the exchange location offers minimal transit distance to a large user population distributed across multiple local service providers, as well as allowing the content provider to exercise a choice in selecting a nonlocal transit provider.

The exchange operator can also add value to the exchange environment by providing additional functions and services, as well as terminating providers' routers and large-volume content services. The exchange location within the overall network topology is an ideal location for hosting multicast services, because the location is optimal in terms of multicast carriage efficiency. Similarly, USENET trunk feed systems can exploit the local hub created by the exchange. The overall architecture of a colocation environment that permits value-added services, which can productively use the unique environment created at an exchange, is indicated in Figure 7.

Figure 7: Exchange-Located Service Platforms



Network Access Points

The role of the exchange was broadened with the introduction of the *Network Access Point* (NAP) in the architecture proposed by the National Science Foundation (NSF) in 1995 when the NSFNET backbone was being phased out.

The NAP was seen to undertake two roles: the role of an exchange provider between regional ISPs who want to execute bilateral peering arrangements and the role of a transit purchase venue, in which regional ISPs could execute purchase agreements with one or more of a set of trunk carriage ISPs also connected at the NAP. The access point concept was intended to describe access to the trunk transit service.

This mixed role of both local exchange and transit operations leads to considerable operational complexity, in terms of the transit providers being able to execute a clear business agreement. What is the bandwidth of the purchased service in terms of requirements for trunk transit, versus the access requirements for exchange traffic? If a local ISP purchases a transit service at one of the NAPs, does that imply that the trunk provider is then obligated to present all the ISP's routes at remote NAPs as a peer? How can a trunk provider distinguish between traffic presented to it on behalf of a remote client versus traffic presented to it by a local service client?

The issue that the quality of the purchased transit service is colored by the quality of the service provided by the NAP operator should also be considered. Although the quality of the transit provider's network may remain constant, and the quality of the local ISP's network and ISP's NAP access circuit may be acceptable, the quality of the transit service may be negatively impacted by the quality of the NAP transit itself.

One common solution is to use the NAP colocation facility to execute transit purchase agreements and then use so-called *backdoor* connections for the transit service provision role. This usage restricts the NAP exchange network to a theoretically simpler local exchange role. Such a configuration is illustrated in Figure 8.

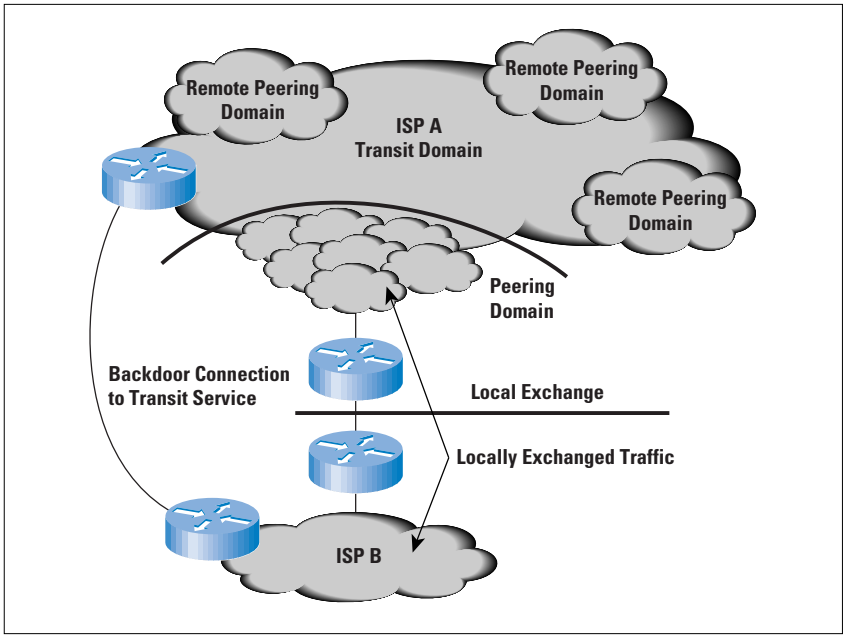


Figure 8: Peering and Transit Purchase

Exchange Business Models

For the ISP industry, many attributes are considered highly desirable for an exchange facility. The common model of an Internet exchange includes many, if not all, of the following elements:

- Operated by a neutral party who is not an ISP (to ensure fairness and neutrality in the operation of the exchange)
- Constructed in a robust and secure fashion
- Located in areas of high density of Internet market space
- Able to scale in size
- Operates in a fiscally sound and stable business fashion

A continuing concern exists about the performance of exchanges and the consequent issue of quality of services that traverse the exchange. Many of these concerns stem from an exchange business model that may not be adequately robust under pressures of growth from participating ISPs.

The exchange business models typically are based on a flat-fee structure. The most basic model uses a fee structure based on the number of rack units used by the ISP to colocate equipment at the exchange. When an exchange participant increases the amount of traffic presented over an access interface, under a flat-fee structure, this increased level of traffic is not accompanied by any increase in exchange fees. However, the greater traffic volumes do imply that the exchange itself is faced with a greater traffic load. This greater load places pressure on the exchange operator to deploy further equipment to augment the switching capacity, without any corresponding increase in revenue levels to the operator.

For an exchange operator to base tariffs on the access bandwidths is not altogether feasible, given that such access facilities are leased by the participating ISPs and the access bandwidth may not be known to the exchange operator. Nor is using a traffic-based funding model possible, because an exchange operator should refrain from monitoring individual ISP traffic across the exchange, given the unique position of the exchange operator. Accordingly, the exchange operator has to devise a fiscally prudent tariff structure at the outset that enables the exchange operator to accommodate large-scale traffic growth, while maintaining the highest possible traffic throughput levels.

Alternatively, there are business models in which the exchange is structured as a cooperative entity among numerous ISPs. In these models, the exchange is a nonprofit common asset of the cooperative body. Although widely used, these models are prone to the economic condition of the *Tragedy of the Commons*. It is in everyone's interest to maximize their exploitation of the exchange, while no single member wants to underwrite the financial responsibility for ensuring that the quality of the exchange itself is maintained.

The conclusion that can be drawn is that the exchange is an important component of Internet infrastructure, and the quality of the exchange is of paramount importance if it is to be of any relevance to ISPs. Using an independent exchange operator whose income is derived from the utility of the exchange is one way of ensuring that the exchange is managed proficiently and that the service quality is maintained for the ISP clients of the exchange.

A Structure for Connectivity

Enhancing the Internet infrastructure is quantified by the following objectives:

- Extension of reachability
- Enhancement of policy matching by ISPs
- Localization of connectivity
- Backup arrangements for reliability of operation
- Increasing capacity of connectivity
- Enhanced operational stability
- Creation of a rational structure of the connection environment to allow scalable structuring of the address and routing space in order to accommodate orderly growth

We have reached a critical point within the evolution of the Internet. The natural reaction of the various network service entities in response to the increasing number of ISPs will be to increase the complexity of the interconnection structure to preserve various direct connectivity requirements. Today, we are in the uncomfortable position of increasingly complex interprovider connectivity environments, a situation that is stressing the capability of available technologies and equipment. The inability to reach stable cost-distribution models in a transit arrangement creates an environment in which each ISP attempts to optimize its position by undertaking as many direct 1:1 connections with peer ISPs as it possibly can. Some of these connections are managed via the exchange structure. Many more are implemented as direct links between the two entities. Given the relative crudity of the inter-*Autonomous System* (AS) routing policy tools that we use today, this structure must be a source of considerable concern. The result of a combination of an increasingly complex mesh of inter-AS connections, together with very poor tools to manage the resultant routing space, is an increase in the overall instability of the Internet environment. In terms of meeting critical immediate objectives, however, such dire general predictions do not act as an effective deterrent to these actions.

The result is a situation in which the inter-AS space is the critical component of the Internet. This space can be viewed correctly as the *demilitarized zone* within the politics of today's ISP-based Internet. In the absence of any coherent policy, or even a commonly accepted set of practices, the lack of administration of this space is a source of paramount concern.

GEOFF HUSTON holds a B.Sc and a M.Sc from the Australian National University. He has been closely involved with the development of the Internet for the past decade. He was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Technologist in the Internet area for Telstra. He is also an active member of the IETF, and is a member of the Internet Society Board of Trustees. He is author of *The ISP Survival Guide*, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, a collaboration with Paul Ferguson. Both books are published by John Wiley & Sons.
E-mail: gih@telstra.net

IPv6—What and Where It Is

by Robert L. Fink, Energy Sciences Network

The current Internet Protocol, known as IPv4 (for version 4), has served the Internet well for over 20 years, but is reaching the limits of its design. It is difficult to configure, it is running out of addressing space, and it provides no features for site renumbering to allow for an easy change of *Internet Service Provider* (ISP), among other limitations. Various mechanisms have been developed to alleviate these problems (for example, *Dynamic Host Configuration Protocol* [DHCP] and *Network Address Translation* [NAT]), but each has its own set of limitations.

The *Internet Engineering Task Force* (IETF) took on this problem in the early 1990s by starting an IPng (*Internet Protocol next generation*) project. After an over two-year-long process of defining goals and features, getting the best possible advice from industry and user experts, and sponsoring a protocol design competition, a new Internet Protocol was selected. Many proposed protocols were reviewed, analyzed, and evaluated. An evolved combination of several of them (*Simple Internet Protocol* [SIP], the *"P" Internet Protocol* [PIP], and *Simple Internet Protocol Plus* [SIPP]), each using fixed-length addressing, resulted in a final variation, called IPv6, which was selected over a version of the ISO OSI *Connectionless Network Protocol* (CLNP) (known as the *TCP and UDP with Bigger Addresses* (TUBA) IPng proposal).

Much work has been done since the selection of IPv6 in 1994. Over 50 implementations of IPv6 are believed to be under way or completed. A constantly growing international IPv6 testbed, called the *6bone,* now spans 260 sites in 39 countries, with over 25 different IPv6 implementations in use. Most router companies, including 3Com, Bay, Cisco Systems, Digital, Nokia, and Telebit support IPv6. IPv6 is also available for Digital, HP, IBM, Sun, WinTel, and many other end-user host systems.

IPv6 Addresses—Larger and Different

The larger 128-bit IPv6 address (versus the 32-bit IPv4 address) allows more flexibility in designing newer addressing architectures, as well as providing large enough address spaces for predicted future growth of the Internet and Internet-related technologies. A new addressing format, called the *Aggregatable Global Unicast Address Format*, has been developed to help solve route complexity scaling problems with the current IPv4 Internet. The current IPv4 provider-based addressing used in the Internet relies on separate IPv4 addresses being assigned to ISPs in contiguously numbered blocks for routing efficiency; that is, the routers need to carry fewer routes.

However, there is currently much fragmentation in the IPv4 address space. This situation, aggravated by sites not being able to easily renumber, causes many more separate routes than necessary, in turn leading to route computation complexity (too many routes, too many dynamic changes, too much computation in routers).

Public Routing Topology Prefixes

With the new aggregatable style addressing (see Figure 1), the left-most 48 bits of the address are defined as a *Public Routing Topology* (PRT) prefix. The first 3-bit field of this prefix specifies that the addressing format is aggregatable. The next 13-bit portion specifies the *Top Level Aggregator* (TLA) ID that constrains the top level of Internet routing to 8,192 major transit providers and a new concept of routing exchanges. Each TLA (top level transit ISP) is then responsible for all the remaining public routing topology assignment below it; that is, the *Next Level Aggregator* (NLA) ID. As shown in Figure 1, the NLA may have a tiered hierarchy to allow multiple levels (NLA1, NLA2, and so on) of other ISPs, each of which would then have control of the assignment of the space below it. The right-most portion of the NLA field, at whatever level it may be, would identify the end-user "leaf" site. An 8-bit reserved field has been defined to allow the growth of either the TLA or the NLA fields.

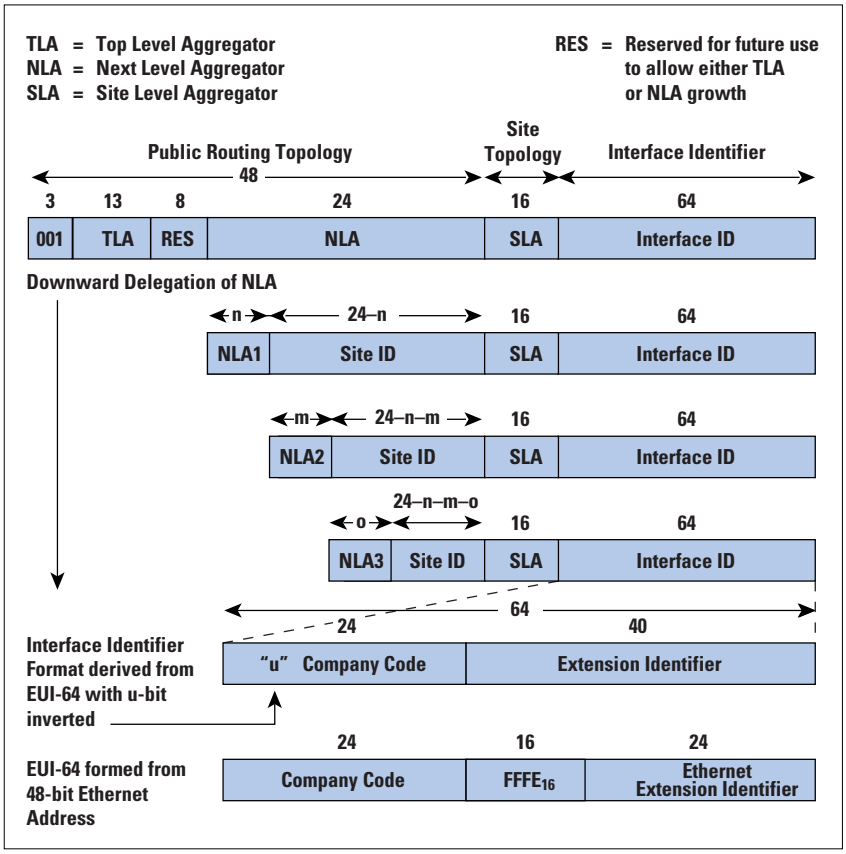


Figure 1: Aggregatable Global Unicast Address Format

The advantage of this style of addressing is that it allows automatic address clustering, or aggregation, into a constrained set of routes, which are represented through the TLA field. If the initial assignment of 13 bits (8,192 TLAs) is insufficient in the future, either the reserved field or another piece of the IPv6 128-bit address space could be utilized. Note that only one-eighth of the current IPv6 address space has been assigned to aggregatable addressing.

Even with this new concept of addressing, sites will still occasionally want to change their ISP (as in the current IPv4-based Internet) and thus will need to readdress to keep the addressing structure constrained. This is where *Site Renumbering*, which will be discussed later, comes in.

IPv6 TLA Assignment

To begin the production use of IPv6, ISPs providing IPv6 service need to be assigned TLAs so they may assign NLAs to transits and sites they are serving. Until recently, this was not possible. Recent discussions between the IETF, the IANA (*Internet Assigned Numbers Authority*), and the major address registries (APNIC, ARIN, and RIPE-NCC), have resulted in agreements that will provide a way to request and assign TLAs by early 2nd quarter 1999.

The process agreed upon is based on the above discussions that have been published as a recommendation in an Informational RFC on TLA assignments. The basic idea is to provide a slow start mechanism for TLAs by assigning one TLA ID to be used for defining a Sub-TLA field of 13 bits out of the reserved and NLA fields (see Figure 2). This will allow transits to demonstrate their need for a full TLA based on usage of the assigned Sub-TLA. These rules, based on much current practice with IPv4, are necessary to keep aggregatable addressing functional and effective for hierarchical routing as IPv6 comes into use.

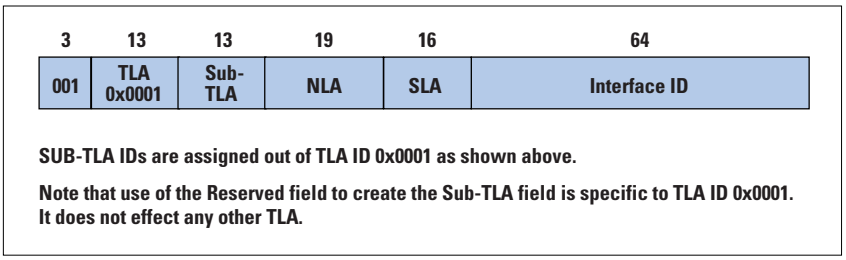


Figure 2: Sub-TLA Format for IPV6 Address Assignment

Rules for assigning these Sub-TLAs include:

- Must have a plan to offer native IPv6 service within three months from assignment; must have a verifiable track record providing Internet transit to other organizations
- Must make payment of a registration fee to the IANA and reasonable fees for services rendered by the address registry
- Must maintain registries of sites and next-level providers and make them available publicly and to the registries; must provide utilization statistics of NLA space below the assigned TLA (or Sub-TLA) and also show evidence of carrying TLA routing and transit traffic

These rules are intended to minimize route explosion and address assignment misuse to aid in the stability of the IPv6-based Internet.

Site Topology Prefixes

In addition to identifying the address of the site with the PRT prefix, aggregatable addressing provides for a site to have aggregation as well using a 16-bit *Site Level Aggregator* (SLA). The SLA might be as simple as a subnet number (more than 64,000 of them!), or a tiered hierarchy such as the NLA provides. However it is structured, the SLA is under the control of the site, and identifies the subnet that a host interface is attached to (IPv6's addressing, as IPv4's, specifies interfaces on systems, not the entire system).

It is very unlikely that an organization will ever need more than one PRT prefix, given the size and flexibility of the SLA and the *System Interface Identifier* field (described below).

System Interface Identifiers

Now that we have identified how to reach the site and the subnet a system is attached to, an interface identifier (ID) specifies the local logical address of the interface on the local subnet (or *link* as it is often called). The interface ID is formed and derived from the new IEEE EUI-64 media-level address that is an expansion of the well-known Ethernet 48-bit address format that allows for more device identifiers to be assigned by each manufacturer. The global/local bit is also inverted to make manually assigned (that is, local) addresses easy to form with only leading zeros.

If the IPv6 node is attached to an Ethernet "link," then the 48-bit address is turned into 64 bits by a filler field inserted in the middle (see Figure 1).

This enlarged Interface ID will allow newer technologies, such as *FireWire*, and newer applications, such as traffic lights and PCS/PDA telephones, to have unique interface identifiers assigned to them from a global address space.

The use of a media-level address for a network-level Interface ID allows the very important IPv6 Stateless Address Autoconfiguration Protocol to work.

Stateless Address Autoconfiguration

Automatic configuration of IPv6 end systems (hosts) is one of the most important features of IPv6. In the current IPv4 Internet, you must either manually configure IP address, network mask, and default gateway, or rely on having a DHCP server. With IPv6, this process can take place automatically, with no reliance on outside systems, using the IPv6 *Stateless Address Autoconfiguration Protocol*.

This can be done because the *Media Access Control* (MAC) address is used to form the host's interface ID. For example, if a host has an Ethernet interface that it is trying to configure for use with IPv6, the 48-bit Ethernet MAC address is formed into a 64-bit interface ID, which is the right-most 64 bits of the IPv6 address (see Figure 1). Then, using the *Neighbor Discovery* (ND) protocol, which is unique to IPv6, this formed interface ID is checked to see that it does not have a duplicate on this link (that is, subnet). If it does, a randomly generated token can be used (though a rare occurrence, it is a necessary protection against illegal Ethernet address usage and situations where the same address may be used on multiple interfaces for legitimate reasons).

At this point, an *ND Router Solicitation* multicast message is sent out to discover if there is a local IPv6 capable router, what the local site's topology ID for the host's subnet is, and what the site's public topology routing prefix is. Neighbor Discovery can also be used to control whether the site then wishes to continue with further configuration using Stateful Autoconfiguration with DHCPv6.

IPv6 Autoconfiguration thus provides for standalone operation of two or more hosts on a local LAN link with no router present, provides for operation within a site with no outside Internet connectivity present, and allows for easy changing of the site's public topology routing prefix, either when external connectivity comes on line, or when the external connectivity is changed, such as when a different ISP is chosen.

Domain Name System—Forward and Reverse

The *Domain Name System* (DNS) is an essential component of the Internet. To provide a mapping from a domain name to an IPv6 address, as well as an IPv4 address, a new DNS record type of "AAAA," or "quad A," is defined. This is a clever word play on the "A" record type that the original DNS specification defines for 32-bit IPv4 addresses, because IPv6 addresses are four times larger (128-bits), hence "AAAA"!

Most existing implementations of DNS already support AAAA records and existing IPv4 queries of DNS can access these records; that is, you don't need a DNS operating over IPv6 to retrieve these new AAAA records. This support also includes reverse lookups, similar to IPv4s, although a new reverse lookup proposal that will allow automatic partitioning of the delegation information on arbitrary bit boundaries is under consideration. This new capability should make for more reliable reverse registry than exists with IPv4, and easier maintenance when sites change their PRT prefix.

When a host with both IPv4 and IPv6 operating on it ("dual stack") queries the DNS for the address of a remote host, the A and AAAA records returned are used to indicate what protocol to use in communicating with that remote host. If no AAAA record is returned, IPv4 must be used. If only a AAAA record is returned, IPv6 must be used. If both A and AAAA are returned, either IPv4 or IPv6 may be used.

A new modification of the IPv6 DNS extensions is nearing completion that allows the automatic joining of the routing prefixes and Interface IDs when a host's IPv6 address is returned, thus making it easier to renumber a site. This new IPv6 DNS feature makes changing a site's PRT prefix (renumbering) very easy as only one entry, the PRT prefix, needs to be changed. This setup also facilitates easy support of multiple addresses for each host. These enhancements are very useful; IPv4 does not have this feature.

Renumbering Sites When ISPs Change

Because IPv6 addressing is based on the PRT prefix assigned by its ISP, it is essential that it be easy for a site to renumber itself when its choice of ISP changes. To aid in this, a new *Router Renumbering* (RR) protocol, in conjunction with Autoconfiguration, Neighbor Discovery and the new Aggregatable Unicast addressing PRT prefix are used.

RR allows a site's network administrator to set new PRT prefixes into the site's routers, as well as lower the lifetime of existing ISP PRT prefixes to specify an overlap interval, after which the old ISP's service is discontinued.

Hosts learn their new routing prefixes either when they restart, and thus are automatically configured with Autoconfiguration, or when they are informed by their local router that a new prefix is to be used during periodic router notification updates using ND.

For example, a new ISP service is readied for service while the old ISP is notified that it will provide service for just 60 more days. After the new PRT prefix is announced to the site's routers by RR, hosts will use the new prefix (that is, new ISP) for all new connections, while existing connections continue to work until the old prefix is withdrawn (that is, after 60 days in this example).

The easy renumbering of an IPv6 site will make easy a task that is currently very painful for an IPv4 site because hosts are often manually configured in many networks.

The 6bone—An IPv6 Testbed

The 6bone is an international IPv6 testbed network that is overseen and directed through the IETF *IPng Transition Working Group* (ngtrans) that provides:

- Testing of IPv6 implementations and standards
- Testing of IPv6 transition strategies
- A place to gain early applications and operations experience
- Motivation and a place for implementers, users, and ISPs to try IPv6
- An experimental first step toward transition

In the early phases of IPv6 deployment, most native IPv6 transport is restricted to site LANs with the ability to experiment with it locally. Some sites in Great Britain, The Netherlands, and Japan are using native IPv6 over WAN links.

ISPs and various other private IPv4 transit providers may not place IPv6 in their production routers in this early phase of IPv6 deployment, leaving early IPv6 testers with the need to use the existing IPv4 Internet infrastructure to deliver IPv6 packets among themselves when remotely located. Thus an IPv6 transition feature, IPv6 encapsulation (that is, *tunneling*) over IPv4, is used for parts of the 6bone where native IPv6 may not be available. In this way, the 6bone is also thoroughly testing out its own transition technology as well as providing IPv6 service.

The 6bone is a diverse community of users, ISPs, and developer organizations, many of whom provide transit on the public spirited basis of promoting and gaining early experience with IPv6. It is expected that production variations of the 6bone will also be created to more formally carry production IPv6 traffic.

Components of the 6bone

The 6bone provides this needed IPv6 transport over the public Internet infrastructure, relying on:

- Dual IPv4/IPv6 stacks in the client host
- IPv6 packets encapsulated (tunneled) in IPv4 packets
- Dual IPv4/IPv6 stack backbone routers that know IPv6 routes of 6bone participants
- DNS that supports IPv6 AAAA records
- A 6bone Routing Registry to keep track of sites and their tunnels
- A mailing list, various IPv6 tools, and a 6bone Web site at: **`www.6bone.net`**

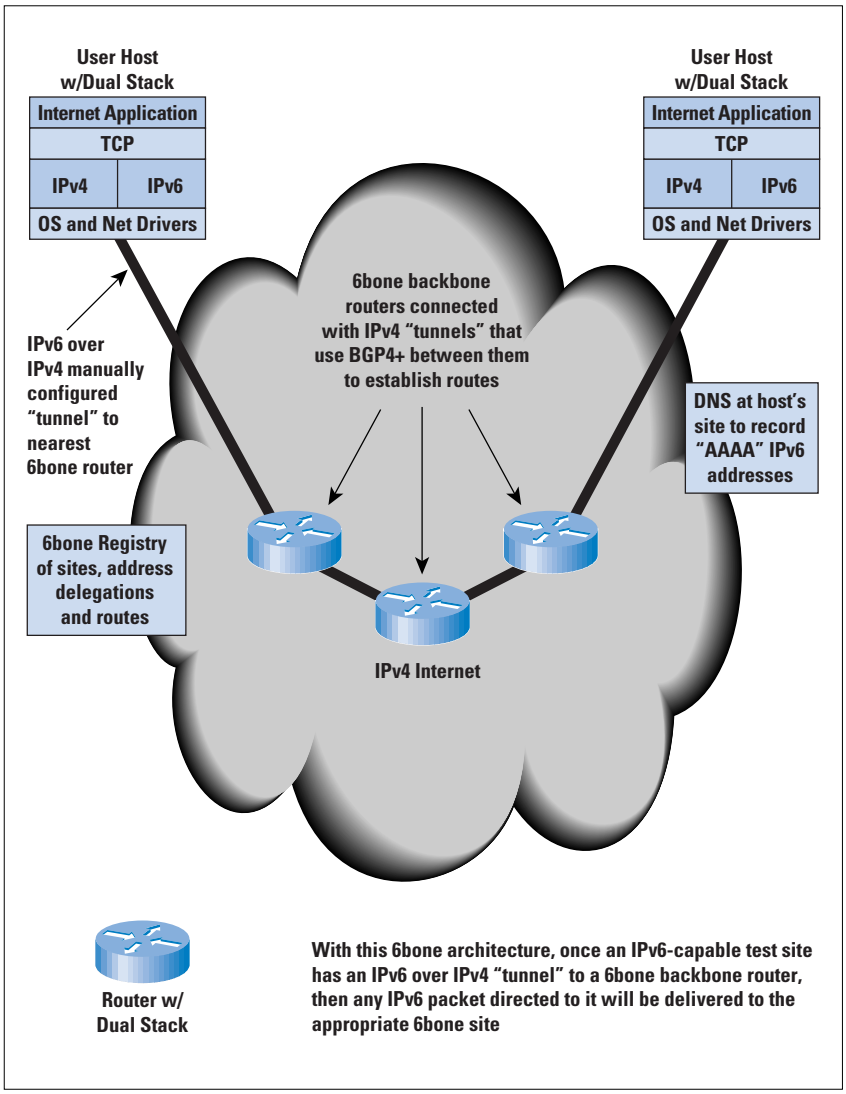
Figure 3 shows a conceptual overview of how a basic 6bone is structured and a picture of the current 6bone backbone structure can be seen at:

`http://www.cs-ipv6.lancs.ac.uk/ftp-archive/6Bone/Maps/full-backbone.gif`

…with the pseudo TLA site-to-site peering indicated by various colored links.

To date, the 6bone has spread to 260 organizations in 39 countries (see Table 1 on page 25).

Figure 3: 6bone Conc eptual Architecture



6bone History

Serious work to evolve and refine the IPv6 protocols sufficient to allow the start of various implementations of IPv6 began in 1994. By early 1996, it was obvious that a testing environment was needed, so in March 1996, several implementers and users met and agreed to start an international testbed called the 6bone.

By June 1996, two groups raced to provide the first IPv6 connectivity: the University of Lisbon (Portugal), the Naval Research Laboratory (U.S.), and Cisco Systems (U.S.); a Danish universities consortium (UNI-C), a French universities consortium (G6), and a Japanese universities consortium (WIDE).

Table 1: Countries with Sites Participating in the 6bone

AT-Austria	FI-Finland	NL-The Netherlands
AU-Australia	FR-France	NO-Norway
BE-Belgium	GB-United Kingdom	PL-Poland
BG-Bulgaria	GR-Greece	PT-Portugal
BR-Brazil	HK-Hong Kong	RO-Romania
CA-Canada	HU-Hungary	RU-Russian Federation
CH-Switzerland	IE-Ireland	SE-Sweden
CM-Cameroon	IT-Italy	SG-Singapore
CN-China	JP-Japan	SI-Slovenia
CZ-Czech Republic	KR-Korea	SK-Slovakia
DE-Germany	KZ-Kazakhstan	TW-Taiwan
DK-Denmark	LT-Lithuania	US-United States
ES-Spain	MX-Mexico	ZA-Zaire

6bone Backbone and Addressing

By the end of 1997, the 6bone converted to the new aggregatable addressing format, a change necessitated by having originally adopted an early prototype provider-based addressing format discussed during early IPv6 design efforts.

Along with the change to a new addressing format was the need to clean up the routing used among the 6bone backbone transit sites. It was originally thought that IDRPv6 (a new Internet Domain Routing Protocol based on earlier IPv4 work) would be the prevailing *Exterior Gateway Protocol* (EGP) used for IPv6 Internet peering.

By mid 1996, various ISPs made it known that a new EGP for IPv6 was not a practical alternative, given the explosive growth of the Internet and the current evolution and widespread use of the *Border Gateway Protocol 4* (BGP4) by ISPs. There was a need to allow for multiprotocol extensions to BGP4, allowing ISPs to more easily adapt their operations to IPv6. This situation led to the rapid evolution of BGP4+, an extension of BGP4 to include IPv6 and IPv4 multiprotocol routing.

By mid 1997, the decision was made to convert the 6bone backbone to BGP4+ for its EGP. See **`http://www.cs-ipv6.lancs.ac.uk/ftp-archive/6Bone/Maps/full-backbone.gif`** for a recent picture of the 6bone backbone sites using the new aggregatable addressing format and the current status of the conversion to BGP4+.

6bone Future Plans

To date, most 6bone efforts have been to prove out basic IPv6 interoperability among the many implementations, and to create a reliable international testbed infrastructure. This has included making its backbone operationally ready with the new aggregatable addressing format and use of BGP4+ for high-reliability routing and transit.

Now that the 6bone has completed these conversions, serious work can begin on testing site renumbering, security, applications, and transition mechanisms.

Other IPv6 Trials and Testing

Other testing venues have also been very important to the evolution of IPv6: the University of New Hampshire *Inter Operability Laboratory* (IOL), various trade show demonstration networks, for example, NetWorld+Interop, and various vendor-sponsored interoperability testing.

By early 1998, the UNH IOL had hosted five IPv6 test sessions, though specific details about participating vendors are not released.

In a positive sign of industry response to evolving IPv6 specifications, the late July 1997 UNH testing resulted in the successful interoperability of all participants using the new aggregatable addressing format, no more than two months from its first Internet Draft.

Implementations

To date, over 50 different IPv6 host and router implementations are either completed or under way. More than 30 implementations have been tested and used on the 6bone.

Router implementations to date include: 3Com, Bay, Cisco Systems, Digital, Fujitsu LR550, Hitachi NR60, Inria BSD, Linux, Merit MRT, Nokia, NRL for BSD, Telebit, WIDE KAME and ZETA for BSD, and WIDE v6d.

Host implementations to date include: Apple MacOS OpenTransport demo version, Digital OpenVMS, Digital UNIX, FTP Software Windows95, Fujitsu LR450, 460, and 550, Hitachi NR60, IBM AIX, Inria BSD, Linux, HP-UX (SICS), Microsoft Research WindowsNT versions 4 and 5, Sony CSL Apertos IPv4/v6 stack, Sun Solaris, Trumpet Winsock for IPv6, UNH for BSD, NRL for BSD, WIDE KAME and ZETA for BSD, and WIDE v6d.

Several new Windows implementations that will operate under Windows95/98/NT are under way.

Transition from IPv4 to IPv6—A Seamless Approach

IPv6 is unlikely to become the Internet network-layer protocol of choice unless there is literally no choice to be made by the end user, little effort by network and system administrators, and it can operate alongside IPv4 for the indefinite future. Therefore, it must be very easy for the private network (your corporate net) and public network (your ISP) operators to equip, enable, and operate IPv6, while operating IPv4, in such a way that the user doesn't notice that IPv6 is there at all.

A system administrator, but not the user, must be conscious of IPv6 in a minimal sense. It is just another protocol stack that any Internet-based applications will operate over if the system is configured and distributed to do so by the system administrator.

At the network operator level, IPv6 is just another routing stack that can easily be turned on in the site's and ISP's routers (many sites certainly support IPX, AppleTalk, DECnet,...). IPv6 interdomain routing can be operated just like IPv4s because it uses BGP4+.

With the aid of the new *Dynamic DNS Registration Protocol* and IPv6's Stateless Autoconfiguration, users can boot up their system after it has been enabled with an IPv6 stack, in addition to its IPv4 stack, and become IPv6-ready without being aware of it at all. The system would automatically be configured with an IPv6 address, have itself registered automatically in the DNS with the host's existing name alongside its new IPv6 address (in addition to its DNS IPv4 address registration), and when finding a remote host with IPv6, start talking IPv6—all this without the user being required to consciously take action.

Early Production IPv6 Networks

In October of 1998, the *6REN* initiative, was established by the U.S. Energy Sciences Network (ESnet). The 6REN is a voluntary coordination initiative of *Research and Education Networks* (RENs) that provide production IPv6 transit service to facilitiate high quality, high performance, and operationally robust IPv6 networks.

The first participants were ESnet (the U.S. Dept. of Energy's Energy Sciences Network), Internet2 (the advanced Internetworking development collaboration comprised of many large U.S. research universities), CANARIE (the Canadian joint government and industry initiative for advanced networking), vBNS (the MCI network for NSF advanced networking) and WIDE (the Japanese research effort to establish a "Widely Integrated Distributed Environment").

Other profit and not-for-profit networks worldwide have been invited to join the 6REN. It is expected that during 1999 a sizable production environment capable of advanced demonstrations and deployment of Internet applications over IPv6 networks will be in place.

The Future for IPv6

It is too early to predict with total certainty that the Internet will adapt to the use of the IPv6 protocol. However, it should be obvious that IPv6 offers many important features for a next-generation Internet: automatic configuration, greatly expanded addressing, easy site renumbering, built-in security, and more.

One possible scenario for IPv6 is where it becomes the protocol of choice for newer applications not currently using Internet technology; for example, controlling traffic lights, reading electric meters, and so on. In these uses, IPv6 does not require coexistence with IPv4 because some form of gateway function would provide interconnection to the current Internet.

Another scenario (which doesn't exclude the previous one) is that Microsoft provides IPv6 support for a future version of Windows Networking on Windows OS, and promotes it within corporate America for its better features in supporting advanced corporate application/networking needs. In this scenario, the Internet will learn to carry IPv6 somehow, even if it is via automatically created tunnels that operate over IPv4 (somewhat similar to the 6bone's tunneling, but with dynamic creation of the tunnels as needed). It is expected that after Microsoft ships IPv6 and large corporations begin using it, ISPs will deploy IPv6 to get their business.

Yet another possibility is that the Internet telephony revolution will come to the conclusion that only IPv6 can provide cost-effective, scalable, end-to-end worldwide telephony implementations. This may be even more important as new classes of wireless networked devices, for example, PDAs and PCS phones, are integrated and built in very large volume.

Also, in parts of Asia and China, where there is little Internet connectivity at present, and very few IPv4 addresses assigned, IPv6 may become very popular because it will allow rapid growth without concerns about address space.

The probability is high that not just one of the above scenarios will happen, but that all will occur, in addition to others not yet imagined.

Whatever the implementation scenario, the probability that IPv6 will augment IPv4 as a part of the Internet of the future is very high!

References

[1] IPng and IPv6 information, including formal specifications can be found at: **http://playground.sun.com/pub/ipng/html/**

[2] 6BONE information, including diagrams, hookup info, and registry access is at: **http://www.6bone.net**

[3] An IEEE EUI-64 overview can be found at: **http://standards.ieee.org/db/oui/tutorials/EUI64.html**

[4] "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1998.

[5] "Neighbor Discovery for IP Version 6 (IPv6)," RFC 2461, December 1998.

[6] "IPv6 Stateless Address Autoconfiguration," RFC 2462, December 1998.

[7] "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)," RFC 2463, December 1998.

[8] "IP Version 6 Addressing Architecture," RFC 2373, July 1998.

[9] "An IPv6 Aggregatable Global Unicast Address Format," RFC 2374, July 1998.

[10] "DNS Extensions to support IP version 6," RFC 1886, December 1995.

[11] "Proposed TLA and NLA Assignment Rules," RFC 2374, December 1998.

[12] "Transition Mechanisms for IPv6 Hosts and Routers," RFC 1933, April 1996.

[13] "Router Renumbering for IPv6," Internet Draft, Work in Progress, **draft-ietf-ipngwg-router-renum-06.txt**, November 1998.

[14] *IPv6: The New Internet Protocol,* Christian Huitema, ISBN 0-13-850505-5, Prentice Hall, 1998.

[15] *IPng: Internet Protocol Next Generation,* Edited by Scott O. Bradner and Allison Mankin, ISBN 0-201-63395-7, Addison-Wesley, 1996.

ROBERT FINK is a network researcher with ESnet (the U.S. Dept. of Energy's Energy Sciences Network) at the Berkeley Lab (the Ernest Orlando Lawrence Berkeley National Laboratory). He is cochair of the IETF ngtrans (IPng Transition) Working Group, and leads the 6bone project. You can reach him at: **fink@es.net**

Secure E-Mail: Problems, Standards, and Prospects

by Marshall T. Rose and David Strom

As we spend more and more time using e-mail, most of us eventually find that we need to be able to prove our identity to our correspondents and secure the contents of our messages so that others can't view them readily. Proving your identity is called *authentication*. In the physical world, this is accomplished by photo identification, such as a driver's license, passport, or corporate identity card. When the time comes to prove who you are (for example, before a major purchase), you show your card. Your appearance and signature match the photo and signature on your card, and the purchase is made.

On the Internet, however, the process isn't as easy. Does e-mail from **`sidney@example.com`** really originate from our friend Sidney at the Example Corporation? Maybe it's from someone else, who just happens to be using Sidney's machine when he is out to lunch. Or, worse, someone trying to impersonate Sidney illicitly. And even if the message actually is from the "real" Sidney, how can we be sure: Is there an electronic analog to a signature?

Most of us are trusting individuals; we tend to believe that people are who they say they are unless we have particular reasons to doubt their identity. But on the Internet, we have to look beyond face value. And proving that someone indeed did send a particular message is a very difficult problem.

This may be one of the main reasons why corporations employ Lotus Notes and other Internet-based messaging systems that are not 100-percent pure. They want to ensure that all messages carry the appropriate authentication with them at all times. In order for new users of Notes to start using the software, they must first obtain an electronic certificate that authenticates them to the system. The certificate is created by the Notes system administrator, who works in conjunction with that particular Notes server owned by that particular corporation.

Securing the message contents is also a challenge: all e-mail sent over the Internet, unless otherwise protected, is sent in clear ASCII text. If you have the tools, the time, and the technical expertise, you can capture this traffic and read anyone's correspondence. It isn't simple, but it is quite possible.

Besides being sent as clear text, e-mail can also be intercepted and its contents changed between the time the sender composes the message and the recipient reads it. Again, this task is neither likely nor simple, but it can be accomplished if someone is determined enough to do it. Therefore, senders can neither prove nor deny that they sent a particular message to you; it could be real or a forgery, and you have no way of knowing which.

Cryptography Standards

It would be great if we could say that the future for secure e-mail is bright, and that there will be standards in place that will help. However, the state of secure e-mail standards for the Internet is best described as a "terrible mess"! (Ed.: a less charitable phrase is used in the book from which this material is adopted.) Think that characterization is unprofessional? It is actually quite detached, considering the amount of culpability enjoyed by the principals of the Internet's secure e-mail debacle. We would love to write an article describing the high crimes and misdemeanors of these scoundrels, but that would only publicize the guilty, not punish them. So, instead we'll survey the horizon and try to make sense of what little terrain there is.[1]

In brief, no technologies for secure e-mail in the Internet meet all of the following criteria:

- Multivendor
- Interoperable
- Approved or endorsed by the Internet's standardization body

There are two competing technologies, each of which satisfies at most one of these criteria. However, for any 100-percent-pure Internet solution to succeed, we feel it must be based on technologies that satisfy all three.

Basic Concepts

In order to understand secure e-mail, you need to know only three concepts:

- Data encryption (privacy)
- Message integrity (authentication)
- Key management

Everything else is a matter of data formats.

Data Encryption

When the contents of a message are to be protected from third-party disclosure, it is necessary to agree upon an encryption algorithm. Because cryptographic algorithms are constantly being scrutinized, a secure e-mail standard must be extensible with respect to the algorithms that it allows.

Historically, *symmetric encryption algorithms* are used for this purpose. A symmetric algorithm is one in which the same key is used to both encrypt and decrypt the data. Symmetric algorithms are chosen because they are computationally less burdensome (in other words, faster to execute) than asymmetric algorithms.

As such, each time a message is to be encrypted, a new session key is generated for that purpose. Although one could send the session key via some secure path, it is easier to include the session key along with the message, but encrypted so that only the intended recipient can decipher it. Upon deciphering the session key, the recipient can apply the encryption algorithm and retrieve the original contents.

For example, Network Associates' Pretty Good Privacy (PGP), one of the two technologies we'll examine, uses an asymmetric algorithm to encrypt the session key and a symmetric algorithm to encrypt the user's data.

Message Integrity

When the contents of a message are to be verified as authored by a particular user and unaltered by any other user, it is necessary to agree upon a *signature* and *hash algorithm*. The former is used to verify the authenticity of the message, and the latter is used to verify the integrity of the message. Again, any secure e-mail standard must be extensible with respect to the algorithms that it uses for these purposes.

For signature algorithms, asymmetric algorithms are typically used. These algorithms utilize a public key and a secret key. A signature algorithm combined with a secret key allows someone to generate a digital signature for the contents of a message. A signature algorithm combined with a public key allows someone to verify the digital signature for a message. As you might expect, signature algorithms are one-way functions: You can't reconstruct the input to a signature function by looking at its output.

Hash algorithms are often called *message digest algorithms*. They simply compute a checksum on their input; no keys are involved. Hash algorithms are also one-way functions, and a good hash algorithm is one in which very similar inputs produce dramatically different outputs. Hence, if even a single bit is altered or corrupted in transit, the hash value will be different.

Key Management

All discussion now hinges on how keys are used for asymmetric algorithms. Specifically, how do you trust the identity of the secret key used to make a digital signature? To start, we have to introduce the notion of a *public key certificate*. Although the actual formats vary, at its heart a certificate contains three things:

- The identity of the "owner" of the certificate
- A public key
- Zero or more guarantees to the validity of the binding between the identity contained in the key and the owner in the "real world"

So, the next step is to ask what these identities and guarantees look like. Unfortunately, we now enter the realm of sociology rather than technology. The only theoretical limitation on an identity is that you have to be able to represent it digitally. It could be a name (for example, "Jim Bidzos") or an e-mail address (for example, **prz@pgp.com**) or a key in some database (for example, the name of an object in a directory). More interesting examples could include a series of assertions (for example, your driver's license number is this, your passport number is that, and so on).

Fortunately, the guarantees are a bit simpler to describe—they are digital signatures from other public keys that vouch for the veracity of the binding. For example, if you encountered a public key certificate in which the identity was someone's passport number, it would be natural to expect that the certificate contains a digital signature from the government entity (or its agent) that issued the passport. However, this begs another question: Why should you trust the entities that have signed someone's public key? It turns out that our two contending technologies have different answers to that question.

As you might expect, certificates have some additional properties, such as a date the certificate becomes valid, the date the certificate expires, and a "fingerprint." The fingerprint is simply a hash of the identity and public key so you can tell if it has been altered in transit.

Finally, *certificate revocation lists* identify certificates that are no longer valid. For example, if the secret key associated with a certificate is accidentally disclosed, then the corresponding certificate is revoked.

Pretty Good Privacy: The Web of Trust

Pretty Good Privacy (PGP) is encryption for the masses. Despite the fact that it required a couple of complete rewrites in order to achieve stability, it gets the job done.

An effort is under way to provide a "standards-based" version of the PGP technology, termed *OpenPGP*. The "pre-standards" version of PGP uses the RSA algorithm for signatures and the IDEA algorithm for encryption. The version being developed is more flexible with respect to the algorithms it supports.

The most remarkable thing about PGP is its trust model. Remember the earlier question: How do you know whether you should believe the identity in a public key certificate? To answer this in the context of PGP, each user assigns two attributes to the PGP certificates that they encounter: *trust* and *validity*. Trust is a measure as to how accurate the certificate's owner is with respect to signing other certificates. Validity indicates whether or not you think the identity in the certificate refers to the certificate's owner.

So, initially your local collection of certificates starts out with one—your own PGP certificate. You then sign your friend's certificate and he or she signs yours. Because you trust yourself when signing those certificates, your friend's certificates are automatically considered valid. Then, based on your judgment of your friend's abilities to sign other certificates accurately, you assign a level of trust to his or her PGP certificates. As you receive messages containing other people's certificates, if they are signed by you, or any of your trustworthy friends, they are automatically deemed valid. This organic, highly decentralized approach toward validating public key certificates is termed the *web of trust*.

Key servers are also available that are repositories of PGP certificates. If you need to send e-mail to someone, but don't have his or her certificate, you can query a server to see if a copy is there. Of course, the usual rules apply with respect to assigning trust and validity—it's up to you! Key servers also help when you receive e-mail from someone new. Although the message will contain a copy of someone's PGP certificate, you may not know about any of the signatories. So, you can go to a key server and fetch the certificates for the signatories; you might decide to trust them after seeing who signed their certificates.

We've simplified the web of trust in that validity isn't "all or nothing," as we implied previously. Rather, PGP offers a flexibility spectrum of possibilities; for example, requiring two trustworthy signatories before considering a certificate to be valid. But the one thing that should be clear is that trust and validity are *different*. You will probably have many keys in your local collection of certificates that are considered valid, but probably only a few of those will be considered authorized to vouch for others.

Secure MIME: The Hierarchy of Trust

There is an interesting concept in advertising called "ambush marketing." The basic idea is that your advertising campaign leverages off the brand and promotion of a competitor. *Secure Multipurpose Internet Mail Extensions,* or S/MIME, is an example of ambush marketing in the Internet. Although MIME is an Internet standard, which has been implemented by hundreds of vendors and provisioned in tens of thousands of networks, S/MIME is the product of a closed vendor consortium.

S/MIME has two versions: version 2 and version 3. As of this writing, products that claim to implement S/MIME implement version 2. They use the RSA algorithm for signatures and a weak algorithm for encryption (RC2 with 40-bit keys). An effort is under way to provide a "standards-based" version of the S/MIME technology—version 3. The version being developed is more flexible with respect to the algorithms it supports. S/MIME uses a hierarchical model for establishing trust. For example, if your employer assigns you an S/MIME certificate, he will act as a certification authority and sign that certificate. As a consequence, trust is established on the basis of a hierarchical relationship between the *subject* of a certificate (the identity) and the *issuer* (the signatory).

This model has some strengths: users rely on the certification authorities implicitly. However, a bootstrapping problem still exists: How do you know to trust the issuer? The answer is that your local collection of certificates also has some "top-level" certificate authorities, and it is these authorities that sign the public key certificates of the issuers. If the hierarchy of trust can be kept to one or two levels, this is manageable in practice.

The web and hierarchical models of trust share many attributes in common. For example, when you receive a message, it contains a copy of the certificate that was used to make the digital signature. If you aren't familiar with the signatories, you can look in a remote repository of keys. The only difference between the two models here is that the hierarchical model needs key servers to make its key infrastructure work. Because of this, keys are usually stored in a directory service accessed via the *Lightweight Directory Access Protocol* (LDAP).

Data Formats

The **`multipart/encrypted`** and **`multipart/signed`** contents are used to convey secure e-mail. Fortunately, they are both very simple content types.

A **`multipart/signed`** content has two subordinate body parts. The first contains the data that is being authenticated and can be any MIME content type (**`text/HTML`**, **`multipart/mixed`**, and so on). The second contains the digital signature used to authenticate the content. The **`multipart/signed`** content has two mandatory parameters. The *protocol* parameter defines the technology used to generate the digital signature, and the *micalg* (for "MIC algorithm") parameter defines the hashing algorithm used (for "MIC" read: *message integrity check*). The value of the protocol parameter is also the content type used for the second body part. The only tricky part is that the digital signature is calculated on the data before a transfer encoding, if any, is applied.

Let's make this a little more concrete. If we assume that the OpenPGP effort produces an Internet standard based on the current draft (a reasonable assumption at 50,000 feet), then the structure of a **`multipart/signed`** message created using PGP technology would look like the following:

- The protocol parameter would be **`application/pgp-signature`**
- The micalg parameter would be **`pgp-md5`**
- The first body part would be labeled as whatever you wanted to sign
- The second body part would labeled as **`application/pgp-signature`**

The second body part, a data structure defined by the OpenPGP document, contains the digital signature along with any supporting material (for example, a copy of the sender's PGP certificate).

Note that you don't encrypt the first body part in a **multipart/signed** content. In this way, if only some of your recipients have secure e-mail, but you still want to sign it for those who do, everyone can still read the first body part.

A **multipart/encrypted** content has two subordinate body parts. The first contains the information needed to decipher the encrypted data (for example, the encrypted session key along with an indication as to the certificate needed to decipher the session key). The second contains the encrypted data, labeled as **application/octet-stream**. The **multipart/encrypted** content has one mandatory parameter, protocol, which defines the technology used to encrypt the data. The value of the protocol parameter is also the content type used for the first body part.

To further define this concept, if we use OpenPGP as the basis for a hypothetical example, then the structure of a **multipart/encrypted** would look like the following:

- The protocol parameter would be **application/pgp-encrypted**
- The first body part would be labeled as **application/pgp-encrypted**
- The second body part would labeled as **application/octet-stream**. In practice, the input to the encryption algorithm would be **multipart/signed** .

Finally, one or more MIME content types might be defined for sending certificates, certificate revocation lists, and so on. These are all specific to the particular secure e-mail technology being used.

Encrypting Your Messages

If we look at popular commercial e-mail products, many of them include support for some kind of encryption. Both Microsoft's Outlook Express and Netscape Messenger include support for S/MIME, although we'll see in a moment that the two have radically different capabilities. And Qualcomm's Eudora Pro package comes with an add-on module for supporting PGP, which you may or may not have installed when you installed the software. In order to encrypt a message, you need to go through the following process:

1. Choose which of the two competing technologies (and specific e-mail software) you wish to use for your encrypted correspondence. Both methods have advantages and disadvantages.
2. Choose whether you want to just digitally sign your messages or encrypt their entire contents, or both.
3. Either choose an enterprise certificate authority and set up the appropriate server software, or obtain a certificate from a public authority. Again, both methods have advantages and disadvantages.
4. Enroll with this certificate authority and obtain an encryption certificate or key for a particular machine and a single e-mail address.

5. Exchange keys with your correspondents, and manage where these keys are stored on your machine.
6. Encrypt and decrypt messages.

If this process seems rather involved and complex, it is. The process is not nearly where it should be to enable encryption to be useful by most e-mail users, and won't be for some time. If all of this seems overwhelming to you, we certainly understand.[2] It is to us, too! But let's go through these six steps in more detail.

PGP vs. S/MIME

Our discussion in the standards section might have convinced you that encryption technology is still very much a work in progress, and after you begin to use the encryption features of your own e-mail software, you'll be further convinced. Nevertheless, unless you plan to test lots of different software products, you should first decide on which product and which encryption technology you intend to use. You definitely want to limit yourself to as small a universe as possible, because running more than one e-mail software product will only make your encryption life miserable. So which to choose?

PGP is everyman's product. It was designed for single individuals to use and still remains the easiest method to set up and get going, although it is far from simple. The version of PGP that comes with the Eudora Pro box is the individual version; a separate and more capable version is available for workgroups or businesses, called *PGP for Business Security*. This business version is the one we recommend, even if you are the only person in your corporation that will use encryption. You'll find that after you start, others will follow, and you might as well start off with the more capable version.

If you want to use PGP, you will need to run a separate piece of software to encrypt and decrypt your messages. If you already use software such as Messenger or Outlook Express, that is certainly more cumbersome than using the built-in S/MIME features of those two products.

In 1999, PGP is more capable than S/MIME when it comes to setting up an enterprise encryption policy and putting it into practice on a daily basis. For example, with PGP you could establish that all outgoing and incoming encrypted messages are first copied to a special archive, and that all outgoing messages are encrypted with a special administrator's key that can be used in an emergency to read the message if the sender forgets his key or leaves the company. S/MIME doesn't have this ability yet, although this feature is being developed for the future.

PGP is a single-vendor solution: All your software must eventually come from Network Associates to run the various certificate servers and encryption modules. With S/MIME, you'll have some degree of choice, although we found that in practice you probably want to make use of

the same e-mail product when exchanging encrypted messages if you want them to be read with a minimum of difficulty. Not all S/MIME packages can exchange encrypted messages with each other because of differences in their implementations. When Dan Backman of *Network Computing* magazine tested five different products, he found several that couldn't read messages sent by others.[3]

Part of the problem with S/MIME is the various choices of "strength" of cryptographic algorithms that are in use in today's browsers and e-mail software. This debate is more about politics than technology, because the U.S. government places restrictions on various algorithms, as mentioned earlier. Two different parameters are of interest: the length of the key itself used in any certificate and the type of encryption technology used. Netscape software supports key lengths ranging from 512 to 1024 bits, for example. In addition, several choices are available for encryption technology; they are labeled *RC2* (which can either be 40-bit encryption, the only one allowed for export by the U.S. government, or more complex encryption of 64, 128, or even 255 bits), and *Data Encryption Standard* (DES). RSA, Inc., developed RC2. On the other hand, the U.S. government developed DES. Debate abounds as to which is the better or more or less proprietary technology.

These details are outside the scope of this article, but you should know that the larger the key size and encryption algorithm, the more difficult it is for someone to decode an intercepted message.

Digital Signature Required?

Your next choice is to consider whether to just make use of a digital signature, to encrypt the entire message, or to make use of both technologies. All encryption products can do both, but in somewhat different ways.

Digital signatures guarantee that your recipients have received your message without any tampering and that they can trust that the message came from you. The actual message body, and any attachments, arrive without any encryption, meaning that someone could still capture this traffic and read your correspondence. You might want to use a digital signature without encrypting the message, if you care that your message was received intact and that your correspondents can know that you sent it.

There are two different types of signed messages: *clear* and *opaque*. With clear-signed messages, you can still read the message text, even if you don't have any encryption functions in your e-mail software. The signature is carried along with the message in a separate MIME portion of the message from the message body, which remains untouched and still readable. This feature can be handy, especially if you correspond with many people and they probably haven't adopted any particular encryption product, or if they are using older versions of e-mail software that don't support encryption. Clear signing is also useful in circum-

stances where your encryption technology isn't compatible with your correspondents' technology. PGP supports only clear signing in its products.

One problem with clear signing is e-mail gateways. They often will break the encryption of the signature, because they will either add or remove characters from the message, and that sloppiness could invalidate the signature block. After all, part of the role of the signature is to ensure that the message was delivered intact and unaltered!

Opaque signing means that your recipients will get a blank message if they aren't running any encryption software, or if their encryption software doesn't work with yours. Opaque signing wraps the entire message in a Base64 encoding, which is usually left alone by most e-mail gateways. This encoded message then gets transmitted and then decoded by the S/MIME recipient.

PGP places its signature inside the encrypted envelope when it sends messages, making it difficult to determine the signature of such a message until you first decrypt it. The PGP producers claim that this feature offers extra protection in case the message is compromised or copied en route. Newer versions of PGP offer a MIME option that places the signature outside the encrypted envelope. This is how S/MIME products work, making it easier to determine who sent it.

Choose Your Certificate Authority

Now you have another decision to face, and that is how to set up what is called the *certificate authority* (CA) for your enterprise. This software runs on a UNIX or NT server and manages the keys or certificates of everyone in your corporation. It serves as a central place of trust and signs all of your users' certificates. If you trust your CA, in theory you should be able to trust the certificates that are signed by the CA, called *inherited trust.*

The problem is that there isn't any "central" CA for the entire universe of e-mail users. Although there are several public CAs that anyone can use, either for free or for a fee, they don't necessarily trust each other, nor should they. What happens if an employee of VeriSign becomes disgruntled and starts issuing bad certificates? There should be checks and audits to ensure that these types of problems can't undermine the entire CA system, just as there are checks and audits to prevent rogue banking employees from crediting their own accounts.

Setting up a CA is the beginning of setting up a very complex security infrastructure for your enterprise. Your CA needs to establish a link of trust from all your users to the administrator or operator of the CA itself, and from your CA to other CAs with which you communicate.

There are two different kinds of CAs: One uses software that you install on your own server inside your enterprise and you maintain; the other is

public servers. Having your own server places the burden on creating and revoking certificates on your security administrator, or whoever is going to operate the CA server. In many cases, these products can be administered from a Web browser after they are installed, and the servers can handle certificates from a wide variety of S/MIME products, one of the few shining spots on the interoperability scene at the moment.

PGP for Business comes with its own version of a certificate server. It runs on a Windows desktop machine and typically is used by the administrator of the entire security apparatus to handle certificates. It can handle only PGP certificates.

Some popular software products that function as certificate servers are listed below.

Vendor	URL	Product
Enterprise CAs:		
Netscape	`www.netscape.com`	Certificate Server
Xcert	`www.xcert.com`	Sentry CA
Public CAs:		
VeriSign	`www.verisign.com`	Secure Server ID
Thawte	`www.thawte.com`	Public CA

Enroll and Acquire Your Certificate

When you have your certificate authority either in mind or installed, you next have to set up how you want to acquire your own certificate.

You have two broad methods: by Web or by e-mail. Actually, you don't have any choice: If you have picked your e-mail product and CA at this point in the process, you have to use whatever method comes with that choice. Netscape Messenger and Microsoft's Outlook Express, among others, make use of their related Web browsers to enroll certificates, as you might suspect. And other products make use of e-mail to send and enroll certificates. For example, Xcert's Sentry CA sends you a message telling you that your certificate has been granted, but in the e-mail it has URLs for both Communicator and Internet Explorer where you can download the certificate and place it inside the appropriate software. Why two different links? Because each product supports a different way of acquiring certificates, of course. So much for standards.

Exchange and Manage Certificates

Now comes the hard part—dealing with the certificates of your correspondents, and managing both theirs as well as other certificates around your corporation.

As we mentioned in our standards section, you need to exchange certificates with your correspondents before you can begin to exchange encrypted e-mail. And that means sending your public key to them, and getting their public keys from them, before you can exchange actual encrypted messages. If you are corresponding with someone who doesn't have the same CA in common, you'll first need to establish a trust relationship and exchange root CA certificates before you can exchange the individual certificates. This is somewhat painful, but when you get the hang of it, it isn't that difficult.

After you begin to exchange more than a few of these certificates, you might think that this is a job for a directory server, and, thankfully, the vendors are already there. The CA server can set up entries in an LDAP directory to keep track of who is issued a certificate, and you can query this LDAP server to find who has them. That is the good news, and indeed the PGP product makes use of its own LDAP server to keep track of its certificates. However, the LDAP server is only used by PGP; if you want a general-purpose LDAP server to keep track of your users, you'll have to install something else.

As a challenge for open systems and interoperability, we installed the Xcert Sentry CA and Netscape's Directory Server on a test network. The Xcert was used to create and manage our certificates for our test corporation, and the entries were placed in the Netscape LDAP directory. We created the certificates using the Netscape browser and stored the information in our Messenger e-mail software. After going through the process described previously, we had a valid certificate and could see it in the Security|Messenger settings. Although the Sentry CA couldn't automatically deposit a certificate in the Netscape LDAP server, we (operating as the security administrator) could do so with a few simple Web forms and keystrokes. So far, so good.

The challenge was trying to pry these certificates loose using other products, such as Outlook Express. There we ran into trouble, mainly because the Netscape software creates the certificate in a nonstandard place in the LDAP directory. According to the standards documents, the certificate should be placed in a particular spot in the LDAP directory schema, called *usercertificate*. Netscape, for whatever reason, places them at a location called *usersmimecertificate*. This meant that non-Netscape products couldn't view the certificates in our directory, because they were looking in the wrong place.

This brings up a very good point: The connection between a user and his or her certificate is tenuous at best. Just because you know that `david@strom.com` is the e-mail address of David Strom and you have his certificate, it doesn't mean that any of your expensive software tools can make this connection. This situation will create all sorts of headaches for your security administrators, and it means that you need to maintain at least two directories on your own machine—one for users and one for certificates.

It would be nice if the address books of our e-mail software could handle this automatically, but they don't.

That's not the only issue with managing certificates. What if someone leaves the company? Or changes his or her e-mail address? Or if you want to use the same certificate, but on several different machines? Most certificates are tied to a particular machine and a particular e-mail address, meaning that any new address will require a new certificate. Again, we find this situation unacceptable.

Encrypt and Decrypt Messages

Now you can finally go and encrypt your messages. Various options are available in your e-mail software to do this, and you can choose to sign a message as well as to encrypt it.

That is the encryption portion. What about the decryption side? If you have done your homework and exchanged certificates as we discussed earlier, then when you receive your encrypted message, it should automatically decrypt and display in plain text. You shouldn't have to do anything else—unless the encryption system is broken by a gateway or product incompatibility.

Futures

The obvious question is whether the Internet needs two standards for secure e-mail.

Proponents for both sides can make superficially compelling arguments. PGP proponents point to a grassroots constituency and a huge installed base of legacy systems. PGP emphasizes privacy for individuals. S/MIME proponents, on the other hand, point to some major vendors and an emphasis on nonrepudiation.

If history is any judge, the PGP side will win because less infrastructure is required to make it work. S/MIME has to solve all the problems that PGP has to solve, plus a few more. However, these things aren't decided overnight. So, our prediction is rather straightforward: The two sides will compete in the Internet marketplace for a couple of years, but ultimately the game is PGP's to lose. It requires less infrastructure and fewer broad agreements to achieve ubiquity.

Endnotes

[0] Our thanks to Dan Backman of *Network Computing* magazine for his help in sharing his lab and providing many valuable insights in the preparation of this article. This article is based, in part, on *Internet Messaging: From the Desktop to the Enterprise,* ISBN 0-13-9786100-4 Prentice-Hall, 1998.

[1] See **http://strom.com/places/smime.html** for details regarding product interoperability testing for encrypted e-mail packages.

[2] There is an alternative to this process. The United Parcel Service has produced a file transfer utility called NetDox, available at **www.netdox.com**. It requires special software to be installed on each computer, and it simplifies the certificate and encryption process somewhat. But this is yet another proprietary solution to the encrypted e-mail problem—something we think goes in the wrong direction.

[3] The article has more in-depth examination of testing MIME interoperability and features of Messenger, Outlook Express, Baltimore's MailSecure, OpenSoft's ExpressMail, and two Worldtalk plug-ins for Eudora and Outlook Express. See "Secure E-Mail Clients: Not Quite Ready for S/MIME Prime Time. Stay Tuned." *Network Computing,* February 1, 1998, **techweb.cmp.com/nc/902/902r2.html**.

MARSHALL T. ROSE is Chief Technology Officer of MessageMedia Inc. (formerly First Virtual Holdings, Inc.). He is responsible for the design, specification, and implementation of several Internet-standard technologies and is an author of over 60 of the Internet's RFCs, and several books on Internet technologies. He can be reached at **mrose@dbc.mtview.ca.us**

DAVID STROM is an independent consultant and frequent speaker at NetWorld+Interop shows around the world, where he teaches a class on e-commerce and Web storefronts. He was founding editor-in-chief of *Network Computing* magazine and has written over a thousand articles for various computer trade publications. He is also publisher of the e-mail newsletter *Web Informant*, an almost-weekly series of essays on Web marketing, technology, and culture. He can be reached at **david@strom.com**

Book Review

IP Multicasting

IP Multicasting: The Complete Guide to Interactive Corporate Networks, by Dave Kosiur, ISBN 0-471-24359-0 Wiley Computer Publishing, 1998, **http://www.wiley.com/compbooks/kosiur**

There is nothing remarkable about the statement: As technology becomes more affordable, applications once limited to power users find their way to the mainstream desktop. Video streaming, audio streaming, collaborative applications, and videoconferencing are all examples of applications once found exclusively on high-end workstations but now making their way to the mainstream desktop. If widespread deployment of these applications is to occur, we must be prepared to supply a supporting infrastructure.

The use of IP multicasting is gaining popularity, but many of the fundamentals that drive this and other network technologies, such as routing protocols and transport protocols, are still being debated. This book supplies a comprehensive view of the state-of-the-art as well as practical procedures one can follow in order to incorporate mulitcasting into existing network topologies.

Organization

Chapter 2 presents an introduction to TCP/IP basics and routing. Chapter 3, The Basics of Multicasting, addresses three sender-based multicasting protocols (ST-II, XTP, and MTP) and concentrates on IP multicast (a receiver-based multicasting protocol). The book would be much easier to follow if this chapter had been combined with Chapter 6.

Chapter 4, Multicast Routing Concepts, Chapter 5, Multicast Routing Protocols, and Chapter 6, Transport Protocols, constitute the heart of this book.

Beginning with basic concepts of unicast routing and routing algorithms, the author extends the models to deal with the problems of routing multicast data. Tree maintenance techniques form the bulk of Chapter 4.

Chapter 5 covers four multicast routing protocols: *Distance Vector Multicast Routing Protocol* (DVMRP); *Multicast Open Shortest Path First* (MOSPF); *Protocol Independent Multicast* (PIM); and *Core-Based Trees* (CBT). Placing the emphasis on PIM, Kosiur covers both PIM-SM (*sparse mode*) and PIM-DM (*dense mode*). He does a nice job of describing each of the protocols and summarizes each by reviewing its advantages and disadvantages. Finally, the author concludes by examining ways of achieving interdomain routing and protocol interoperability.

In Chapter 6, Kosiur provides an overview of the *Real-Time Transport Protocol* (RTP)/*Real-Time Transport Control Protocol* (RTCP) and the *Real-Time Streaming Protocol* (RTSP).

In addition, he discusses a dozen or more multicast protocols, all trying to answer the question: "How is retransmission of lost packets handled?" He classifies the protocol approaches into *receiver-based* or *sender-based*. In my opinion, this is the most interesting problem of multicasting. Answer this question wrong, and you find yourself with a nonscalable network cluttered with acknowledgments (ACKs).

Chapters 4 through 7 all consider delivering Quality of Service and so I was a little surprised to see Chapter 7 devoted to the subject.

Kosiur provides a good introduction to RSVP (*Resource ReserVation Protocol*), but until we see RSVP in wide deployment I would look at the previous three chapters for practical knowledge on the topic. In Chapter 7, and then in Chapter 11 he covers a lot of practical issues concerning Quality of Service, as well as ways to support multicasting over various networks, such as ATM, Frame Relay, and ISDN/dialup networks.

Chapter 9 is a compilation of some free and commercial software packages that use multicasting. Chapter 10 covers *Mbone* (the Multicast backbone), a popular experimental multicasting network. It is arguable that the state of multicasting wouldn't be where it is today without the Mbone.

A C+

This book rates a C+. Kosiur certainly has an understanding of the material, but his descriptions are neither clear nor concise. Reading this book is difficult, and learning from it even more so, but better organization could turn it into a gem.

—Neophytos Iacovou
University of Minnesota
Academic & Distributed Computing Services
iacovou@boombox.micro.umn.edu

Letter to the Editor

I just read the September 1998 issue of *The Internet Protocol Journal* and thoroughly enjoyed it. It was well written with excellent technical detail but more importantly, the contributors wrote in an understandable and organized method. This is not always the norm for good technical resources; so many times it is simply the reprint of a vendor's documentation.

"What is a VPN—Part II," written by Paul Ferguson and Geoff Huston, was a great article which described the various components and methodologies of VPNs. The information and explanation of the Virtual Private Dial Networking implementations, voluntary versus compulsory tunneling, subscriber's perspectives and real world applications clarified my understanding and knowledge on this subject. I also appreciate an article that ends with a conclusion. I have already located Part I of this article and will be reading it soon. There is one comment; it would be interesting to know which vendor when an example is used, regarding specifically the Frame Relay service provider.

The "Reliable Multicast Protocols and Applications" article was useful and informative, including the scaling issues and the information regarding the new reliable multicast protocols. The details of the *Pretty Good Multicast* (PGM) protocol and how it may improve scaling for multicast was very interesting.

The *Gigabit Ethernet* book review written by Ed Tittel was one of the most informative and well structured book reviews that I have read, especially in a smaller publication. Thanks for providing three pages for book reviews in a forty-seven page publication. This review provided all the information that would assist with the determination of purchasing the book or not.

I hope you continue to publish IPJ in hard copy. I do read and gather information from the Web like everyone else, but I prefer a physical copy to carry with me if I am traveling or at my home. Thanks again for a great publication and I can hardly wait for the next issue.

—Joe Brannan
joe.brannan@pepsi.com

Ed.: We appreciate your comments about our publication. Regarding your question about the Frame Relay example, it is our policy to avoid as much as possible any discussion of products, but we encourage readers to contact the authors directly for that kind of information.

We certainly plan to continue the print edition of IPJ. We are also developing a companion Web site (at **www.cisco.com/ipj***) that will contain additional information such as glossaries, links to other documents, updates, corrections, and so on. Thanks for writing.*

—Ole Jacobsen
ole@cisco.com

Fragments

ICANN Update

Back in the summer of 1997, the Clinton Administration decided that it was time to privatize the remaining Internet functions that were being managed within the federal research establishment, mostly dealing with Internet names and addresses. These functions had been handled very successfully over many years by the *Internet Assigned Numbers Authority* (IANA) under the direction Dr. Jon Postel and his staff at the Information Sciences Institute of the University of Southern California under contract to DARPA. But it was clear from the rapid expansion of the Internet, the emergence of important players on the industry side, and rising controversy over issues such as Network Solutions' monopoly in issuing domain names for **.com**, that change was necessary.

After two major policy papers and months of argumentative debate, the government recognized the *Internet Corporation for Assigned Names and Numbers* (ICANN) as the new body to assume responsibility for these largely technical management functions. Working from plans drawn up by Dr. Postel, his advisors and the Jones Day law firm, ICANN is endeavoring to satisfy the many constituencies that seek a voice in future decisions on Internet naming and addressing. Sadly, Jon died last fall just as his plan was approaching endorsement by the federal government.

The young organization, incorporated at the end of September, 1998, began operation in early November, has an initial Board of nine appointed Directors headed by Chairman Esther Dyson, and an interim President/CEO Mike Roberts. They are responsible for completing organizational details, devising a representation structure for electing their successors, and beginning to deal with a backlog of undone policy work, such as a determination on if, how and when new top level domains (TLDs) will be created. The new Board has Directors from six countries and plans to hold meetings quarterly in locations throughout the world, beginning with Singapore in March, 1999 and Berlin in May, 1999.

Being neither a Congressionally chartered corporation nor an industry trade association, but something in between, ICANN is an international organization that faces a tough political future with many skeptics challenging the notion that the Internet community can successfully govern itself in the important naming and addressing area. But with a startup fund from corporate contributions, Chairman Dyson and President Roberts, both short timers by design, are determined to get ICANN off the ground and into operation in coming months. More information is available at: **www.icann.org**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Cisco News Publications Group, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-J4
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

June 1999 Volume 2, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

In this issue, Geoff Huston concludes his two-part article on Interconnection, Peering, and Settlements. Last time Geoff discussed the technical aspects for Internet Service Provider (ISP) interconnection. This time he examines the associated business relationships that arise out of ISP peering arrangements. He also looks at some future directions for the ISP interconnection environment, particularly with respect to Quality-of-Service considerations.

A recurring theme in this journal has been the traditional lack of security in Internet technologies and systems. We have examined several ways in which security has been added at all levels of the protocol stack. This time we look at *firewalls*, a popular way to segregate internal corporate intranet traffic from Internet traffic while still maintaining Internet connectivity. Fred Avolio gives the history of firewalls, their current state, and future directions.

Computer viruses have probably existed for as long as we have had computers. However, the ease with which viruses can be distributed as Internet e-mail attachments has made the problem more prevalent. Recently, the *Melissa* virus achieved some notoriety because of its "self-replication" properties. Barbara Fraser, Lawrence Rogers, and Linda Pesante of the Software Engineering Institute at Carnegie Mellon University examines some of the issues raised by this kind of virus.

This issue is the first anniversary issue of *The Internet Protocol Journal* (IPJ). You can find all of our back issues in PDF format at the IPJ Web site: **www.cisco.com/ipj**. Please let us know if you have suggestions for articles, books you want to review, or general feedback for this journal. Our contact address is: **ipj@cisco.com**.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Interconnection, Peering and Settlements—Part II

by Geoff Huston, Telstra

In Part I we examined the business drivers behind the adoption of the exchange model as the common basis of interconnection, and also examined the advantages and pitfalls associated with the operation of such exchanges within the public Internet. (See *The Internet Protocol Journal,* Volume 2, No. 1, March 1999.) In continuing our examination of the technology and business considerations that are significant within the subject of Internet Service Provider (ISP) interconnection, in this part we focus on the topic from a predominately business perspective.

Interaction Financials: Peering and Settlements

Any large multiprovider distributed service sector has to address the issue of cost distribution at some stage in its evolution. Cost distribution is the means by which various providers can participate in the delivery of a service to a customer who purchases a service from a single provider, and providers can each be compensated for their costs in an equitable structure of interprovider financial settlement.

As an example, when an airline ticket is purchased from one air service provider, various other providers and service enterprises may play a role in the delivery of the service. The customer does not separately pay the service fee of each airport baggage handler, caterer, or other form of service. The customer's original fare, paid to the airline, is distributed to other providers who incurred cost in providing components of the total service. These costs are incurred through sets of service contracts, and are the subject of various forms of interprovider financial settlements, all of which are invisible to the customer.

The Internet is in a very similar situation. Some 50,000 constituent networks must interconnect in one fashion or another to provide comprehensive end-to-end service to each client. In supporting a data transaction between two clients, the two parties often are not clients of the same network. Indeed, the two-client service networks often do not directly interconnect, and one or more additional networks must act in a transit provider role to service the transaction. Within the Internet environment, how do all the service parties to a transaction who incur cost in supporting the transaction receive compensation for their cost? What is the cost distribution model of the Internet?

Here, we examine the basis for Internet interprovider cost distribution models and then look at the business models currently used in the interprovider Internet environment. This area commonly is termed *financial settlement,* a term the Internet has borrowed from the telephony industry.

The Currency of Interconnection

What exactly is being exchanged between two ISPs who want to interconnect? In the sense of the meaning of currency as the circulating medium, the question is: What precisely is being circulated at the exchange and within the realm of interconnection? The technical answer to the question is: *routing entries*. When two parties exchange routing entries, the outcome is that traffic flows in response to the flow of routing entries. The route advertisement and traffic flows move in opposite directions, as indicated in Figure 1, and a bilateral routing-mediated flow occurs only when routes are passed in both directions.

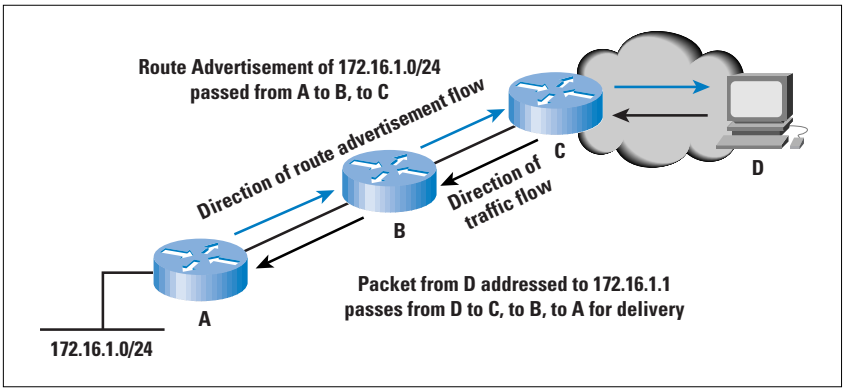


Figure 1: Routing and Traffic Flows

Within the routing environment of an ISP there are many different classes of routes, with the classification based predominately on the way in which the route has been acquired by the ISP:

- *Client routes* are passed into the ISP's routing domain by virtue of a service contract with the client. The routes may be statically configured at the edge of the ISP's network, learned by a *Border Gateway Protocol* (BGP) session with the client, or they may constitute part of an ISP pool of addresses that are dynamically assigned to the client as part of the dialup session.
- *Internal ISP routes* fall into numerous additional categories. Some routes correspond to client services operated by the ISP, solely for access to the clients of the ISP, such as Web caches, *Post Office Protocol* (POP) mail servers, and game servers. Some routes correspond to ISP-operated client services that require Internet-wide access, such as *Domain Name System* (DNS) forwarders and *Simple Mail Transfer Protocol* (SMTP) relay hosts. Lastly are internal services with no visibility outside the ISP network, such as *Simple Network Management Protocol* (SNMP) network management platforms.
- *Upstream routes* are learned from upstream ISPs as part of a transit service contract the ISP has executed with the upstream provider.
- *Peer routes* are learned from exchanges or private interconnections, corresponding to routers exported from the interconnected ISP.

How then should the ISP export routes so that the inbound traffic flow matches the outbound flows implied by this route structure? The route export policy is generally structured along the following lines:

- *Clients:* All available routes in the preceding four categories, with the exception of internal ISP service functions, should be passed to clients, either in the form of a *default route* or as *explicit route entries* passed via a BGP session.
- *Upstream providers:* All client routes and all internal ISP routes corresponding to Internet-wide services should be passed to upstream providers. Some clients may want further restrictions placed on their routes being advertised in such a fashion. The ability for a client to specify such caveats on the routing structure, and the mechanism used by the ISP to allow this to happen, should be clearly indicated in the service contract.
- *Peer ISPs:* All client routes and all ISP routes corresponding to Internet-wide service should be passed to peer ISPs. Again the clients may want to place a restriction on such an advertisement of their routes as a qualification to the ISP's own route export policy.

This structure is shown in Figure 2.

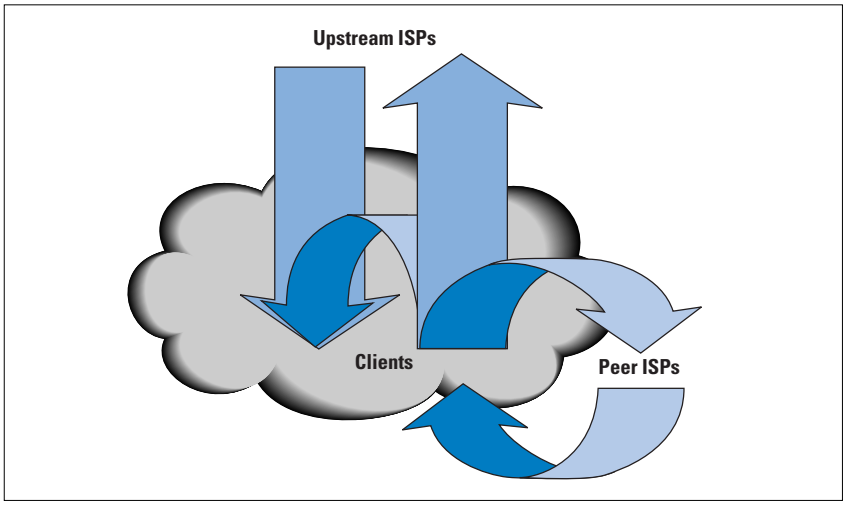


Figure 2: External Routing Interaction

The implicit outcome of this routing policy structure is that the ISP does not act in a transit role to peer ISPs and permits neither peer-to-peer transit nor peer-to-upstream transit. Peer ISPs have visibility only to clients of the ISP. From the service visibility perspective, client-only services are not visible to peer ISPs or upstream ISPs, and, therefore, value-added client services are implicitly visible only to clients and only when they access the service through a client channel.

Settlement Options

Financial settlements have been a continual topic of discussion within the domain of Internet interconnection. To look at the Internet settlement environment, let's first look at the use of interprovider financial settlements within the international telephony service industry. Then, we will look at the application of these generic principles to the Internet environment.

Within the traditional telephony model, interprovider peering takes place within one of three general models:

Bilateral Settlements

The first, and highly prevalent, international peering model is that of bilateral settlements. A *call-minute* is the unit of settlement accounting. A call is originated by a local client, and the local client's service provider charges the client for the duration of the entire end-to-end call. The call may pass through, or transit, many providers, and then terminate within the network of the remote client's local provider. The cost distribution mechanism of settlements is handled bilaterally. In the most general case of this settlement model, the originating provider pays the next hop provider to cover the costs of termination of the call. The next hop provider then either terminates the call within the local network, or undertakes a settlement with the next hop provider to terminate the call. The general telephony trunk model does not admit many multiparty transit arrangements. Most telephony settlements are associated with trunk calls that involve only two providers: the originating and terminating providers.

Within this technology model, the bilateral settlement becomes easier, because the model simplifies to the case where the terminating provider charges the originating provider a per-call-minute cost within an accounting rate that has been bilaterally agreed upon between the two parties. Because both parties can charge each other using the same accounting currency, the ultimate financial settlement is based on the net outcome of the two sets of call-minute transactions with the two call-minute termination accounting rates applied to these calls. (There is no requirement for the termination rates for the two parties to be set at the same level.) Each provider invoices the originating end user for the entire call duration, and the financial settlements provide the accounting balance intended to ensure equity of cost distribution in supporting the costs of the calls made between the two providers. Where there is equity of call accounting rates between the two providers, the bilateral interprovider financial settlements are used in accordance with originating call-minute imbalance, in which the provider hosting the greater number of originating call-minutes pays the other party according to a bilaterally negotiated rate as the mechanism of cost distribution between the two providers.

As a side note, the *Federal Communications Commission* of the United States (FCC) asserts that U.S. telephone operators paid out some $5.6 billion in settlement rates in 1996, and the FCC is voicing the view that accounting rates have now shifted into areas of non-cost-based settings, rather than working as a simple cost distribution mechanism.

This accounting settlement issue is one of the drivers behind the increasing interest in voice-over-IP solutions, because typically no accounting rate settlement component exists in such solutions, and the call termination charges are cost-based, without bilateral price setting. In those cases

where accounting rates have come to dominate the provider's call costs, voice-over-IP is perceived as an effective lever to bypass the accounting rate structure and introduce a new price point for call termination in the market concerned.

Sender Keeps All

The second model, rarely used in telephony interconnection, is that of *Sender Keeps All* (SKA), in which each service provider invoices its originating client's user for the end-to-end services, but no financial settlement is made across the bilateral interconnection structure. Within the bilateral settlement model, SKA can be regarded as a boundary case of bilateral settlements, where both parties simply deem the outcome of the call accounting process to be absolutely equal, and consequently no financial settlement is payable by either party as an outcome of the interconnection.

Transit Fees

The third model is that of transit fees, in which one party invoices the other party for services provided. For example, this arrangement is commonly used as the basis of the long-distance/local access provider interconnection arrangements. Again, this case can be viewed as a boundary case of a general bilateral settlement model, where in this case the parties agree to apply call accounting in only one direction, rather than bilaterally.

Telephony Settlement Trends

The international telephony settlement model is by no means stable, and currently, significant pressure is being placed on the international accounting arrangements to move away from bilaterally negotiated uniform call accounting rates to rates separately negotiated for calls in each direction of a bilateral interconnection. Simultaneously, communications deregulation within many national environments is changing the transit fee model, as local providers extend their network into the long-distance area and commence interconnection arrangements with similar entities. Criticism also has been directed at the bilaterally negotiated settlement rates, because of the observation that in many cases the accounting rates are not cost-based rates but are based on a desire to create a revenue stream from accounting settlements.

Internet Considerations

Numerous critical differences exist between the telephony models of interconnection and the Internet environment; these differences have confounded all attempts to cleanly map telephony interconnection models into the Internet environment.

Internet Settlement Accounting by the Packet

Internet interconnection accounting is a packet-based accounting issue, because there is no "call-minute" in the Internet architecture. Therefore, the most visible difference between the two environments is the replacement of the *call* with the *packet* as the currency unit of interconnection.

Although we can argue that a TCP session has much in common with a call, this concept of an originating TCP call-minute is not always readily identified within the packet forwarding fabric, and accordingly it is not readily apparent that this is a workable settlement unit. Unlike a telephony call, no concept of state initiation exists to pass a call request through a network and lock down a network transit path in response to a call response. The network undergoes no state change in response to a TCP session, and therefore, no means is readily available to the operator to identify that a call has been initiated, and by which party. Of course the use of *User Datagram Protocol* (UDP), and various forms of tunnelling traffic, also confound any such TCP call-minute accounting mechanism.

Packets may be dropped

When a packet is passed across an interconnection from one provider to another, no firm guarantee is given by the second provider that the packet will definitely be delivered to the destination. The second provider, or subsequent providers in the transit path, may drop the packet for quite legitimate reasons, and will remain within the protocol specification in so doing. Indeed, the TCP protocol uses packet drop as a rate-control signal. For the efficient operation of the TCP protocol, some level of packet drop is a useful and anticipated event. However, if a packet is used as the accounting unit in a general cost distribution environment, should the provider who receives and subsequently drops the packet be able to claim an accounting credit within the interconnection? The logical response is that such accounting credits should apply only to successfully delivered packets, but such an accounting structure is highly challenging to implement accurately within the Internet environment.

Packet paths are not predetermined

Packet transit paths can be within the explicit control of the end user, not the provider. Users can exercise some significant level of control of the path a packet takes to transit the Internet if source routing is honored, so that the relative packet flows between two providers can be arbitrarily manipulated by any client, if so desired.

Routing and traffic flow are not paired

Packet forwarding is not a verified operation. A provider may choose to forward a packet to a second provider without reference to the particular routes the second provider is advertising to the first party. A packet may also be forwarded to the second provider with a source address that is not being advertised to the second provider. Given that the generic Internet architecture strives for robustness under extreme conditions, attempts to forward a packet to its addressed destination are undertaken irrespective of how the packet may have arrived at this location in the first place, and irrespective of how a packet with reverse header IP addresses will transit the network.

Comprehensive routing information is not uniformly available

Complete information is not available to the Internet regarding the status and reachability of every possible Internet address. Only as a packet is forwarded closer to the addressed destination does more complete information regarding the status of the destination address become apparent to the provider. Accordingly, a packet may have incurred some cost of delivery before its ultimate undeliverability becomes evident. An intermediate transit provider can never be completely assured that a packet is deliverable.

Settlement Models for the Internet

Where a wholesale or retail service agreement is in place, one ISP is, in effect, a customer of the other ISP. In this relationship, the customer ISP (downstream ISP) is purchasing transit and connectivity services from the supplier ISP (upstream ISP). The downstream ISP resells this service to its clients. The upstream ISP must announce the downstream ISP's routes to all other customers and other egress points of the ISP's networks to honor the service contract to the downstream ISP customer.

However, given two ISPs who interconnect, the decision as to which party should assume the upstream provider role and which party should assume the downstream customer role is not always immediately obvious to either party, or even to an outside observer. Greater geographic coverage may be the discriminator here that allows the customer/provider determination. However, this factor is not the only possible one within the scope of the discussion. One ISP may host significant content and may observe that access to this content adds value to the other party's network, which may be used as an offset against a more uniform customer relationship. In a similar vein, an ISP with a very large client population within a limited geographic locality may see this large client base as an offset against a more uniform customer relationship with the other provider. In many ways, the outcome of these discussions can be likened to two animals meeting in the jungle at night. Each animal sees only the eyes of the other, and from this limited input, they must determine which animal should attempt to eat the other!

An objective and stable determination of which ISP should be the provider and which should be the client is not always possible. In many contexts, the question is inappropriate, given that for some traffic classes the respective roles of provider and client may swap over. The question often is rephrased along the lines of, "Can two providers interconnect without the implicit requirement to cast one as the provider and the other as the client?" Exploration of some concepts of how the question could possibly be answered is illustrative of the problem space here.

Packet Cost Accounting

One potential accounting model is based on the observation that a packet incurs cost as it is passes through the network. For a small interval of time, the packet occupies the entire transmission capacity of each circuit over which it passes.

Similarly, for a brief interval of time, the packet is exclusively occupying the switching fabric of the router. The more routers the packet passes through, and the greater the number and distance of transmission hops the packet traverses, the greater the incurred cost in carrying the packet.

A potential settlement model could be constructed from this observation. The strawman model is that whenever a packet is passed across a network boundary, the packet is effectively sold to the next provider. The sale price increases as the packet transits through the network, accumulating value in direct proportion to the distance the packet traverses within the network. Each boundary packet sale price reflects the previous sale price, plus the value added in transiting the ISP's infrastructure. Ultimately, the packet is sold to the destination client. This model is indicated in Figure 3.

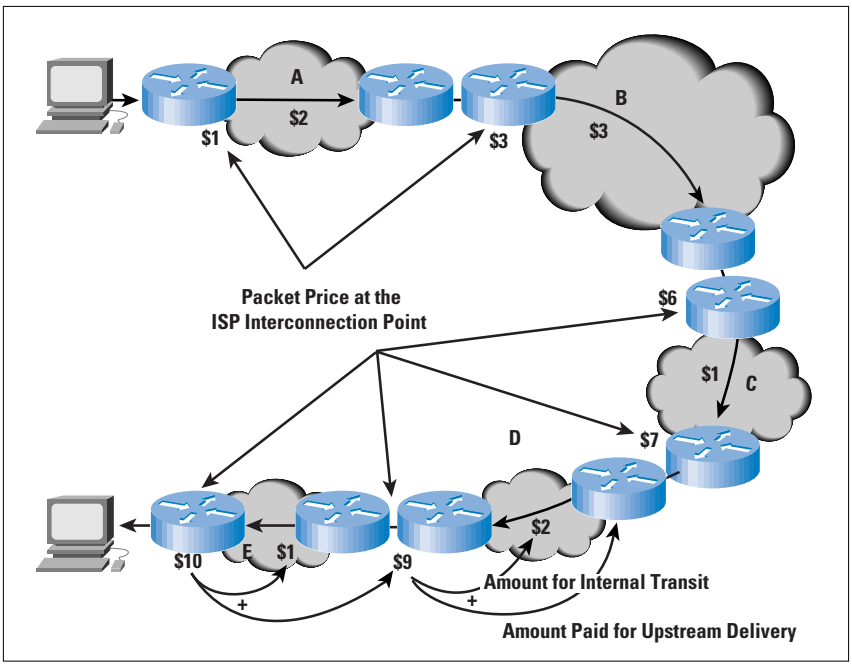


Figure 3: Financial Interprovider Settlement via Packet Cost Accounting

As with all strawman models, this one has numerous critical weaknesses, but let's look at the strengths first. An ISP gains revenue from a packet only when delivered on egress from the network, rather than in network ingress. Accordingly, a strong economic incentive exists to accept packets that will not be dropped in transit within the ISP, given that the transmission of the packet generates revenue to the ISP only on successful delivery of the packet to the next hop ISP or to the destination client. This factor places strong pressure on the ISP to maintain quality in the network, because dropped packets imply foregone revenue on local transmission. Because the packet was already purchased from the previous provider in the path, packet loss also implies financial loss. Strong pressure also is exerted to price the local transit function at a commodity price level, rather than attempt to undertake opportunistic pricing. If the chosen transit price is too great, the downstream provider has the opportunity to extend its network to reach the next upstream

provider in the path, resulting in bypassing the original upstream ISP and purchasing the packets directly from the next hop upstream source. Accordingly, this model of per-packet pricing, using a settlement model of egress packet accounting, and locally applied value increments to a cumulative per-packet price, based on incremental per-hop transmission costs, does allow for some level of reasonable stability and cost distribution in the interprovider settlement environment.

However, weaknesses of this potential model cannot be ignored. First, some level of packet drop is inevitable, irrespective of traffic load. Generally, the more remote the sender from the destination, the less able the sender is to ascertain that the destination address is a valid IP address, and the destination host is available. To minimize the liability from such potential packet loss, the ISP should maintain a relatively complete routing table and accept only packets in which a specific route is maintained for the network. More critical is the issue that the mechanism is open to abuse. Packets that are generated by the upstream ISP can be transmitted across the interface, which in turn results in revenue being generated for the ISP. Of course, per-packet accounting within the core of the network is a significant refinement of existing technology. Within a strict implementation of this model, packets require the concept of an attached value that ISPs augment on an ingress-to-egress basis, which could be simplified to a hop-by-hop value increment. Implementations feasibly can use a level of averaging to simplify this process by using a tariff for domestic transit and a second for international transit.

TCP Session Accounting

These traffic-based metrics do exhibit some weaknesses because of their inability to resist abuse and the likelihood of exacting an interprovider payment even when the traffic is not delivered to an ultimate destination. Of more concern is that this settlement regime has a strong implication in the retail pricing domain, where the method of payment on delivered volume and distance is then one of the more robust ways that a retail provider can ensure that there is an effective match between the interprovider payments and the retail revenue. Given that there is no intrinsic match of distance, and therefore cost, to any particular end-to-end network transaction, such a retail tariff mechanism would meet with strong consumer resistance.

Does an alternative settlement structure that can address these weaknesses exist? One approach is to perform significantly greater levels of analysis of the traffic as it transits a boundary between a client and the provider, or between two providers, and to adopt financial settlement measures that match the type of traffic being observed. As an example, the network boundary could detect the initial TCP SYN handshake, and all subsequent packets within the TCP session could be accounted against the session initiator, while UDP traffic could be accounted against the UDP source. Such detailed accounting of traffic passed across a provider boundary could allow for a potential settlement structure based on duration (*call-minutes*), or volume (*call-volumes*).

Although such settlement schemes are perhaps limited more by imagination in the abstract, very real technical considerations must be borne to bear on this speculation. For a client-facing access router to detect a TCP flow and correctly identify the TCP session initiator requires the router to correctly identify the initial SYN handshake, the opening packet, and then record all in-sequence subsequent packets within this TCP flow against this accounting element. This identification process may be completely impossible within the network at an interprovider boundary. The outcome of the routing configuration may be an asymmetric traffic path, so that a single interprovider boundary may see only traffic passing in a single direction.

However, the greatest problem with this, or any other traffic accounting settlement model, is the diversity of retail pricing structures that exist within the Internet today. Some ISPs use pricing based on received volume, some on sent volume, some on a mix of sent and received volume, and some use pricing based on the access capacity, irrespective of volume. This discussion leads to the critical question when considering financial settlements: Given that the end client is paying the local ISP for comprehensive Internet connectivity, when a client's packet is passed from one ISP to another at an interconnection point, where is the revenue for the packet? Is the revenue model one in which the packet sender pays or one in which the packet receiver pays? The packet egress model described here assumes a uniform retail model in which the receiver pays for Internet packets. The TCP session model assumes the session initiator pays for the entire traffic flow. This uniformity of retail pricing is simply not mirrored within the retail environment of the Internet today.

Although this session-based settlement model does attempt to promote a quality environment with fair carriage pricing, it cannot address the fundamental issue of financial settlements.

Internet Settlement Structures

For a financial settlement structure to be viable and stable, the settlement structure must be a uniform abstraction of a relatively uniform retail tariff structure. This conclusion is critically important to the entire Internet financial settlement debate.

The financial structure of interconnection must be an abstraction of the retail models used by the two ISPs. If the uniform retail model is used, the party originating the packet pays the first ISP a tariff to deliver the packet to its destination within the second ISP; then the first ISP is in a position to fund the second ISP to complete the delivery through an interconnection mechanism. If, on the other hand, the uniform retail model is used in which the receiver of the packet funds its carriage from the sender, then the second ISP funds the upstream ISP. If no uniform retail model is used, when a packet is passed from one provider to the other, no understanding exists about which party receives the revenue for the carriage of the packet and accordingly, which party settles with

the other party for the cost incurred in transmission of the packet. The answer to these issues within the Internet environment has been to commonly adopt just two models of interaction. These models sit at the extreme ends of the business spectrum, where one is a customer/provider relationship, and the other is a peering relationship without any form of financial settlement, or SKA. These models approximately correspond to the second and third models described previously from traditional models of interconnection within the communications industry. However, an increasing trend has moved toward models of financial settlement in a bilaterally negotiated basis within the Internet, using non-cost-based financial accounting rates within the settlement structure. Observing the ISP industry repeat the same well-trodden path, complete with its byways into various unproductive areas and sometimes mistakes of the international telephony world, is somewhat interesting to say the least. Experiential learning is often observed to be a rare commodity in this area of Internet activity.

No Settlement and No Interconnection

Examining the option of complete autonomy of operation, without any form of interaction with other local or regional ISPs, is instructive within this examination of settlement options.

One scenario for a group of ISPs is that a mutually acceptable peering relationship cannot be negotiated, and all ISPs operate disconnected network domains with dedicated upstream connections and no interconnection. The outcome of such a situation is that third-party connectivity would take place, with transit traffic flowing between the local ISPs being exchanged within the domain of a mutually connected third-party ISP (or via transit across a set of third-party ISPs). For example, for an Asian country, this situation would result in traffic between two local entities, both located within the same country, being passed across the Pacific, routed across numerous network domains within the United States, and then passed back across the Pacific. Not only is this scenario inefficient in terms of resource utilization, but this structure also adds a significant cost to the operation of the ISPs, a cost that ultimately is passed to the consumer in higher prices for Internet traffic.

Note that this situation is not entirely novel; the Internet has seen such arrangements appear in the past; and these situations are still apparent in today's Internet. Such arrangements have arisen, in general, as the outcome of an inability to negotiate a stable local peering structure.

However, such positions of no interconnection have proved to be relatively short-lived because of the high cost of operating international transit environments, the instability of the significantly lengthened interconnection paths, and the unwillingness of foreign third-party ISPs to act (often unwittingly) as agents for domestic interconnection in the longer term. As a result of these factors, such off-shore connectivity structures generally have been augmented with domestic peering structures.

The resultant general operating environment of the Internet is that effective isolation is not in the best interests of the ISP, nor is isolation in the interests of other ISPs or the consumers of the ISPs' services. In the interests of a common desire to undertake rational and cost-effective use of communications resources, each national (or regional) collection of ISPs acts to ensure local interconnectivity between such ISPs. A consequent priority is to reach acceptable ISP peering arrangements.

Sender Keeps All

Sender Keeps All (SKA) peering arrangements are those in which traffic is exchanged between two or more ISPs without mutual charge (an interconnection arrangement with no financial settlement). Within a national structure, typically the marginal cost of international traffic transfer to and from the rest of the Internet is significantly higher than domestic traffic transfer. In these cases, any SKA peering is likely to relate to only domestic traffic, and international transit would be provided either by a separate agreement or independently by each party.

This SKA peering model is most stable where the parties involved perceive equal benefit from the interconnection. This interconnection model generally is used in the context of interconnection or with providers with approximate equal dimension, as in peering regional providers with other regional providers, national providers with other national providers, and so on. Oddly enough, the parties themselves do not have to agree on what that value or dimension may be in absolute terms. Each party makes an independent assessment of the value of the interconnection, in terms of the perceived size and value of the ISP and the value of the other ISP. If both parties reach the conclusion that in their terms a net balance of value is achieved, then the interconnection is on a stable basis. If one party believes that it is larger than the other and SKA interconnection would result in leverage of its investment by the smaller party, then an SKA interconnection is unstable.

The essential criterion for a stable SKA peering structure is perceived equality in the peering relationship. This criterion can be achieved in many ways, including the use of entry threshold pricing into the peering environment or the use of peering criteria, such as the specification of ISP network infrastructure or network level of service and coverage areas as eligibility for peering.

A typical feature of the SKA peering environment is to define an SKA peering in terms of traffic peering at the client level only. This definition forces each peering ISP to be self-sufficient in the provision of transit services and ISP infrastructure services that would not be provided across a peering point. This process may not result in the most efficient or effective Internet infrastructure, but it does create a level of approximate parity and reduces the risks of leverage within the interconnection. In this model, each ISP presents at each interconnection or exchange only those routes associated with the ISP's customers and accepts only traffic

from peering ISPs at the interconnection or exchange directed to such customers. The ISP does not accept transit traffic destined to other remote exchange locations, nor to upstream ISPs, nor traffic directed to the ISP's infrastructure services. Equally, the ISP does not accept traffic that is destined to peering ISPs, from upstream transit providers. The business model here is that clients of an ISP are contracting the ISP to present their routes to all other customers of the ISP, to the upstream providers of the ISP, and to all exchange points where the ISP has a presence. The particular tariff model chosen by the ISP in servicing the customers is not material to this interconnection model. Traffic passed to a peer ISP at the exchange becomes the responsibility of the peer ISP to pass to its customers at its cost.

Another means of generating equity within an SKA peering is to peer only within the terms of a defined locality. In this model, an ISP would present routes to an SKA peer in which the routes correspond to customers located at a particular access POP, or a regional cluster of access POPs. The SKA peer's ability to leverage advantage from the greater level of investment (assuming that the other party is the smaller party) is now no longer a factor, because the smaller ISP sees only those parts of the larger ISP that sit within a well-defined local or regional zone. This form of peering is indicated in Figure 4.

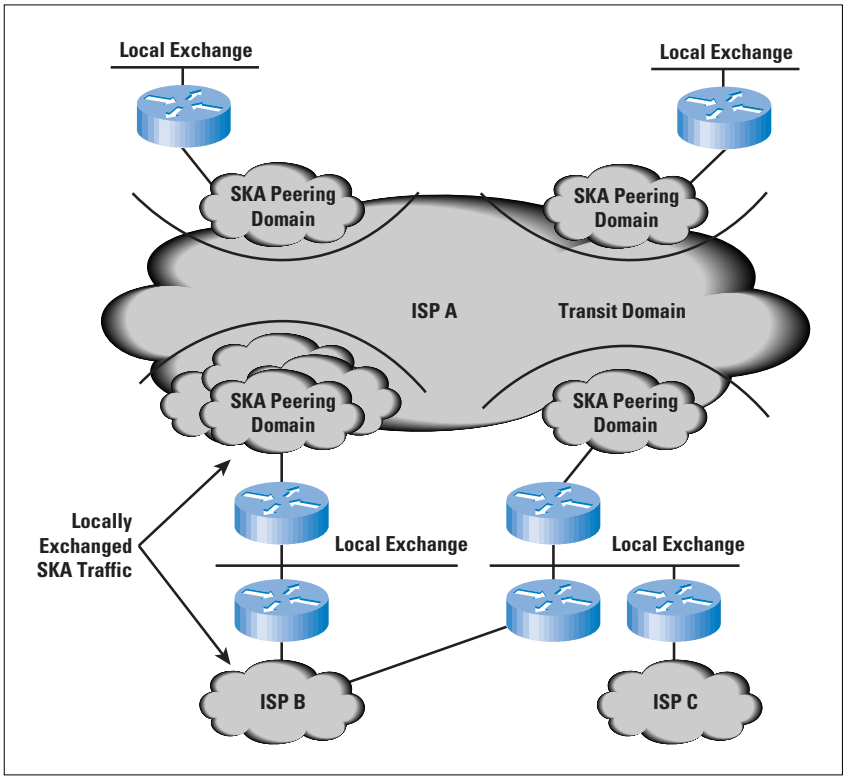
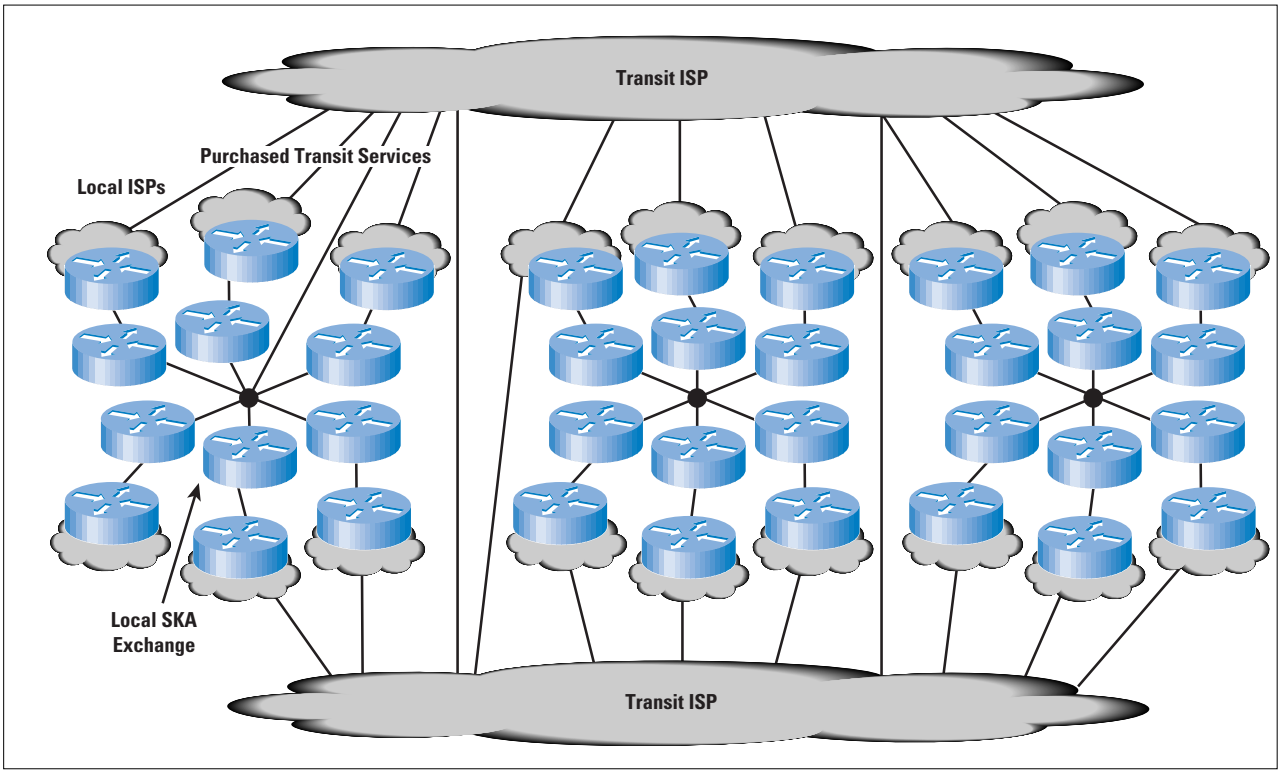


Figure 4: SKA Peering Using Local Cells

The probable outcome of widespread use of SKA interconnections is a generalized ISP domain along the lines of Figure 5. Here, the topology is segregated into two domains consisting of a set of transit ISPs, whose predominate investment direction is in terms of high-capacity carriage infrastructure and high-capacity switching systems, and a collection of local ISPs, whose predominate investment direction is in service infrastructure supporting a string retail focus. Local ISPs participate at exchanges and announce local routes at the exchange on an SKA basis of interconnection with peer ISPs. Such ISPs are strongly motivated to prefer to use all routes presented at the exchange within such peering sessions, because the ISP is not charged any transit cost for the traffic under an SKA settlement structure. The exchange does not provide comprehensive connectivity to the ISP, and this connectivity needs to be complemented with a separate purchase of transit services. In this role, the local ISP becomes a client of one or more transit ISPs explicitly for the purpose of access to transit connectivity services.

Figure 5: ISP Structure of Local and Transit Operations



In this model, the transit ISP must have established a position of broad-ranging connectivity, with a well-established and significant market share of the wholesale transit business. A transit ISP also must be able to present customer routes at a carefully selected set of major exchange locations and have some ability to exchange traffic with all other transit ISPs. This latter requirement has typically been implemented using private interconnection structures, and the associated settlements often are negotiated bilaterally. These settlements possibly may include some element of financial settlement.

Negotiated Financial Settlement

The alternative to SKA and provider/client role selection is the adoption of a financial settlement structure. The settlement structure is based on both parties effectively selling services to each other across the interconnection point, with the financial settlement undertaking the task of balancing the relative sales amounts.

The simplest form of undertaking this settlement is to measure the volume of traffic being passed in each direction across the interconnection and to use a single accounting rate for all traffic. At the end of each accounting period, the two ISPs would financially settle based on the agreed accounting rate applied to the net traffic flow.

Which way the money should flow in relationship to traffic flow is not immediately obvious. One model assumes that the originating provider should be funding the terminating provider to deliver the traffic, and therefore, money should flow in the same direction as traffic. The reverse model assumes that the overall majority of traffic, is traffic generated in response to an action of the receiver, such as web page retrieval or the downloading of software. Therefore, the total network cost should be imposed on the discretionary user, so that the terminating provider should fund the originating provider. This latter model has some degree of supportive evidence, in that a larger provider often provides more traffic to a smaller attached provider than it receives from that provider. Observation of bilateral traffic flow statistics tends to support this, indicating that traffic-received volumes typically coincide with the relative interconnection benefit to the two providers.

The accounting rate can be negotiated to be any amount. There is a caveat on this ability to set an arbitrary accounting rate, because where an accounting rate is not cost-based, business instability issues arise. For greater stability, the agreed settlement traffic unit accounting rate would have to match the average marginal cost of transit traffic in both ISP networks for the settlement to be attractive to both parties. Refinements to this approach can be introduced, although they are accompanied by significant expenditure on traffic monitoring and accounting systems. The refinements are intended to address the somewhat arbitrary determination of financial settlement based on the receiver or the sender. One way is to undertake flow-based accounting, in which the cost accounting for the volume of all packets associated with a TCP flow is directed to the initiator of the TCP session. Here, the cost accounting for all packets of a UDP flow is directed to the UDP receiver. The session-based accounting is significantly more complex than simple volume accounting, and such operational complexity would be reflected in the cost of undertaking such a form of accounting. However, asymmetric paths are a common feature of the inter-AS environment, so that it may not always be possible to see both sides of a TCP conversation and perform an accurate determination of the session initiator.

Another refinement is to use a different rate for each provider, where the base rate is adjusted by some agreed size factor to ensure that the larger provider is not unduly financially exposed by the arrangement. The adjustment factor can be the number of Points of Presence, the range of the network, the volume carried on the network, the number of routes advertised to the peer, or any other metric related to the ISP's investment and market share profile. Alternatively, a relative adjustment factor can simply be a number, without any basis in a network metric, to which both parties agree.

Of course, such a relative traffic volume balance is not very robust either, and the metric is one that is vulnerable to abuse. The capability to adjust the relative traffic balance comes from the direct relationship between the routes advertised and the volume of traffic received. To reduce the amount of traffic received, the ISP reduces the number of routes advertised to the corresponding peer. Increasing the number of routes, and at the same time increasing the number of specific routes, increases the amount of received traffic. When there is a rich mesh of connectivity, the primary objective of routing policy is no longer that of supporting basic connectivity, but instead the primary objective is to maximize the financial return to the operator. If the ISP is paying for an "upstream" ISP service, the motivation is to minimize the cost of this contract, either by maximizing the amount of traffic covered under a fixed cost, or minimizing the cost by minimizing the traffic exchanged with the upstream ISP. Where there is a financially settled interconnection, the ISP will be motivated to configure its routing policies to maximize its revenue from such an arrangement. And of course an ISP will always prefer to use customer routes wherever possible, as a basic means of maximizing revenue into the operation.

Of greater concern is the ability to abuse the interconnection arrangements. One party can generate and then direct large volumes of traffic to the other party. Although overt abuse of the arrangements is often easy to detect, greed is a wonderful stimulant to ingenuity, and more subtle forms of abuse of this arrangement are always possible. To address this, both parties would typically indicate in an interconnection agreement their undertaking not to indulge in such forms of deliberate abuse.

Notwithstanding such undertakings by the two providers, third parties can still abuse the interconnection in various ways. Loose source routing can generate traffic flows that pass across the interconnection in either direction. The ability to remotely trigger traffic flows through source address spoofing is possible, even where loose source routing is disabled. This window of financial vulnerability is far wider than many ISPs are comfortable with, because it opens the provider to a significant liability over which it has a limited ability to detect and control. Consequently, financial settlement structures based on traffic flow metrics are not a commonly deployed mechanism, because they introduce significant financial risks to the ISP interconnection environment.

The Settlement Debate

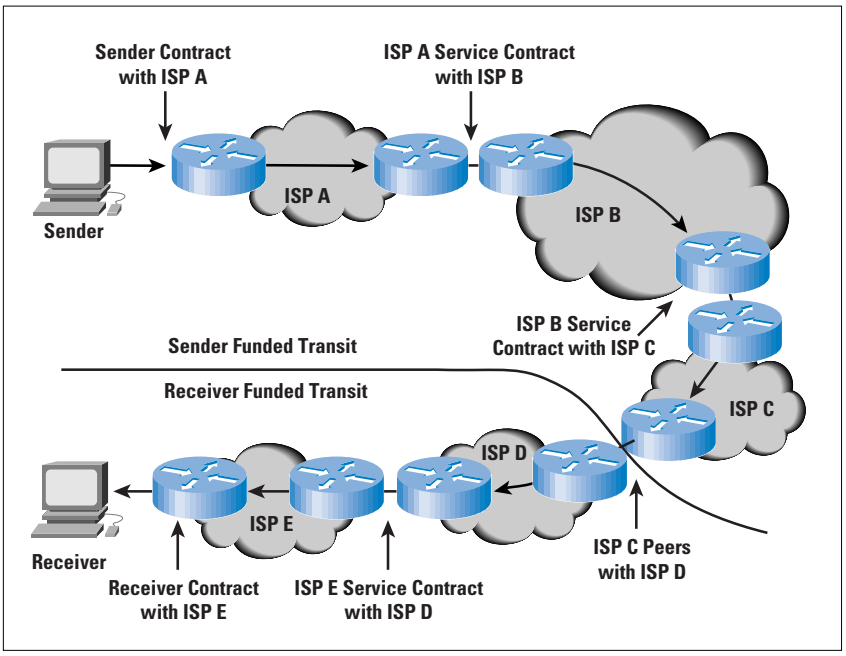
The issue of Internet settlements, and associated financial models of settlement, has occupied the attention of a large number of ISPs, traditional communications carriers, public regulators, and many other interested bodies for many years now. Despite these concentrated levels of attention and analysis, the Internet interconnection environment remains one where there are no soundly based models of financial settlement in widespread use today.

It is useful to look further into this matter, and pose the question: "Why has the Internet managed to pose such a seemingly intractable challenge to the ISP industry?" The prime reason is likely to be found within the commonly adopted retail model of ISP services. The tariff for an ISP retail service does not implicitly cover the provision of an Internet transmission service from the client to all other Internet-connected hosts. In other words, the Internet service, as retailed to the client, is not a comprehensive end-to-end service.

In a simple model of the operation of the Internet, each ISP owns and operates some local network infrastructure, and may choose to purchase services from one or more upstream service providers. The service domain offered to the clients of this network specifically encompasses an Internet subdomain limited to the periphery of the ISP network together with the periphery of the contracted upstream provider's service domain. This is a recursive domain definition, in that the upstream provider in turn may have purchased services from an upstream provider at the next tier, and so on. After the client's traffic leaves this service domain, the ISP ceases to directly, or indirectly, fund the carriage of the client's traffic, and the funding burden passes over to a funding chain linked to the receiver's retail service.

For example, when traffic is passed from an ISP client to a client of another provider, the ISP funds the traffic as it transits through the ISP and indirectly funds the cost of carriage through any upstream provider's network. When the traffic leaves the provider's network, to be passed to either a different client, another ISP, or to a peer provider, the sender's ISP ceases to fund the further carriage of the traffic. This scenario is indicated in Figure 6. In other words, these scenarios illustrate the common theme that the retail base of the Internet is not an end-to-end tariff base. The sender of the traffic does not fund the first hop ISP for the total costs of carriage through the Internet to the traffic's destination, nor does the ultimate receiver pay the last hop ISP for these costs. The ISP retail pricing structure reflects an implicit division of cost between the two parties, and there is no consequent structural requirement for inter-provider financial balancing between the originating ISP and the terminating ISP.

Figure 6: Partial-Path Paired Services



An initial reaction to this partial service model would be to wonder why the Internet works at all, given that no single party funds the carriage of traffic on the complete path from sender to receiver. Surely this would imply that once the traffic had passed beyond the sending ISP's service funded domain the traffic should be discarded as unfunded traffic? The reason why this is not the case is that the receiver implicitly assumes funding responsibility for the traffic at this handover point, and the second part of the complete carriage path is funded by the receiver. In an abstract sense, the entire set of connectivity paths within the Internet can be viewed as a collection of bilaterally funded path pairs, where the sender funds the initial path component and the receiver funds the second terminating path component. This underscores the original observation that the generally adopted retail model of Internet services is not one of end-to-end service delivery, but instead one of partial path service, with no residual retail price component covering any form of complete path service.

Financial settlement models typically are derived from a different set of initial premises than those described here. The typical starting point is that the retail offering is a comprehensive end-to-end service, and that the originating service provider utilizes the services of other providers to complete the delivery of all components of the retailed service. The originating service provider then undertakes some form of financial settlement with those providers who have undertaken some form of an operational role in providing these service elements. This cost-distributed business structure allows both small and large providers to operate with some degree of financial stability, which in turn allows a competitive open service market to thrive. Through the operation of open competition, the consumer gains the ultimate price and service benefit of cost-efficient retail services.

The characteristics of the Internet environment tend to create a different business environment to that of a balanced cost distribution structure. Here there is a clear delineation between a customer/provider relationship and a peer relationship, with no stable middle ground of a financially settled inter-ISP bilateral relationship. An ISP customer is one that assumes the role of a customer of one or a number of upstream providers, with an associated flow of funding from the customer to the upstream provider, whereas an ISP upstream service provider views the downstream provider as a customer. An ISP peer relationship is where the two ISPs execute a peering arrangement, where traffic is exchanged between the two providers without any consequent financial settlement, and such peering interactions are only stable while both providers perceive some degree of parity in the arrangement; for example, when the two providers present to the peering point Internet domains of approximate equality in market coverage and market share. An ISP may have multiple simultaneous relationships, being a customer in some cases, an upstream provider in others, and a peer in others. In general, the relationships are unique within an ISP pairing, and efforts to support a paired relationship which encompasses elements of both peering and customer/provider pose significant technical and business challenges.

The most natural business outcome of any business environment is for each provider to attempt to optimize its business position. For an ISP, this optimization is not simply a case of a competitive impetus to achieve cost efficiency in the ISP's internal service operation, because the realization of cost efficiencies within the service provider's network does not result in any substantial change in the provider's financial position with respect to upstream costs or peering positioning. The ISP's path toward business optimization includes a strong component of increasing the size and scope of the service provider operation, so that the benefits of providing funded upstream services to customers can be maximized, and non-financially settled peering can be negotiated with other larger providers.

The conclusion drawn is that the most natural business outcome of today's Internet settlement environment is one of aggregation of providers, a factor quite evident in the Internet provider environment at present.

Quality of Service and Financial Settlements

Within today's ISP service model, strong pressure to change the technology base to accommodate more sophisticated settlement structures is not evident. The fundamental observation is that any financial settlement structure is robust only where a retail model exists that is relatively uniform in both its nature and deployment, and encompasses the provision of services on an end-to-end basis. Where a broad diversity of partial-service retail mechanisms exists within a multiprovider environment, the stability of any form of interprovider financial settlement structure will always be dubious at best.

If paired partial path service models and SKA peering interconnection comfortably match the requirements of the ISP industry today, is this entire financial settlement issue one of simple academic interest?

Perhaps the strongest factor driving change here is the shift towards an end-to-end service model associated with the current technology impetus toward support of distinguished *Quality of Service* (QoS) mechanisms. Where a client signals the requirement for some level of preemption or reservation of resources to support an Internet transaction or flow, the signal must be implemented on an end-to-end basis in order for the service request to have any meaning or value. The public Internet business model to support practical use of such QoS technologies will shift to that of the QoS signal initiator undertaking to bear the cost of the entire end-to-end traffic flow associated with the QoS signal. This is a retail model where the application initiator undertakes to fund the entire cost of data transit associated with the application. This model is analogous to the end-to-end retail models of the telephony, postal, and freight industries. In such a model, the participating agents are compensated for the use of their services through a financial distribution of the original end-to-end revenue, and a logical base for inter-agent financial settlements is the outcome. It is, therefore, the case that meaningful inter-provider financial settlements within the Internet industry are highly dependent on the introduction of end-to-end service retail models. There financial settlements are, in turn, dependent on a shift from universal deployment of a best effort service regime with partial path funding to the introduction of layered end-to-end service regimes that feature both end-to-end service-level undertakings and end-to-end tariffs applied to the initiating party.

The number of conditionals in this argument is not insignificant. If QoS technologies are developed that scale to the size of the public Internet, that provide sufficiently robust service models to allow the imposition of service level agreements with service clients, and are standardized such that the QoS service models are consistent across all vendor platforms, then this area of inter-provider settlements will need to change as a consequence. The pressure to change will be emerging market opportunities to introduce interprovider QoS interconnection mechanisms and the associated requirement to introduce end-to-end retail QoS services. The consequence is that there will be pressure to support this with inter-provider financial settlements where the originating provider will apportion the revenue gathered from the QoS signal initiator with all other providers that are along the associated end-to-end QoS flow path.

Such an end-to-end QoS settlement model assumes significant proportions that may in themselves impact on the QoS signaling technologies. It is conceivable that each provider along a potential QoS path may need to signal not only their capability of supporting the QoS profile of the potential flow, but also the unit settlement cost that will apply to the flow. The end user may then use this cost feedback to determine

whether to proceed with the flow given the indication of total transit costs, or request alternate viable paths in order to choose between alternative provider paths so as to optimize both the cost and the resultant QoS service profile. The technology and business challenges posed by such an end-to-end QoS deployment model are certainly an impressive quantum change from today's best effort Internet.

With this in mind, one potential future is that the public Internet environment will adopt a QoS mediated service model that is capable of supporting a diverse competitive industry through interprovider financial settlements. The alternative is the current uniform best effort environment with no logical role for interprovider settlements, with the associated strong pressures for provider aggregation. The reliance on Internet QoS technologies to achieve not only Internet service outcomes, but also to achieve desired public policy outcomes in terms of competitive pressures, is evident within this perspective. It is unclear whether the current state of emerging QoS technologies and QoS interconnection agreements will be able to mature and be deployed in time to forge a new chapter in the story of the Internet interconnection environment. The prognosis for this is, however, not good.

Futures

Without the adoption of a settlement regime that supports some form of cost distribution among Internet providers, there are serious structural problems in supporting a diverse and well populated provider industry sector. These problems are exacerbated by the additional observation that the Internet transmission and retail markets both admit significant economies of scale of operation. The combination of these two factors leads to the economic conclusion that the Internet market is not a sustainable open competitive market. Under such circumstances, there is no natural market outcome other than aggregation of providers, leading to the establishment of monopoly positions in the Internet provider space. This aggregation is already well underway, and direction of the Internet market will be forged through the tension between this aggregation pressure and various national and international public policy objectives that relate to the Internet industry.

The problem stated here is not in the installation of transmission infrastructure, nor is it in the retailing of Internet services. The problem faced by the Internet industry is in ensuring that each provider of infrastructure is fairly paid when the infrastructure is used. In essence, the problem is how to distribute the revenue gained from the retail sale of Internet access and services to the providers of carriage infrastructure. While explosive growth has effectively masked these problems for the past decade, after market saturation occurs and growth tapers off, these issues of financial settlement between the various Internet industry players will then shape the future of the entire global ISP industry.

[This article is based in part on material in *The ISP Survival Guide,* by Geoff Huston, ISBN 0-471-31499-4, published by JohnWiley & Sons in 1998. Used with permission.]

Annotated Reading List

The following articles and publications address various aspects of Internet interconnection and peering, and the underlying issues of the economics of Internet carriage.

[0] Huston, G., "Interconnection, Peering and Settlements—Part I," *The Internet Protocol Journal,* Volume 2, Number 1, March 1999.

The first part of this article.

[1] Huston, G., *ISP Survival Guide,* ISBN 0-471-31499-4, John Wiley & Sons, November 1998.

A more comprehensive view of the technology, business and strategy behind the Internet service sector.

[2] Halabi, B., *Internet Routing Architectures,* ISBN 1-56205-652-2, Cisco Press, April 1997.

An excellent information resource on how to configure BGP to express policies for interconnecting networks.

[3] Frieden, R., "Without Public Peer: The potential Regulatory and Universal Service Consequences of Internet Balkanization," *Virginia Journal of Law and Technology,* ISSN 1522-1687, Vol. 3, Sept. 1998. **http://vjolt.student.virginia.edu/graphics/vol3/vol3_art8.html.**

A good briefing paper from an economic perspective on interconnection issues, with particular attention to the domestic situation in the United States.

[4] Cukier, K., "Peering and Fearing: ISP Interconnection and Regulatory Issues," Presented paper at the Harvard Information Infrastructure Project Conference on the Impact of the Internet on Communication Policy, December 3–5 1997.

Conference program is at:
http://ksgwww.harvard.edu/iip/iicompol/agenda.html

The Cukier paper is at:
http://ksgwww.harvard.edu/iip/iicompol/Papers/Cukier.html

[5] Shapiro, C., Varian, H., *Information Rules: A Strategic Guide to the Information Economy,* ISBN 087584863X, Harvard Business School Press, November 1998.

A broader look at the Internet from an economic perspective, looking at both content and service provider economics.

[6] Varian, H., "The Information Economy—The Economics of the Internet, Information Goods, Intellectual Property and Related Issues," **http://www.sims.berkeley.edu/resources/infoecon/**

This is a collection of references to other online resources, and is a useful starting point for further reading on this topic.

GEOFF HUSTON holds a B.Sc and a M.Sc from the Australian National University. He has been closely involved with the development of the Internet for the past decade. He was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Technologist in the Internet area for Telstra. He is also an active member of the IETF, and is a member of the Internet Society Board of Trustees. He is author of *The ISP Survival Guide,* and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, a collaboration with Paul Ferguson. Both books are published by John Wiley & Sons.
E-mail: **gih@telstra.net**

Firewalls and Internet Security, the Second Hundred (Internet) Years

by Frederic Avolio,
Avolio Consulting

Interest and knowledge about computer and network security is growing along with the need for it. This interest is, no doubt, due to the continued expansion of the Internet and the increase in the number of businesses that are migrating their sales and information channels to the Internet. The growth in the use of networked computers in business, especially for e-mail, has also fueled this interest. Many people are also presented with the post-mortems of security breaches in high-profile companies in the nightly news and are given the impression that some bastion of defense had failed to prevent some intrusion. One result of these influences is that that many people feel that Internet security and Internet firewalls are synonymous. Although we should know that no single mechanism or method will provide for the entire computer and network security needs of an enterprise, many still put all their network security eggs in one firewall basket.

Computer networks may be vulnerable to many threats along many avenues of attack, including:

- *Social engineering*, wherein someone tries to gain access through social means (pretending to be a legitimate system user or administrator, tricking people into revealing secrets, etc.)
- *War dialing,* wherein someone uses computer software and a modem to search for desktop computers equipped with modems that answer, providing a potential path into a corporate network
- *Denial-of-service attacks,* including all types of attacks intended to overwhelm a computer or a network in such a way that legitimate users of the computer or network cannot use it
- *Protocol-based attacks,* which take advantage of known (or unknown) weaknesses in network services
- *Host attacks,* which attack vulnerabilities in particular computer operating systems or in how the system is set up and administered
- *Password guessing*
- *Eavesdropping* of all sorts, including stealing e-mail messages, files, passwords, and other information over a network connection by listening in on the connection.

Internet firewalls have been around for a hundred years—in Internet time. Firewalls can help protect against some of these attacks, but certainly not all. Firewalls can be very effective at what they do. The people who set up and use them must have the knowledge of how they work, and also be aware of what they can and cannot protect. In this article, we examine the Internet firewall, touch on its history, see how firewalls are used today, and discuss changes that are in place for the next hundred years.

Internet History

In the beginning, there was no Internet. There were no networks. There was no e-mail, and people relied on postal mail or the telephone to communicate. The very busy sent telegrams. Few people used ugly names to refer to others whom they had never met. Of course, the Internet has changed all this. The Internet, which started as the *Advanced Research Projects Agency Network* (ARPANET), was a small, almost closed, community. It was a place, to borrow a line from the theme to *Cheers,* "where everybody knows your name, and they're always glad you came."

On November 2, 1988, something happened that changed the Internet forever. Reporting this incident, Peter Yee at the NASA Ames Research Center sent a note out to the TCP/IP Internet mailing list that reported, "We are currently under attack from an Internet VIRUS! It has hit Berkeley, UC San Diego, Lawrence Livermore, Stanford, and NASA Ames." Of course, this report was the first documentation of what was to be later called *The Morris Worm*. The researchers and contributors that had built the Internet, as well as the organizations that were starting to use it, realized at that moment that the Internet was no longer a closed community of trusted colleagues. In fact, it hadn't been for years. To their credit, the Internet community did not overreact to this situation. Rather, they started sharing information on their practices to prevent future disruptions.

(One of the results of this problem was a growth in the number of Internet mailing lists dedicated to security and bug tracking. The *firewalls* list—subscribe with e-mail to **`Majordomo@lists.gnac.net`**—and the *bugtraqs* list—**`LISTSERV@netspace.org`**—are two examples, as well as the *CERT Coordination Center*—**`http://www.cert.org/`**.)

Other famous, and general, attacks followed:

- Bill Cheswick's "evening with Berferd"[4]
- Clifford Stoll's run-in with German spies[7]
- The massive password capture of the winter of 1994
- The IP spoofing attack that Kevin Mitnick used against Tsutomu Shimomura[6]
- The rash of denial-of-service attacks in January 1996, and the "Web site break-in of the week."

All these viruses have made it into the popular press, and all have raised awareness of the need for good computer and network security. As these, and other, events were unfolding, the firewall was starting its rapid evolution. Although the development of firewall technology and products may be seen as very fast, it sometimes seems that firewalls are just barely keeping up with the new applications and services that spring up and immediately become a "requirement" for many Internet users.

Firewall History

We are used to firewalls in other disciplines, and, in fact, the term did not originate with the Internet. We have firewalls in housing, separating, for example, a garage from a house, or one apartment from another. Firewalls are barriers to fire, meant to slow down its spread until the fire department can put it out. The same is true for firewalls in automobiles, segregating the passenger and engine compartments.

Cheswick and Bellovin, in the definitive text on Internet firewalls[4], said an Internet firewall has the following properties: it is a single point between two or more networks where all traffic must pass (choke point); traffic can be controlled by and may be authenticated through the device, and all traffic is logged. In a talk, Bellovin later stated, "Firewalls are barriers between 'us' and 'them' for arbitrary values of 'them.'"

The first network firewalls appeared in the late 1980s and were routers used to separate a network into smaller LANs. In these scenarios—and using Bellovin's definition, above—"us" might be—well, "us." And "them" might be the English Department. Firewalls like this were put in place to limit problems from one LAN spilling over and affecting the whole network. All this was done so that the English Department could add any applications to its own network, and manage its network in any way that the department wanted. The department was put behind a router so that problems due to errors in network management, or noisy applications, did not spill over to trouble the whole campus network. The first security firewalls were used in the early 1990s. They were IP routers with filtering rules. The first security policy was something like the following: allow anyone "in here" to access "out there." Also, keep anyone (or anything I don't like) "out there" from getting "in here." These firewalls were effective, but limited. It was often very difficult to get the filtering rules right, for example. In some cases, it was difficult to identify all the parts of an application that needed to be restricted. In other cases, people would move around and the rules would have to be changed.

The next security firewalls were more elaborate and more tunable. There were firewalls built on so-called *bastion hosts*. Probably the first commercial firewall of this type, using filters and application gateways (proxies), was from Digital Equipment Corporation, and was based on the DEC corporate firewall. Brian Reid and the engineering team at DEC's Network Systems Lab in Palo Alto originally invented the DEC firewall. The first commercial firewall was configured for and delivered to the first customer, a large East Coast-based chemical company, on June 13, 1991. During the next few months, Marcus Ranum at Digital invented security proxies and rewrote much of the rest of the firewall code. The firewall product was produced and dubbed DEC SEAL (for *Secure External Access Link*). The DEC SEAL was made up of an external system, called *Gatekeeper,* the only system the Internet could talk to, a filtering gateway, called *Gate,* and an internal *Mailhub* (see Figure 1).

In this same time frame, Cheswick and Bellovin at Bell Labs were experimenting with circuit relay-based firewalls. Raptor Eagle came out about six months after DEC SEAL was first delivered, followed by the ANS InterLock.

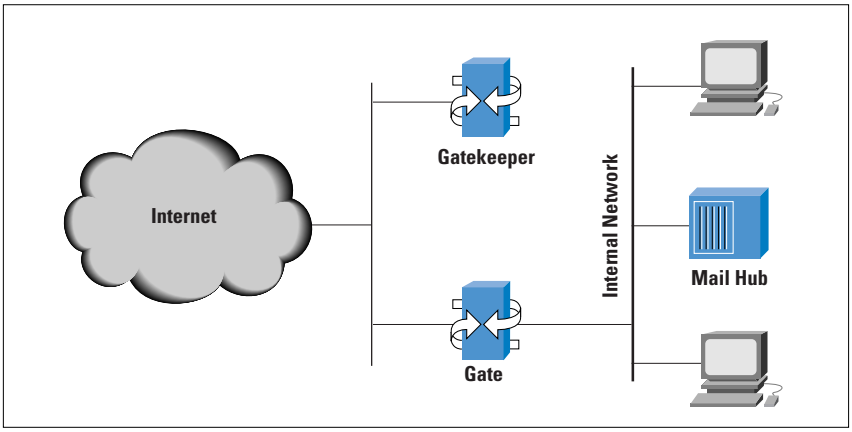


Figure 1: DEC SEAL—First Commercial Firewall

On October 1, 1993, the Trusted Information Systems (TIS) *Firewall Toolkit* (FWTK) was released in source code form to the Internet community. It provided the basis for TIS' commercial firewall product, later named *Gauntlet*. At this writing, the FWTK is still in use by experimenters, as well as government and industry, as a basis for their Internet security. In 1994, Check Point followed with the *Firewall-1* product, introducing "user friendliness" to the world of Internet security. The firewalls before Firewall-1 required editing of ASCII files with ASCII editors. Check Point introduced icons, colors, and a mouse-driven, X11-based configuration and management interface, greatly simplifying firewall installation and administration.

Early firewall requirements were easy to support because they were limited to the Internet services available at that time. The typical organization or business connecting to the Internet needed secure access to remote terminal services (Telnet), file transfer (*File Transfer Protocol* [FTP]), electronic mail (*Simple Mail Transfer Protocol* [SMTP]), and USENET News (the *Network News Transfer Protocol*—NNTP). Today, we add to this list of "requirements" access to the World Wide Web, live news broadcasts, weather information, stock quotes, music on demand, audio and videoconferencing, telephony, database access, file sharing, and the list goes on.

What new vulnerabilities are there in these new "required" services that are daily added to some sites? What are the risks? Too often, the answer is "we don't know."

Types of Firewalls

There are four types of Internet firewalls, or, to be more accurate, three types plus a hybrid. The details of these different types are not discussed here because they are very well covered in the literature.[1, 3, 4, 5]

Packet Filtering

One kind of firewall is a packet filtering firewall. Filtering firewalls screen packets based on addresses and packet options. They operate at the IP packet level and make security decisions (really, "to forward, or not to forward this packet, that is the question") based on the headers of the packets.

The filtering firewall has three subtypes:

- *Static Filtering,* the kind of filtering most routers implement—filter rules that must be manually changed
- *Dynamic Filtering,* in which an outside process changes the filtering rules dynamically, based on router-observed events (for example, one might allow FTP packets in from the outside, if someone on the inside requested an FTP session)
- *Stateful Inspection,* a technology that is similar to dynamic filtering, with the addition of more granular examination of data contained in the IP packet

Dynamic and stateful filtering firewalls keep a dynamic state table to make changes to the filtering rules based on events.

Circuit Gateways

Circuit gateways operate at the network transport layer. Again, connections are authorized based on addresses. Like filtering gateways, they (usually) cannot look at data traffic flowing between one network and another, but they do prevent direct connections between one network and another.

Application Gateways

Application gateways or proxy-based firewalls operate at the application level and can examine information at the application data level. (We can think of this as the *contents* of the packets, though strictly speaking proxies do not operate with packets.) They can make their decisions based on application data, such as commands passed to FTP, or a URL passed to HTTP. It has been said that application gateways "break the client/server model."

Hybrid firewalls, as the name implies, use elements of more than one type of firewall. Hybrid firewalls are not new. The first commercial firewall, DEC SEAL, was a hybrid, using proxies on a bastion host (a fortified machine, labeled "Gatekeeper" in Figure 1), and packet filtering on the gateway machine ("Gate"). Hybrid systems are often created to quickly add new services to an existing firewall. One might add a circuit gateway or packet filtering to an application gateway firewall, because it requires new proxy code to be written for each new service provided. Or one might add strong user authentication to a stateful packet filter by adding proxies for the service or services.

No matter what the base technology, a firewall still basically acts as a controlled gateway between two or more networks through which all traffic must pass. A firewall enforces a security policy and it keeps an audit trail.

What a Firewall Can Do

A firewall intercepts and controls traffic between networks with differing levels of trust. It is part of the network perimeter defense of an organization and should enforce a network security policy. By Cheswick's and Bellovin's definition, it provides an audit trail. A firewall is a good place to support strong user authentication as well as private or confidential communications between firewalls. As pointed out by Chapman and Zwicky[2], firewalls are an excellent place to focus security decisions and to enforce a network security policy. They are able to efficiently log internetwork activity, and limit the exposure of an organization.

The exposure to attack is called the "zone of risk." If an organization is connected to the Internet without a firewall (Figure 2), every host on the private network can directly access any resource on the Internet. Or to put it as a security officer might, every host on the Internet can attack every host on the private network. Reducing the zone of risk is better. An internetwork firewall allows us to limit the zone of risk. As we see in Figure 3, the zone of risk becomes the firewall system itself. Now every host on the Internet can attack the firewall. With this situation, we take Mark Twain's advice to "Put all your eggs in one basket—and watch that basket."

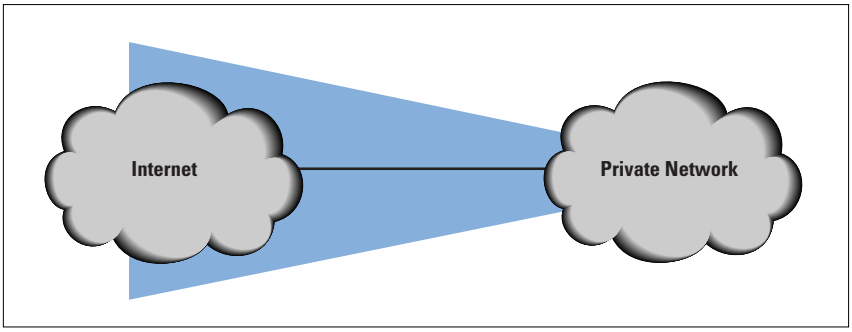


Figure 2: Zone of Risk for an Unprotected Private Network

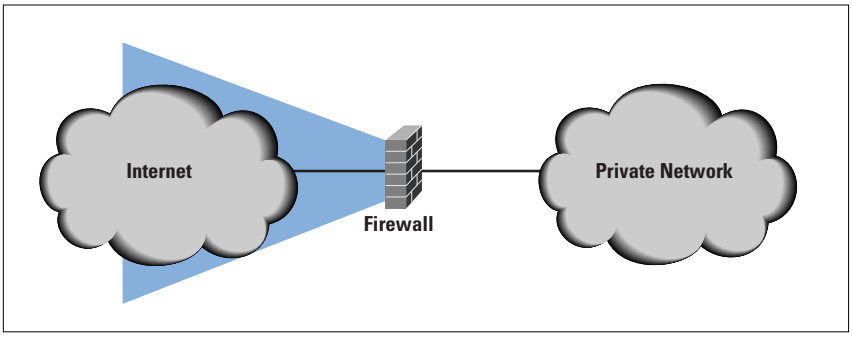


Figure 3: Zone of Risk with a Firewall

What a Firewall Cannot Do

Firewalls are terrible at reading people's minds or detecting packets of data with "bad intent." They often cannot protect against an insider attack (though might log network activity, if an insider uses the Internet gateway in his crime). Firewalls also cannot protect connections that do not go through the firewall. In other words, if someone connects to the Internet through a desktop modem and telephone, all bets are off. Firewalls provide little protection from previously unknown attacks, and typically provide poor protection against computer viruses.

Firewalls Today: Additions

The first add-on to Internet firewalls was strong user authentication. If your security policy allows access to the private network from an outside network, such as the Internet, some kind of user authentication mechanism is required. User authentication simply means "to establish the validity of a claimed identity." A username and password provides user authentication, but not *strong* user authentication. On a nonprivate connection, such as an unencrypted connection over the Internet, a username and password can be copied and replayed. Strong user authentication uses cryptographic means, such as certificates, or uniquely keyed cryptographic calculators. These certificates prevent "replay attacks"—where, for example, a username and password are captured and "replayed" to gain access. Because of where it sits—on both the "trusted" and "untrusted" networks—and because of its function as a controlled gateway, a firewall is a logical place to put this service.

The next add-on to Internet firewalls was firewall-to-firewall encryption, first introduced on the ANS InterLock Firewall. Today, such an encrypted connection is known as a Virtual Private Network, or VPN. It is "private" through the use of cryptography. It is "virtually" private because the private communication flows over a public network—the Internet, for example. Although VPNs were available before firewalls via encrypting modems and routers, they came into common use running on firewalls. Today, most people expect a firewall vendor to offer a VPN option. Firewalls act as the endpoint for VPNs between the enterprise and mobile users or telecommuters, keeping communication confidential from notebook PC, home desktop, or remote office.

In the past two years, it has become popular for firewalls to also act as content screening devices. Some additions to firewalls in this area include virus scanning, URL screening, and key word scanners (also known in U.S. government circles as "guards"). If the security policy of your organization mandates screening for computer viruses—and it should—it makes sense to put such screening at a controlled entry point for computer files, such as the firewall. In fact, standards exist for plugging antivirus software into the data flow of the firewall, to intercept and analyze data files. Likewise, URL screening—firewall controlled access to the World Wide Web—and content screening of files and messages seem like logical additions to a firewall. After all, the data is

flowing through the fingers of the firewall system, so why not examine it and allow the firewall to enforce the security policies of the organization? The downside to this scenario is performance. Also virus scanning must ultimately be performed on each desktop because data may come in to the desktops from paths other than through the firewall—for instance, the floppy.

Recently, some firewall and router vendors have been making the case for a relatively new firewall add-on called "flow control" to deliver Quality of Service (QoS). QoS, for example, can limit the amount of network bandwidth any one user can take up, or limit how much of the network capacity can be used for specific services (such as FTP or the Web). Once again, because the firewall is the gateway, it is the logical place to put a QoS arbitrating mechanism.

Firewalls Tomorrow

In 1997, The Meta Group, and others, predicted that firewalls would be the center of network and internetwork security[7]. After all, firewalls were the first big security item, the first successful Internet security product, and the most visible security device. They quickly became a "must have"—this is good—and a "good enough"—this is not good because firewalls alone are not sufficient. Firewalls became synonymous with security, as mentioned above. The firewall console becoming the network security console seemed natural at that time. But this scenario has not happened, nor will it happen. The reason? The firewall is just another mechanism used to enforce a security policy. This specific enforcement device will not be the policy management device.

As organizations broaden the base of measures and countermeasures used to implement a comprehensive network and computer security policy, firewalls will need to communicate with and interact with other devices. Intrusion detection devices—running on or separate from the firewall—must be able to reconfigure the firewall to meet a new perceived threat (just as dynamic filtering firewalls today "reconfigure" themselves to meet the needs of a user).

Firewalls will have to be able to communicate with network security control systems, reporting conditions and events, allowing the control system to reconfigure sensors and response systems. A firewall could signal an intrusion detection system to adjust its sensitivity, as the firewall is about to allow an authenticated connection from outside the security perimeter. A central monitoring station could watch all this, make changes, react to alarms and other notifications, and make sure that all antivirus software and other content screening devices were functioning and "up to rev." Some products have started down this path already. The *Intrusion Detection System* (IDS) and firewall reconfiguration of network routers based on perceived threat is a reality today. Also, firewall-resident IDS and help-desk software enable another vendor's system to expand from a prevention mechanism into detecting and re-

sponding. The evolution continues and firewalls are changing rapidly to address the next 100 (Internet) years.

In June 1994, the author wrote[5], "Firewalls are a stopgap measure—needed because many services are developed that operate either with poor security or no security at all." This statement is erroneous. Firewalls are *not* a stopgap measure. Firewalls play an important part in a multilevel, multilayer security strategy. Internet security firewalls will not go away, because the problem firewalls address—access control and arbitration of connections in light of a network security policy—will not go away.

As use of the Internet and internetworked computers continues to grow, the use of Internet firewalls will grow. They will no longer be the only security mechanism, but will cooperate with others on the network. Firewalls will morph—as they have—from what we recognize today, just as walls of brick and mortar were eventually replaced by barbed wire, motion sensors, and video cameras—and brick and mortar. But Internet firewalls will continue to be a required part of the methods and mechanisms used to enforce a corporate security policy.

References

[1] Avolio, F. and Ranum, M., "A Network Perimeter with Secure External Access," Proceedings of the ISOC NDSS Symposium, 1996. (**http://www.avolio.com/netsec.html**)

[2] Chapman, D. B. and Zwicky, E., *Building Internet Firewalls,* ISBN 1-56592-124-0, O'Reilly and Associates, 1995.

[3] Cheswick, W. and Bellovin, S., *Firewalls and Internet Security: Repelling the Wily Hacker,* ISBN 0201633574, Addison-Wesley, 1994.

[4] Ranum, M. and Avolio, F., "A Toolkit and Methods for Internet Firewalls," Proceedings of the summer USENIX conference, 1994. (**http://www.avolio.com/fwtk.html**)

[5] Shimomura, T. and Markoff, J., *Takedown: The Pursuit and Capture of Kevin Mitnick, America's Most Wanted Computer Outlaw—By the Man Who Did It,* ISBN 0-7868-89136, Warner Books, 1996.

[6] Stoll, C., *The Cuckoo's Egg: Tracking a Spy through the Maze of Computer Espionage,* ISBN 0671726889, Reprint edition, Pocket Books, 1995.

[7] Meta Global Networking Strategies File 549, November 24, 1997.

FREDERICK M. AVOLIO is an independent security consultant. He has lectured and consulted on Internet gateways and firewalls, security, cryptography, and electronic mail configuration for both government and industry, working in the UNIX and TCP/IP communities since 1979. He is a top-rated speaker and contributor to NetWorld+Interop, USENIX, SANS, TISC, and other security-related forums. With Paul Vixie, Avolio wrote the book *Sendmail: Theory and Practice*, published by Digital Press. He has an undergraduate degree in Computer Science from the University of Dayton and a Master of Science from Indiana University. E-mail: **fred@avolio.com**

Was the Melissa Virus So Different?

by Barbara Y. Fraser, Lawrence R. Rogers, and Linda H. Pesante,
Software Engineering Institute, Carnegie Mellon University

Was the recent electronic mail-based *Melissa* virus so different from similar events in our noncyberspace lives that it merits special behavior? We don't think so. But recent events raise some interesting questions about where to draw the line in our concern about the safety of our mailbox contents.

We regularly receive samples in the mail and don't give them much thought. They run the gamut from laundry detergents to shampoos to cereals to pain relievers. How often do we rip open that sample box of sugar-coated cereal and chomp down a few handfuls as a snack? Do we question whether the labeling accurately reflects the contents of the package? And what about the shampoo samples in those convenient little bottles, just the right size for tossing into our travel bag for the next trip. We use the shampoo with no thought that it might really be hair dye that would turn our hair purple or green. Then there are the sample medications and herbal remedies. Do we use the sample, assuming that it is exactly what it seems to be, without verifying it in some way?

For many of us, these examples represent common behavior today. When we open the samples we find in our mailbox, we don't question whether someone intent on harming us has sent a product that appears to be something we would use and that seems to come from a trusted source. Rarely, if ever, would we call manufacturers and ask whether they had really sent the sample.

How different is this from our approach to the contents of our electronic mailbox? We urge people *never* to click on an attachment before verifying its contents—or at least not until they've verified that it came from the stated sender. Surely we must make these recommendations because of malicious code in electronic mail messages. But we may be asking people to behave differently in cyberspace than they typically do in their noncyberspace life.

What are we to do then? Responsible cyberspace behavior says to trust nothing and verify everything as completely as possible. This scenario would mean that attachments added to an electronic mail messages must be analyzed before being used. To be the most effective, analyzers must be kept up-to-date with the latest information. Even then, rapidly spreading viruses like Melissa can slip under our "radar" for a while. Tools that support authentication and integrity are another building block we should use to gain trust in information that we should otherwise consider untrustworthy.

In our noncomputer lives, how do we know that the medication sample that came in the mail actually came from the attributed vendor? How do we know that the sample was not changed after it left the manufacturing point? The best we can to is to call the manufacturer and exchange some information about the sample: product numbers, packaging color, descriptions of the sample, and so on. Still, we cannot be completely sure that the product is what the packing says it is. Similarly, how do we know that the electronic mail attachment actually came from the stated sender or that it was not changed in transit?

Here cyberspace has the edge over noncyberspace. Technologies are available that help us to verify the mail sender (authentication) and the validity of the message (integrity). Alas, none of the available technologies are multivendor, interoperable, or approved or endorsed by the Internet's standardization body. These technologies are an improvement over their noncyberspace counterparts, but they are not yet mature enough or widespread enough to be as effective as they ultimately will become. Unfortunately, we need that maturity now.

Returning to our original question: Was the Melissa virus so different? Our answer is *no,* it was not so different from the comparable free samples we receive in our noncyberspace lives. Unfortunately, those lives are fraught with the same kind of problems, yet we accept those risks with little concern for our well-being. The real answer is that both our cyberspace and noncyberspace lives need to change to reflect the challenges of our modern world.

About Melissa

The CERT CC began receiving reports of a new virus on Friday, March 26, 1999. The macro virus is activated when a user opens an infected document in Microsoft Word 97 or Word 2000 with macros enabled. The virus is then quickly spread by sending an infected document to the first 50 addresses in the victim's Microsoft Outlook address book. It also infects the **`Normal.dot`** template file, a situation which in turn causes other Word documents created using this template to be infected with the virus. If these newly infected documents are opened by a second user, the document, including the virus, will propogate, sending the docuemnt to 50 addresses in the second user's address book. The CERT CC handled over 300 reported incidents involving Melissa, affecting over 100,000 computers. This estimate is very convervative because it counts only those who contaced the CERT CC. It is believed that millions of host computers were infected.

References

[1] **`http://www.cert.org/advisories/CA-99-04-Melissa-Macro-Virus.html`**

[2] **`http://www.melissavirus.com/`**

[3] **`http://www.nai.com/valert`**

[4] **`http://www.datafellows.com/news/pr/eng/19990327.htm`**

[5] **`http://www.mcafee.com/about/press_releases/pr040299.asp`**

[6] **`http://www.cert.org/other_sources/viruses.html`**

To subscribe to CERT Advisories:
`http://www.cert.org/contact_cert/certmaillist.html`

BARBARA FRASER is a senior member of the technical staff at the Software Engineering Institute (SEI) located at Carnegie Mellon University. She is currently working in the Networked Systems Survivability Program of the SEI and the CERT® Coordination Center. Barbara leads the team that is currently developing an adaptive security management model for networked systems that will allow organizations to adapt to technology and organization changes while maintaining an appropriate level of security in their networked systems. Her professional interests are in developing tools and techniques for improving the survivability of technologies currently deployed in the Internet. Barbara has been involved with the CERT Coordination Center since 1990. She has developed and delivered many talks and courses on Internet security and security incident response, and has worked with many organizations to help them understand and address security issues as they relate to the Internet. Barbara is currently coteaching a graduate course, "The Economics of Information Security," for the Heinz School of Public Policy at Carnegie Mellon University. Barbara is active in the security area of the Internet Engineering Task Force (IETF) and was one of the authors of RFC 1281, "Guidelines for the Secure Operation of the Internet," and RFC 2196, "Site Security Handbook." She is currently a member of the Security Area Directorate and chairs two IETF working groups (GRIP and SSH). Prior to joining the SEI, Barbara was a senior engineer at Martin Marietta Corporation (now Lockheed Martin), where she led a team of software engineers in the development of aircraft simulator software. Barbara holds a bachelor's degree in biology and an M.S. degree in computer science. E-mail: **byf@cert.org**

LAWRENCE R. ROGERS is a senior member of the technical staff in the Networked Systems Survivability Program at the Software Engineering Institute (SEI). The CERT Coordination Center is also a part of this program. Larry's primary focus in this group is analyzing system and network vulnerabilities and helping to transition security technology into production use. His professional interests are in the areas of the administering systems in a secure fashion and software tools and techniques for creating new systems being deployed in the Internet. Before joining the SEI, Larry worked for ten years at Princeton University, first in the Department of Computer Science on the Massive Memory Machine project, and later at the Department of Computing and Information Technology (CIT). While at CIT, Larry directed and managed the UNIX Systems Group that was charged with administering the UNIX computing facilities used for undergraduate education and campus-wide services. Larry coauthored the book Advanced Programmer's Guide to UNIX Systems V with Rebecca Thomas and Jean Yates. Larry received a B.S. degree in Systems Analysis from Miami University in 1976 and an M.A. degree in Computer Engineering in 1978 from Case Western Reserve University. E-mail: **lrr@cert.org**

LINDA HUTZ PESANTE has been a member of the technical staff of the Software Engineering Institute (SEI) since 1987. She is currently the leader of the Information Services Team for the CERT Coordination Center and SEI Networked Systems Survivability Program. She also teaches communication skills in the Master of Software Engineering Program at Carnegie Mellon University. At the University, she is a member of the Institutional Review Board for the Protection of Human Subjects in Research. She holds a B.A. in English and M.A. in professional writing from Carnegie Mellon, and an M. Ed. from the University of Pittsburgh. She has published on the topics of technical communication, network security, and teaching writing in computer science and software engineering programs. E-mail: **lhp@cert.org**

Book Review

OPSF *OSPF: Anatomy of an Internet Routing Protocol,* John T. Moy, Addison Wesley Longman, ISBN 0-201-63472-4, 1998.
http://www.awl.com/cseng/titles/0-201-63472-4

Audience

John Moy takes the somewhat difficult topic of Internet routing and presents an understandable and engaging tour of specific parts of routing and how this one instance interrelates with other parts of Internet routing. This book is not for the routing novice, although the first couple of chapters provide a quick overview and history of routing and one viewpoint on the distinctions between two architectural choices in routing protocol design, *Distance Vector* and *Link State*. This book is really targeted for people that have a basic understanding of what routing is and would like to gain an understanding of this particular tool in the Internet routing "toolbox."

Organization

The second section goes into great detail on one implementation of the Link State architecture, *Open Shortest Path First Protocol* (OSPF). There is a companion volume which contains OSPF specific details and includes source code for building an OSPF service on FreeBSD systems. He covers some background in the design phases of OPSF, delineating why certain choices were made in the evolution of OSPF as we know it today and then starts into what I think of as the heart of the book, an understandable, brief discussion of OSPF design with packet formats. In this section of the book, the author takes a textbook approach and closes each chapter with a series of exercises which test understanding of the principles covered in each chapter. At the end of the section, the FAQ answers a number of questions which operators that are considering OSPF will ask.

The book then changes focus and examines the basics of routing in the context of multicast aware infrastructure. This is an area that is still very dynamic and several of the presumptions that John makes in this section may not be as relevant in today's networking environment. However, he does demonstrate the ability of OSPF to support new features, in this case the variant called *Multicast OSPF* or MOSPF. A discussion of the integration of MOSPF into OSPF networks as well as MOSPF in *Distance Vector Multicast Routing Protocol* (DVMRP) networks points out how different routing protocols can work together. DVMRP forms the central core of the Multicast Backbone or *Mbone*. Both DVMRP and MOSPF lack policy features that many operators demand and so this section remains more of academic interest in understanding how multicast can work.

The fourth section covers configuration and management of OSPF in real networks. Of specific interest to me is the discussion on how OSPF can take advantage of authentication features to ensure the integrity of the routing protocol and the data it sends. Others may find that a discussion of tools for troubleshooting more interesting. A fair amount of the discussion in this section deals with the use of *Simple Network Management Protocol* (SNMP) as the tool for managing and configuring OSPF. Its not clear to me that operators of parts of the Internet are comfortable with this approach since SNMP has known vulnerabilities. Such techniques are useful for monitoring OPSF activities and may be used in private networks with a higher comfort level.

Protocol Review

The book closes with a review of popular routing protocols, both current and historic for unicast and multicast environments. John covers some basic ideas on protocol interactions when systems run more than one but does not cover the interactions between multicast and unicast protocols.

—Bill Manning, USC-ISI
manning@isi.edu

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

Fragments

ICANN Update

As mentioned in previous issues of IPJ, the *Internet Corporation for Assigned Names and Numbers* (ICANN) began operation in early November 1998. Recently, ICANN announced that five companies have been selected to participate in the initial testbed phase of the new competitive *Shared Registry System.* These five participants will be the first to implement the new system for competition in the market for **.com**, **.net**, and **.org** domain name registration services. Currently, registration services for these domains are provided by Network Solutions, Inc. (NSI), which has enjoyed an exclusive right to handle registrations under a 1993 Cooperative Agreement with the U.S. Government. The five registrars participating in the testbed are, in alphabetical order: America Online, CORE (*Internet Council of Registrars*), France Telecom/Oléane, Melbourne IT, and register.com.

Under the Cooperative Agreement between NSI and the U.S. Government, the competitive registrar testbed program began on April 26 and will last until June 24, 1999 (Phase I). Following the conclusion of Phase I, the Shared Registry System for the **.com**, **.net**, and **.org** domains will be opened on equal terms to all accredited registrars, meaning that any company that meets ICANN's standards for accreditation will be able to enter the market as a registrar and offer customers competitive domain name registration services in these domains.

Meanwhile, ICANN continues to work on the formation of several *supporting organizations,* namely the *Domain Name Supporting Organization* (DNSO), the *Address Supporting Organization* (ASO), and the *Protocol Supporting Organization* (PSO). More information is available at: **www.icann.org**

IETF and Related links

The *Internet Engineering Task Force* (IETF) is responsible for the development of standards for Internet technology. Membership to the IETF is open and you can participate in person or subscribe to the IETF mailing list. The IETF meets three times per year. For a list of future meetings and other IETF information see: **http://www.ietf.org**

SIGCOMM

If you want to learn about the latest developments on the research side of networking you should check out SIGCOMM, the Association for Computing Machinery's Special Interest Group on Communications. You can find out more about the group and their annual conference at:

http://www.acm.org/sigcomm/sigcomm99

Send us your comments!

We look forward to hearing your comments and suggestions regarding anything you read in this publication. Send us e-mail at: **ipj@cisco.com**

The Internet Protocol Journal is published quarterly by the Cisco News Publications Group, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

September 1999 Volume 2, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

More and more of the data traffic on the Internet is due to World Wide Web activity. Given the often-complex graphics contents of Web pages, this traffic represents a significant amount of data and leads to an overall requirement for more bandwidth across the system. But building "bigger pipes" is not the only way to achieve better performance. Generally speaking, Web pages are relatively static objects that reside in *one* location and are accessed repeatedly by *many* users, often from "far away." If the contents of the most frequently accessed pages can be stored by a proxy residing more "local" with respect to the end user, significant reductions in download delay can be accomplished. Since the Internet comprises many expensive international circuits, such local mirroring of content is also highly desirable from the point of view of the Internet Service Providers. Storing information in a proxy server is called *caching,* and it is the subject of our first article. Geoff Huston explains the motivation behind—and the different approaches to—caching.

The most popular Local-Area Network (LAN) technology is *Ethernet.* Invented in 1973 by Bob Metcalfe as a 3-Mbps technology, Ethernet has evolved to the now-familiar 10Base-T and 100Base-T standards. Standardized in 1998, *Gigabit Ethernet* is the subject of our second article. Bill Stallings gives an overview of the Gigabit Ethernet standards and their application in enterprise networks. There is already discussion about 10-Gigabit Ethernet and even 100-Gigabit Ethernet. We will keep you posted on these developments.

Some readers have suggested that we publish a few short articles on limited topics. In this issue we bring you the first in what we hope will become a series of articles under the general heading "One Byte at a Time." The article is by Tom Thomas and he discusses *active* and *passive* modes of the File Transfer Protocol (FTP). If you have suggestions for future topics in this series, please contact us at **ipj@cisco.com**

The so-called "Millennium Bug" or "Y2K Problem" has been well reported in all the media. Our *Fragments* section gives some specific information relating to Y2K and the Internet.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Web Caching

by Geoff Huston, Telstra

Web browsing dominates today's Internet. More than two-thirds of the traffic on the Internet today is generated by the Web. In looking at how to improve the quality of service delivered by the Internet, a very productive way to start is examining the performance of Web transactions. It is here that Web caching can play a valuable role in improving service quality for a large range of Internet users.

There are two types of Web caches—a *browser cache* and a *proxy cache*. A browser cache is part of all popular Web browsers. The browser keeps a local copy of all recently displayed pages, and when the user returns to one of these pages, the local copy is reused. By contrast, a proxy cache is a shared network device that can undertake Web transactions on behalf of a client, and, like the browser, the proxy cache stores the content. Subsequent requests for this content, by this or any other client of the cache, will trigger the cache to deliver the locally stored copy of the content, avoiding a repeat of the download from the original content source. In this article we look at proxy caches in further detail, particularly at the aspects of deployment of proxy caches in Internet Service Provider (ISP) networks.

What Is Proxy Web Caching?

When a browser wishes to retrieve a URL, it takes the host name component and translates that name to an IP address. A HTTP session is opened against that address, and the client requests the URL from the server.

When using a proxy cache, not much is altered in the transaction. The client opens a HTTP session with the proxy cache, and directs the URL request to the proxy cache instead (Figure 1).

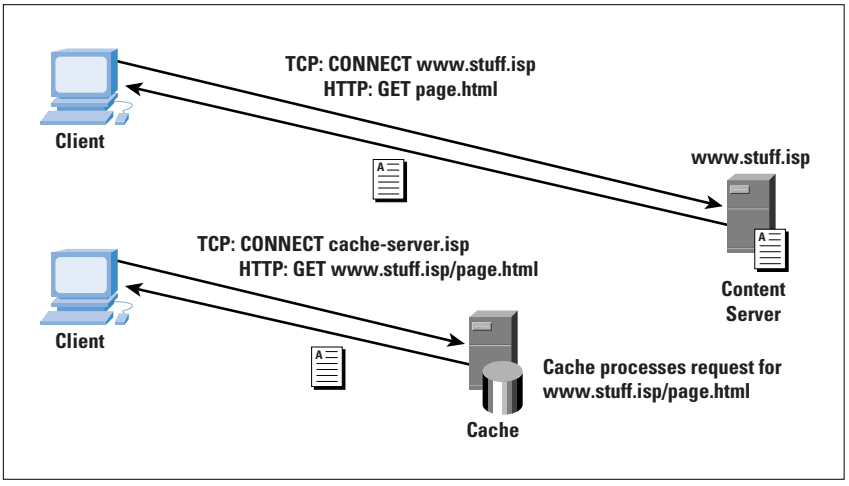


Figure 1: A Proxy Web Transaction

If the cache contains the referenced URL it is checked for freshness by comparing with the "Expires:" date field of the content, if it exists, or by some locally defined freshness factor. Stale objects are revalidated with the server, and if the server revalidates the content, the object is re-marked as fresh. Fresh objects are delivered to the client as a *cache hit*.

If the cache does not have a local copy of the URL, or the object is stale, this is a *cache miss*. In this case the cache acts as an agent for the client, opens its own session to the server named in the URL, and attempts a direct transfer to the cache.

The Pros and Cons of End-to-End Web Access

The original design principle of the Internet architecture is that of the end-to-end model[2, 3]. Within this model the network is a passive instrument that undertakes a best effort to forward packets to the specified destination. Each packet generated by a host is assumed to be forwarded to the addressed destination, and any response to the datagram is assumed to come from that destination address.

The World Wide Web transaction protocol, the *Hypertext Transfer Protocol* (HTTP)[4, 5], is constructed upon this model, where a client's Web fetch causes a TCP session to be opened with the specified target host. The ensuing HTTP conversation identifies the requested data on the destination host, and this data is then passed back to the client. This delivery model is best expressed as a *just-in-time delivery model,* where the data is passed to the client on demand.

This delivery model has many significant advantages. The content server can modify the content, and all subsequent client requests are provided with the updated information, so that updates are immediately reflected in the delivered data. The content server is also able to track all content requests, allowing the content provider to track which particular content is being requested, the identity of each requestor, and how often each content item is referenced. The content provider can also differentiate between various clients, and, using some form of security model, the content provider can authenticate the client and deliver privileged information to certain clients. In this model the content provider can also differentiate between clients, delivering certain information to some clients, and *different* information to other clients of the content server.

Many web systems have been constructed based on the capability of this end-to-end delivery model. Continuously updating Web pages that use either *server push* or *client pull* to regularly update the content on the client's display are used to display stock market prices, weather maps, or network management screens. Client identification can be used to create combined public and virtual private information servers, where a class of identified users can be directed to internal content environments, while other clients are passed to a default public content environment. Such systems form the basis of extranet environments, and can also be used to form part of a virtual private network.

Where information has a defined locality, this tool is very useful. Security and authentication is also used to provide services where the transaction requires some level of privacy. Electronic trading systems, credit card transactions, and related financial systems on the Web make use of such client authentication capabilities. The individual transaction can be encrypted using socket-level encryption,[13] or the entire TCP session can be encrypted using an IP session-level encryption tool such as IP Security (IPSec).

For all these benefits available in an end-to-end model of Web content delivery, there are some balancing drawbacks. A server providing very popular content is placed under considerable stress, both in the number of simultaneous client connections active at any time and in the total volume of data being delivered from the server in the surrounding network. This load is expressed both as a server system load, and as load on the surrounding network. Improving the performance of such systems may entail improving the server throughout, increasing the number of servers through the use of server farms and a traffic manager, and improving the capacity of the local network to deliver the increased volume. However, all these measures may not address all the problems in maintaining quality of the content delivery. Modem-based client systems, and low-bandwidth wireless-based client systems are constrained by a combination of the restricted bandwidth of this last hop and the associated imposed end-to-end delay in conversing with the server. Improving the capacity of the server may not necessarily reduce the number of simultaneously active client connections. Reducing the delay between the client and the point of delivery of the content will improve the performance of content delivery.

In addition, the network itself may not be efficiently utilized. Web traffic does have considerable levels of duplication, where a set of clients request copies of the same content, and the network carries duplicates of the data to each client. For a network provider, where transmission capacity is a business cost, importing the content just once, and then passing local copies of this content to each client, is one method of improving the carriage efficiency of the network.

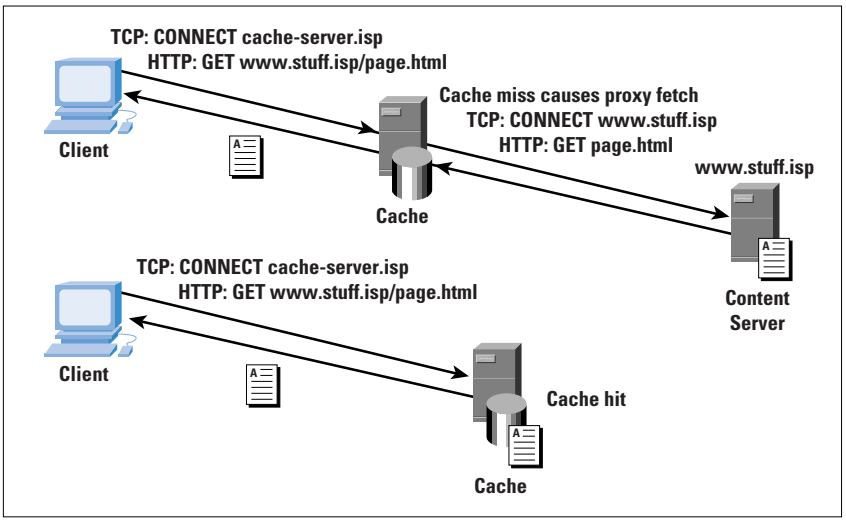
In terms of the ability to improve the service performance of delivery of content to a global network of clients, and in terms of the ability to improve the carriage efficiency of the network, caching of content makes some sense to the content provider, to the ISP, and to the end client.

The Pros and Cons of Web Proxy Caching

The same benefits of improved performance and reduced outbound traffic loads can be realized for World Wide Web traffic through the deployment of Web caches. Web caches are basically no different from any other form of caching. The client request is passed through a *cache agent,* which makes the request to the original source as a proxy for the client. The response of the server is retained in a local cache, and a copy

is passed to the client. If the same request is passed to the cache agent soon after the original request was serviced, the response can be generated from the cache without further reference to the original source. The operation of a Web cache is shown in Figure 2.

Figure 2: A Web Cache



Measurements of ISP traffic profiles indicate that some 70 percent of a typical ISP's traffic is Web-based traffic. An analysis of Web requests indicates that the typical level of similarity of requests (for the same object as one previously requested) can be as high as 50 percent of all Web-based traffic.

There are two hit-rate measures, a *page hit rate* and a *byte hit rate*. A page hit rate measures the proportion of individual HTTP requests that can be served from the cache, irrespective of the size of the page. A byte hit rate measures the ratio of the number of bytes delivered from the cache in hits against the number of bytes in misses. Experience to date has indicated that page hit rates of somewhere between 40 to 55 percent are achievable for a well-configured cache. In such circumstances the associated byte hit rate is between 20 and 35 percent. The major contributor to the hit rate is in image files.

For many ISPs, particularly those operating outside of North America, transmission costs dominate the cost profile of the ISP's operation. If the cache performed at even 60 percent of a theoretical maximum caching performance, the ISP could reduce its external traffic volume requirements by some 13 percent. When the costs of caching are compared to the costs of transmission, this difference can be a significant one in the cost base of the ISP's operation.

For example, if the average cost of transmission is $150 per gigabyte, and the ISP has a typical carriage profile of purchasing 1000 gigabytes per month from an upstream ISP with a 70-percent Web traffic profile, then a cache operating at a 25-percent byte hit rate can save the ISP a recurrent expenditure of $26,250 per month. If the cache costs $100,000

as a capital expenditure and $2000 per month in operational costs to support the service, then a business case analysis would see the cache activity return some $18,000 per month to the business, net of annualized capital and operational expenditures.

The other benefit is to the client, where the reduced network delay between the client and the local cache results in an increase in speed of Web page delivery for cached content.

The average size of a Web transaction is some 16 data packets within the TCP flow. Within a TCP slow-start flow-control process, the first cycle will transmit one packet and wait for an ACK. The reception of the ACK will trigger transmission of two more packets in the second round-trip cycle, and then the sender will await two ACKs. Reception of these two ACKs will trigger a further four packets in the third cycle and eight in the next cycle, and the remaining single packet in the fifth cycle. Therefore, allowing for optimal behaviour of the TCP slow-start algorithm, this average Web transaction takes some five round-trip times. If a user is located some distance away from the Web page, and the round-trip time to the source is 300 ms, the propagation delay of the page load will be 1.5 seconds. In comparison, if the round-trip time to the local Web cache is 2 ms, then the propagation delay of the page load will be 10 ms. These latency figures assume an uncongested network in both cases. In this case, as long as the Web cache search can complete within 1 second, the cache will appear to be far faster to the user.

A slightly different analysis is possible when comparing the performance of a cache configured at the headend of a cable-IP system versus the performance of direct access. The difference in latency in this case is due to both the closer positioning of the cache to the user and the greatly increased effective bandwidth from the cache to the user. A cache download can operate at speeds of megabits per second, as compared to kilobits or tens of kilobits per second when using dialup modem or ISDN services. For a 100K image download, the dial user may experience a 60-second delay, and the same delivery from a local cache via cable-IP may take less than half a second.

The trade-off with caching is that of balancing the the cost of carriage capacity, both in terms of monetary cost of the carriage and the performance cost of the transaction time of the application, against the cost of the use of caching. For non-North American ISPs, in which there is typically a large cache hit rate against North American server locations, the benefits of widespread use of caching are quite substantial. For cable-IP operators, the benefits of local cache operation lie in the ability to exploit the benefits of the very-high-speed final hop from the headend to the end user. For other ISPs, the benefits of caching may be less dramatic, but nevertheless, there are tangible positive outcomes of caching in terms of performance and cost that can be exploited.

As with direct-access models, this approach also has drawbacks. We have already noted the various ways in which the end-to-end model of Web content delivery has been exploited to provide time-based content, client-based content, and secure delivery of content. Caches insert themselves within the end-to-end semantics of the original transaction model, and intercept the transaction by presenting a proxy of the original endpoint. The content delivered from the cache is the content based on the time the cache undertook its request to the server, and the content delivered from the server is based on the server's view of the identity of the cache, rather than the identity of the end client.

With cached content in operation, the cached-content server no longer has an accurate picture of the number of times an item of content is viewed, and by whom. The server cannot authenticate the client, nor can the server deliver any information that is based on the supposed identity of the client. Equally, the client has potential problems, because the client may not be aware that the content has been delivered by the proxy cache. The content may not properly reflect the client's identity, and the information may be based on the security trust model of the server to the cache, rather than the server to the end client, and again the client may not be aware of such a change in security domains. If the content is time-dependent, the content will reflect the time at which the cache retrieved the content, rather than the time the client made the request.

All of this tends to suggest that caching is not a universally applicable tool. Part of the challenge in deploying cache servers is to understand the models of cache deployment and Web content delivery, and ensure that the cache does not intrude in ways that distort the integrity of content delivered to the end user.

Web Cache Hits Versus Web Server Hits

One of the biggest tensions is the balance between the cache operator's desire to maximize the hit rate of the cache system and the desire of many Web page publishers to maintain an accurate count on the number of hits of the page and from where those hits occur. In most cases, it is the requests that are of interest here, rather than the control of delivery of the content. The Web publisher is not necessarily interested in absorbing the hits for Web content. Indeed, many Web publishers see value in distributing the load of content delivery of fixed-content material further out toward the client base, rather than the Web publisher bearing the cost of the distribution load from the local site.

Static pages, composed of plain text and images, are readily cached. As a consequence, the original page publisher may not obtain an accurate count of the number of times the page was displayed by users if the Web server's log was analysed. Some Web page designers place information in the Web page directives; this information directs the Web cache server not to reuse a cached page. The most common way of doing this is to set the "Expires:" Web page information header to the current date and

time, so the next time the page is referenced, a new fetch will be undertaken. One of the more common hacks to cache servers to attempt to improve the hit rate is to allow this directive to be ignored.

This server hit-count problem has plagued cache deployment for many years now. Although there are real requirements in the areas of authentication and security, time-based content, and client-based content that mandate certain types of content being flagged as non-cacheable, much of the data that is marked as non-cacheable has been marked in this way simply for the server to capture the identity of the client. Such "cache-busting" practices are unnecessarily wasteful of network resources, and can overload the content server. There is an Internet Proposed Standard extension to HTTP[6] intended to provide a "Meter" header, where a cache can communicate demographic information relating to client "hits" back to the original content server. The extension also proposes usage limiting, where a server can provide content with a limit on the number of times the information can be used by the proxy cache before revalidating the content with the server.

Web-Caching Models

There are many models of how to invoke a proxy cache.

Explicit Caching

Some proxy cache systems are deployed as a user-invoked option, in which the user nominates a cache server to the browser as a proxy agent, and the browser then directs all Web requests to the proxy cache. At any stage, the user can instruct the browser to turn off the use of the proxy cache, and request the browser to undertake the transaction directly with the client. Modern browsers when configured with a proxy cache may also use the approach of attempting direct access when a request via a proxy cache results in a fetch error. In the proxy cache mode of operation, the destination address of the underlying transport session is then the address of the cache server, while the HTTP content of the transaction remains unaltered. Such caches can be deployed within a client's local network, with the intent of minimizing the amount of traffic passed to the external provider ISP. Additionally, The ISP can operate such a voluntary cache for use by its clients. If the ISP operates in this mode, the benefits to the user in using the cache need to be clearly stated and understood by both the client and the ISP, and the client must be made aware of the location of the cache in configuring his or her local browser.

Forced Explicit Caching

Some ISPs, notably in the dialup service provider sector, operate in a highly cost-competitive market. In such a market service performance and service price are critical business factors, and the provider may choose to operate its network in a forced-cache mode. Here, all Web traffific on TCP port 80 (the port used by the HTTP Web transport protocol) is blocked from direct outbound access, and the ISP's clients are forced to configure their browsers to use the provider's cache for external Web access. This technique is commonly termed *forced caching*.

Transparent Caching

The use of a cache for all Web traffic also can be undertaken by the ISP, without the explicit configuration of the identity of the proxy cache into the user's browser. Irrespective of precisely how this setup is engineered, and there are numerous ways of engineering it, this technique is termed *transparent caching*. With transparent caching the user, and the user's browser, may not be explicitly aware that caching is being undertaken when processing the user's requests. Here the network has to intercept HTTP packets destined to remote Web servers, and present these packets to the proxy cache. Once the page is located, either as a cache hit or a cache miss, the cache must then respond to the original requestor by assuming the identity of the original destination (Figure 3).

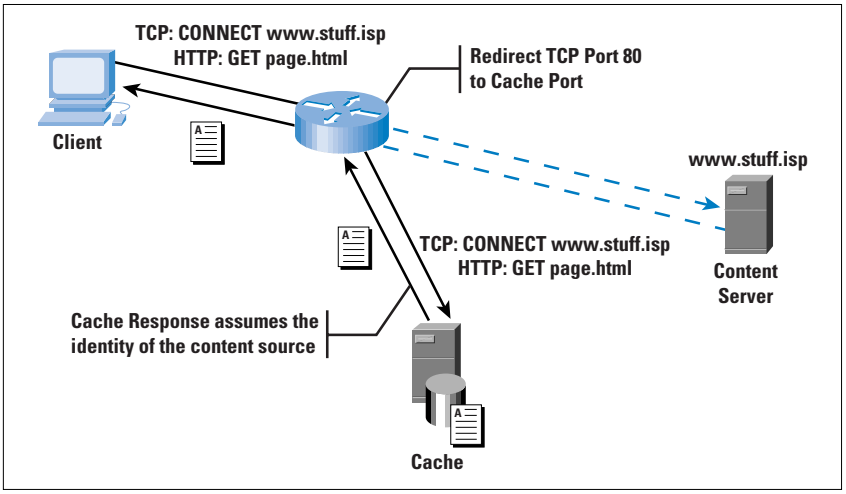


Figure 3: Transparent Caching

It should be noted that no mechanism to date of explicit or transparent caching is completely transparent to both the Web client and the Web server. Where the Web server uses an end-to-end security access model the transparent cache may fail, because the cache will present its address as the source of the request, rather than that of a client. This scenario may result in a page-denied error to the cache request, whereas the client could have completed the transaction directly with the server. In those situations where the use of the cache is mandated, either through filters and a forcing function, or through transparent network redirection, there is no user-visible workaround to the error, and the level of user frustration with the entire cache service rises dramatically.

Under some circumstances it may be possible to work around transparent cache fetch errors. One approach is for a cache fetch error to trigger the cache subsystem to establish an HTTP session with the content server using the source address of the client, and then pass the original HTTP GET request to the server. The server's response is then passed to the client using a TCP bridge. (A TCP bridge is where the connecting device is required to translate the sequence numbers of the TCP headers between the two TCP sessions). Having the cache subsystem intercept the server's packets addressed to the client does require careful coordination with the cache router, and TCP bridging is also quite complex in its

operation, so such solutions tend to be somewhat unstable under load stress. An alternative approach is for the cache to pass a TCP RST back to the client, and instruct the cache router to insert a temporary entry in its redirection filter so that any subsequent TCP port 80 connection from the client to the server's address is not redirected to the cache.

If the sole benefit to the client is improved speed of response, then the ISP must understand that the performance of the Web cache systems must be continually tuned to be highly responsive to Web requests under all load conditions experienced by the ISP. Performance of cache hits must be maintained at a level consistently faster than the alternative of direct client access to the original client site. Performance of cache misses must be at a level that is not visibly slower than that of direct access to the original site. If the user's perception of performance of the cache drops, the benefit to the user also drops. In the case of user-selected caching, the users will turn off the cache option in their browser and return to a mode of direct access.

The business model of a cache is that the capital and operational costs associated with localizing traffic to the cache result in cost reductions to the ISP, when compared to the operation of a noncached network. These cost reductions can be passed on to all users through operation of the entire service at a lower price point or selectively passed on to those clients who make use of the cache through some form of cache-use tariff. The generic model of applying the cost reduction to the ISP's service tariff is certainly an advantage in a price-competitive marketplace. However, unless the performance of the cache is consistently very high, and the transparency of the cache is close to perfect, each individual user may attempt to use direct-access methods.

The alternate business model is to pass on the marginal cost savings to those clients who make use of the cache, and at a level that corresponds to the client's use of the cache and its effectiveness in operating at a high cache hit rate. If, for example, the ISP uses a charging model that includes a tariff component based on the amount of data delivered to the client during the accounting period, this tariff component could be adjusted by the amount of use the client made of the cache system and the relative operating efficiency of the cache in generating cache hits.

As an example, if traffic is tariffed at $100 per gigabyte as delivered to the customer, a discounted value can be derived for traffic delivered from the Web cache. If the average cache byte hit rate is 30 percent, then after factoring in the costs of capital equipment and operational support, the traffic from the cache could be tariffed at $80 per gigabyte. Here, the benefit of using the Web cache is passed directly to those clients who make use of the cache, who both enjoy lower tariffs in direct proportion to their use of the cache and derive superior performance through using the cache. The accounting for this marketing model is certainly a more involved process, involving additional accounting systems and processing to undertake an accurate per-client view of cache usage.

It is becoming increasingly evident that a robust business model associated with a model of discretionary use of a Web cache is that of access to a lower unit price of traffic. In this way, the user sees the incentive of immediate financial benefit in choosing to use the cache system. When the provider deploys transparent or forced caching, translating the benefits of caching into an overall reduced tariff structure for all clients is a more robust business model.

Web-Cache Systems

Cache systems can take a variety of forms. The original Web server from CERN, the original location of the development of Web software, allowed a mode of proxy behaviour. This cache server model was developed significantly in the Harvest Project, a research project at the University of Colorado. As an evolutionary path, the *Harvest* cache server is being further developed within the scope of the development of the *Squid* cache server software and the associated *Internet Caching Protocol* (ICP).

Currently numerous freely available proxy cache systems are available, such as Squid, and many systems are available commercially, such as the Cisco Systems *Cache Engine*. Some of these systems are software packages that operate on a conventional operating system platform, while some use a customized platform kernel, which is optimized for the demands of a cache-delivery environment.

Many of the characteristics of Web caching systems are relevant to the performance of the caching environment. The first is the *size* of the cache server. The relationship between the size of the cache and its hit rate is not a linear relationship. For typical patterns of Web use generated from a relatively large user population, a cache of 1 gigabyte or so will yield reasonable hit rates. Further increase of the cache size will yield incremental improvements in the cache hit rate, where the incremental rate is best described by a negative exponential relationship. Thus, caching systems with 10 gigabytes of storage do not produce performance characteristics markedly different from larger 100-gigabyte caching systems. No objective best size of cache system can be determined, because local environments differ, but every environment exhibits the law of diminishing returns, in which the addition of further cache capacity yields no tangible difference in the cache effectiveness. Large caches take some time, in the order of days or even weeks, to build up a sufficiently large repository of cached data to produce an improved cache hit rate. Generally, 10- to 100-gigabyte cache systems provide extremely effective cache performance, as long as the cache is allowed to stabilize for some weeks following startup. Memory demands in a cache also need to be carefully configured. The URL index of the storage system is stored in memory in most cache architectures in order to perform fast cache lookups, so that the more disk storage configured, the larger the memory requirements.

The next parameter is the *number of simultaneous cache requests* that the cache server can manage efficiently. Note that this metric is different from the number of requests per second that the cache server can manage. The number of simultaneous sessions that the cache server can support is related to the amount of resources allocated to the cache request and the total resource capacity of the box.

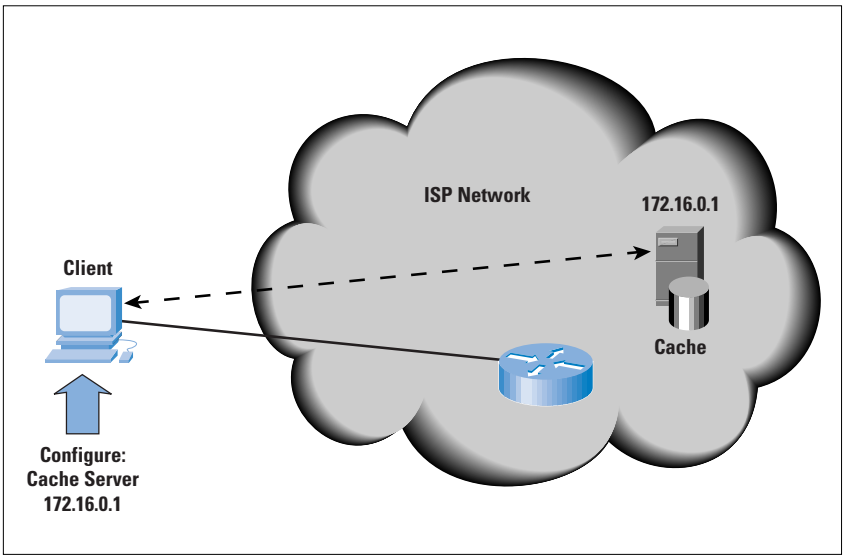
The environment of deployment is very relevant to the performance of the cache environment. The related metric to the number of simultaneous requests that can be managed is the average time to process a request. Combining these two metrics provides the number of requests per second that the cache system can process. The same unit will have a different performance metric of requests per second when deployed in different parts of the Internet. If the cache system is deployed with a satellite-based feed, then the average time to process a cache miss is considerably longer because of the higher latency of the satellite path. This scenario leaves the process of managing the original request open for a longer period, blocking other requests from using this process slot. If the same unit is deployed in a location where cache misses take fractions of a second to process, the process slot can be quickly reused. Each active client connection also consumes memory, and the client connection will remain open for as long as it takes to complete the Web transaction, either for a hit or a miss. The greater the mean round-trip delay for a miss, the greater the number of concurrent active sessions held in the cache. Similarly, the greater the number of low-speed modem or wireless-based clients, the greater the number of concurrent active sessions in the cache. Whether the client operates in transparent mode or in explicit proxy caching mode is also an important consideration. Browser clients use an explicit proxy cache with a persistent connection, while if the cache is a transparent cache, the cache will see clients bring up and drop HTTP connections each time the base URL changes. This session reestablishment, together with the additional Domain Name System (DNS) resolution load imposed on the client, can add up to half a second to the transparent cache response time as compared to the explicit cache response.

Web Cache Deployment Models

In this section we first examine scaling issues for explicitly referenced cache configurations, and then look at the changes to the model introduced through transparent caching.

The simplest deployment model of an explicit cache is that of deployment of a single cache system as a browser-selectable resource. This system can be deployed within an ISP's server environment with a TCP port-80 interface opened for client access. Such a deployment model is shown in Figure 4.

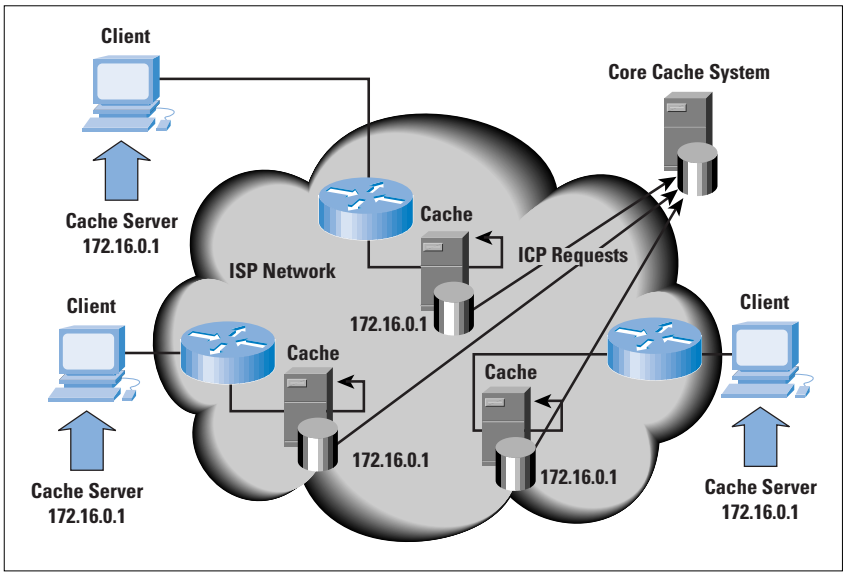
Figure 4: A Selectable Web Cache



Single Web proxy cache systems can be placed under some significant load, and an overloaded and poorly performing cache is perhaps worse than no cache at all. However, scaling this deployment model can prove challenging. Where an ISP operates multiple access points, or points of presence (POPs), one scaling solution is to deploy a server at each POP and use the same IP address for each server. This solution allows the ISP to provide a consistent configuration to all clients and to augment capacity at any location seamlessly. If the cache itself is responsible for advertising the common IP address into the routing system, the caches can also act in a mutual backup role. Failure of a single server will shut down the local route advertisement. Traffic directed to this address will then be carried by the routing system to the next closest proxy cache. There may be some level of TCP session resets for sessions that were active on the failed unit, but in all other respects the switchover is seamless to the client base, and the recovery of an operational state among a set of such servers can be left to the routing system. This deployment model is indicated in Figure 5. Such servers can be configured as a set of local satellite systems to a larger caching core, using an *Internet Cache Protocol* (ICP) configuration to set up a caching hierarchy.

ICP is a lightweight message format for communicating between Web caches[7]. The message format is a simple two- packet exchange, where a Web cache passes a URL query to another cache. The response is either a hit or a miss, indicating the presence of the URL object on the remote cache. On top of this protocol can be constructed cache hierarchies, to allow multiple neighboring caches to pool their resources effectively.

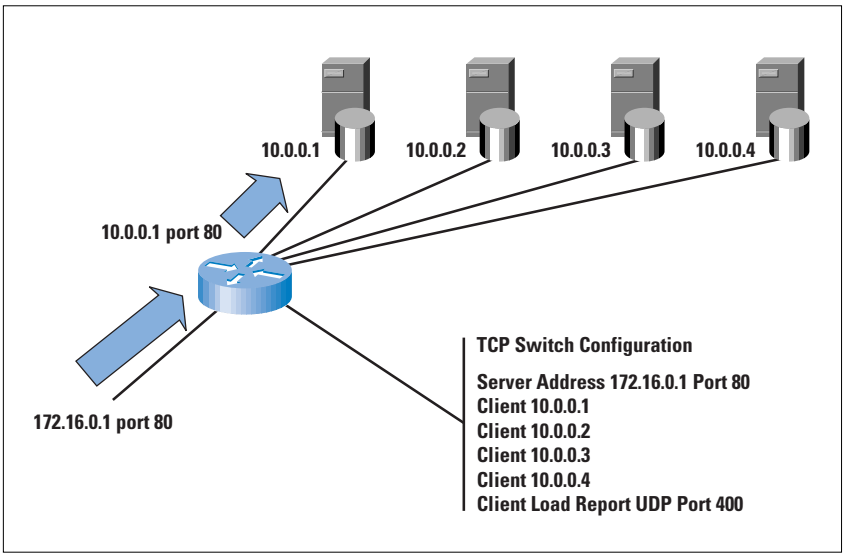
Figure 5: Replicated Web Caches



The proposed mode of configuration of caches is into a tree-structured hierarchy[8]. In such a hierarchy every participating cache is organized with a connection of neighboring peers and an ICP parent. When a cache request cannot be serviced from the local cache, the cache first uses a set of local configuration rules to determine if the server is local. If so, the cache queries the server directly for the content. If the server is remote, the cache issues a set of simultaneous ICP queries to all its cache peers. If any peer responds with an ICP hit, the cache then requests the peer to provide the referenced content. If all peers respond negatively to the ICP query, or a two-second timeout elapses, the cache then requests the URL from its designated parent. The parent may use a peer referral, or the parent may refer the query to its parent, or perform a cache retrieval on behalf of the original request. The intent of this mode of operation is to use a lightweight query response protocol to allow a local collection of caches to pool their cached data. ICP has also been used with additional policy constraints, although the protocol itself is not capable of describing or carrying overly complex retrieval policies. Other intercache protocols are available, including the *Hyper Text Caching Protocol* (HTCP) and the *Cache Array Routing Protocol* (CARP), which offer functionality in terms of intercache cooperation similar to that of ICP[9].

Another scaling measure is to alter the single server to multiple servers, using a TCP-based, load-sharing mechanism in the switching system to ensure that the servers are evenly loaded. This setup is shown in Figure 6. Such a simple load-sharing system may even the load on each server, but it will cause each server to act independently of its sibling servers. It is essential in such an environment to use ICP to coordinate the servers so that they will refer to each other before initiating a new fetch from the content server.

Figure 6: Load-Balancing Web Caches



In such a configuration each cache will contain content also held in neighboring caches. Although this scenario may allow some form of server load balancing, particularly when the servers continually communicate their current load conditions to the load-balancing switch, there is still some inefficiency in the cache farm operation through the potential replication of content on each of the component caches. One direction of scaling the cache servers is to take a collection of cache servers and allow each cache server to specialize in the content it holds. However, the outer TCP destination address does not help the server determine which URL is being requested. In an explicit cache configuration, the browser is directing the TCP session to the externally advertised TCP address of the server farm. The URL information is embedded within the HTTP payload. Some developments have been made in this area, where, with a combination of TCP spoofing and TCP session bridging, a server switch can select the appropriate cache for each HTTP-referenced URL, and then logically connect the client's TCP session to a TCP session to the selected cache to deliver the URL to the client.

Transparent caching presents some further deployment challenges. The functional requirement is to pass all Web requests through a proxy cache server without the explicit knowledge of the client. Two generic techniques exist to achieve this goal:

- *Inline caches:* The first of these approaches is to pass all traffic through a two-port cache server. All non-HTTP traffic is simply passed straight through the device without alteration. HTTP traffic is intercepted and passed to a cache module. The major concern with this approach is the introduction of a single point of failure with an active network element. Any failure of the cache may well prevent all further traffic from entering or leaving the served subnetwork.

- *Redirection caches:* A technique that does not place the cache as a critical point of potential failure is to use policy redirection within the router, redirecting all port-80 traffic to the attached cache. Normally such a policy redirect would infer that the cache is located one hop away from the router, so that such a redirection is normally a local solution. Redirection to a tunnelled interface does allow some greater flexibility in this setup, and the one cache farm could, in such an approach, service a collection of redirecting routers. The failure mode of this form of operation remains a concern, because the redirection mechanism in the router would not normally be aware of the operational status of the cache.

Transparent caches need to ensure that the full URL is inserted into the HTTP level request. When the browser assumes that the request is directed to the content server, the GET request may specify a URL relative to the server. In such cases, the transparent server will need to perform a DNS lookup of the destination IP address of the TCP session in order to reconstruct the complete URL.

Although the DNS lookup does have some performance implications to transparent caches, the major issue for transparent caches is to devise a fail-safe mechanism, so that if the cache server fails for any reason, the caching redirection is disabled. One solution is to use a redirection function within the router in conjunction with a keepalive-based Web cache management protocol. This scenario is the basis of the *Web Cache Coordination Protocol* (WCCP)[10]. WCCP also adds the ability to load share across multiple cache servers through content distribution. Transparent caching assists in this task because the destination address in the IP packets can be used as the basis of the cache selector. The keepalive exchange between the router and the cache server system allows the router to cease redirecting Web traffic upon failure of the servers.

Alternative solutions rely on the cache itself participating in a local routing environment. The redirecting router uses policy-based redirection to forward all port-80 traffic to an address announced by the cache system at a high routing priority. The same address is also announced by the default path router at a low routing priority. Failure of the cache system will result in a withdrawal of the high-priority route, and while the redirection will remain in place on the router, the redirection will be in the direction of the default route.

Another challenge is to process cache misses at a speed comparable with normal noncached Web retrieval. A process of pulling the document into the cache and then serving the document to the original requestor does not meet that objective. The transparent cache has to feed the document to the requestor while simultaneously creating a stored copy for subsequent cache serving.

However, the largest challenge to the transparent cache is that it can serve only documents that are not dependent on the identity of the requestor being preserved. Web servers that use an end-to-end model of access, based on source address identification, or Web servers that attempt to present different documents to the client based on the client's source address, do not fit within the transparent caching model. There is much interest in solutions that allow a transparent cache to effectively shut down in the case of a Web retrieval error, and allow the original requestor the ability to conduct a HTTP conversation directly with the server in such situations. Although there is interest in a network-only solution, it appears at this stage that some level of assistance from the browser may be required. One model of operation is that a transparent cache records the network-level flow identification of a failed Web retrieval, and passes a retry signal back to the requesting browser, and also passes this flow identifier back to the redirector as a temporary filter entry. When the requestor retries the query, as per the signal from the cache, the redirecting router will refrain from redirecting the flow to the cache, and allow an end-to-end session to operate.

Accounting for Web Cache Use

These deployment systems allow for user-optional cache configuration. If the ISP wants to account for the use of the cache, then the cache server or the switch that feeds the cache server must play an active role in accounting collection.

If every network address is uniquely advertised to the ISP by a particular client, then the task of accounting for cache use can be performed using the logged records of the cache system itself. Because every IP address can be uniquely mapped to an ISP client, it is possible to also associate the volume of bytes delivered by the cache to the identified client.

Unfortunately, two factors make this supposition of address uniqueness somewhat weak. First, dialup address assignment implies that the association of an IP address to a client is held only within dialup accounting records in the first instance, and the binding is valid only between the times referenced in the start and stop records. This scenario can be configured into an accounting model by simultaneously processing the dial accounting records when attempting to associate a particular IP address at a particular time to a client.

The second factor is slightly more challenging. For an ISP that offers permanent access transit services, the potential exists that any particular IP address may not be uniquely routed. Normally, such multiple access environments are part of a Border Gateway Protocol (BGP)-based interaction with multiple clients. Knowing the IP address of the query agent is not enough. Ascertaining the next-hop Autonomous System (AS) number as well as the IP address is now necessary to determine the client using the cache.

The implication is that the accounting records now need to be generated on the router that is also the entry point to the cache. In addition, the router must participate in the interior BGP (iBGP) core mesh to maintain current AS path-selection choices. Given the considerable overheads that such an engineering design entails, an alternative approach is to restrict the cache accounting role to account for those cases where the cache client is readily identified. A common measure is that the lower tariff is available only to customers who are "singly homed" with the ISP. Not only is this a strong market incentive for customer loyalty, it also allows simple engineering solutions for cache accounting, because the lookup from the IP address in the cache log to a customer account is then relatively straightforward. Such measures allow a cache-use tariff to be very competitively positioned in the market.

As well as accounting issues, another component for the consideration of optional use of Web caches is that of the necessity of restricting the use of the cache to clients of the ISP. The motive for so doing is to ensure that the cache is available only to clients of the service and not to clients of peer ISPs. It may not be an issue worth the effort of solving, and the first questions ISPs should ask is, "To what extent does this happen, and what impact does it have on the operation of the Web cache systems?" In most cases, the accounting of cache usage may reveal that this issue is one of negligible proportions, and any effort expended in devising an engineering solution would far outweigh the loss to the ISP through such use of the service.

If the measurement of such usage is considered sufficient to warrant engineering solutions, then the mechanisms available to the ISP are to ensure that the Web cache access is filtered at the edges of the ISP network and to ensure that access is possible only by ISP clients, or that the address of the cache is not exported in the routing system to peer ISPs or upstream ISPs.

Further Deployment Challenges

It is highly likely that further development will occur with cache servers in the near future. Large-scale backbone IP networks that use OC-3c (155 Mbps) or OC-12c (622 Mbps) transport cores may carry tens of thousands of requests per second. Designing transparent caches that fit within a transport core at such a scale does present dramatic scaling issues in terms of cache system performance. This factor continues to elude many of today's products available on the market. The generic architecture today is to use a cache network that attempts to place the cache systems closer to the access edge of the network, where the Web request volumes are within the scale of today's cache systems.

Transparency of the cache remains an issue, and it is perhaps an area of further refinement within the specification of the underlying HTTP Web server protocol, as well as further refinement of the operation of Web browsers and transparent cache systems. A potential implementation within Web browsers may allow the user to state the acceptability of using a cache to complete a request, and allow noncache Web page retrieval attempts on cache failure, in the same way that the provider can use page expiration directives to direct a cache not to store the presented data.

References and Further Reading

[1] Huston, G. *ISP Survival Guide,* ISBN 0-471-31499-4, John Wiley & Sons, November 1998.
A more comprehensive view of the technology, business, and strategy behind the Internet service sector.

[2] Clark, D.D. "The Design Philosophy of the DARPA Internet Protocols," Proceedings of SIGCOMM 88, ACM *Computer Communications Review* (CCR), Volume 18, Number 4, August 1988, pp. 106–114 (reprinted in ACM CCR Volume 25, Number 1, January 1995, pp. 102–111).
The original paper describing the end-to-end design philosophy used within the Internet protocols.

[3] Carpenter, B., Ed. "Architectural Principles of the Internet," RFC 1958, Informational RFC, June 1996.
A summary of the design principles underlying the current Internet architecture.

[4] Berners-Lee, T., et al. "Hypertext Transfer Protocol—HTTP/1.0," RFC 1945, Informational RFC, May 1996.
The specification of Version 1.0 of the HTTP protocol.

[5] Fielding, R., et al. "Hypertext Transfer Protocol—HTTP/1.1," RFC 2616, Draft Standard RFC, June 1999.
The specification of Version 1.1 of the HTTP protocol.

[6] Mogul, J., and Leach, P. "Simple Hit-Metering and Usage-Limiting for HTTP," RFC 2227, Proposed Standard RFC, October 1997.
A proposed extension to HTTP to allow a content server to receive hit reports from a proxy cache.

[7] Wessels, D., and Claffey, K. "Internet Cache Protocol (ICP), Version 2," RFC 2186, Informational RFC, September 1997.
A description of the ICP protocol.

[8] Wessels, D., and Claffey, K. "Application of Internet Cache Protocol (ICP), Version 2," RFC 2187, Informational RFC, September 1997.
A description of the structure of cache hierarchies, and their ICP-based interaction.

[9] Melve, I. "Inter Cache Communications Protocols," Internet Draft, Work in progress, **draft-melve-intercache-comproto-00.txt**, November 1998.
An overview of intercache communications protocols currently available, and a collection of references that describe these protocols in further detail.

[10] Cieslak, M., and Foster, D. "Web Cache Coordination Protocol V1.0," Internet Draft, Work in progress, **draft-ietf-wrec-web-pro-00.txt,** June 1999.
A description of Version 1 of the WCCP protocol to support the operation of transparent caches. The protocol defines the interaction between a router and a neighboring cache system.

[11] "Squid Internet Object Cache"—Resource Web page.
http://squid.nlanr.net
A very useful page of resources and references related to the Squid implementation of Web caching.

[12] "Distribution of Stored Information on the Web," Online Tutorial, Ross, K., Institut Eurecom, October 1998. Available at:
http://www.eurecom.fr/~ross/CacheTutorial/DistTutorial.html
A good overview of proxy caching technologies, and also a good analysis of their efficiency of operation.

[13] Stallings, W. "SSL: Foundation for Web Security," *The Internet Protocol Journal*, Volume 1, Number 1, June 1998.

[This article is based in part on material in The ISP Survival Guide, by Geoff Huston, ISBN 0-471-31499-4, published by Wiley in 1998[1]. Used with permission].

GEOFF HUSTON holds a B.Sc and a M.Sc from the Australian National University. Closely involved with the development of the Internet for the past decade, particularly within Australia, he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Technologist in the Internet area for Telstra. He is also an active member of the IETF, and is the chair of the Internet Society Board of Trustees. He is author of *The ISP Survival Guide*, ISBN 0-471-31499-4, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, ISBN 0-471-24358-2, a collaboration with Paul Ferguson. Both books are published by John Wiley & Sons, E-mail: **gih@telstra.net**

Gigabit Ethernet

by William Stallings

In late 1995, the IEEE 802.3 committee formed a High-Speed Study Group to investigate means for conveying packets in Ethernet format at speeds in the gigabit-per-second range. A set of 1000-Mbps standards have now been issued.

The strategy for Gigabit Ethernet is the same as that for 100-Mbps Ethernet. While defining a new medium and transmission specification, Gigabit Ethernet retains the carrier sense multiple access collision detect (CSMA/CD) protocol and frame format of its 10- and 100-Mbps predecessors. So it is compatible with the slower Ethernets, providing a smooth migration path. As more organizations move to 100-Mbps Ethernet, putting huge traffic loads on backbone networks, demand for Gigabit Ethernet is intensifying.

Figure 1 shows a typical application of Gigabit Ethernet. A 1-Gbps LAN switch provides backbone connectivity for central servers and high-speed workgroup switches. Each workgroup LAN switch supports both 1-Gbps links, to connect to the backbone LAN switch and to support high-performance workgroup servers, and 100-Mbps links, to support high-performance workstations, servers, and 100-Mbps LAN switches.

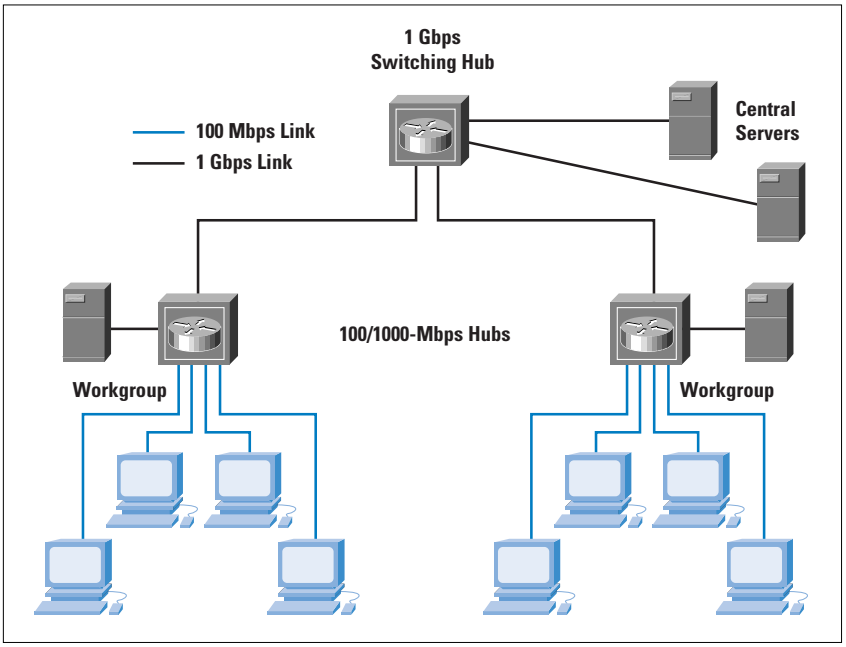


Figure 1: Example Gigabit Ethernet Configuration

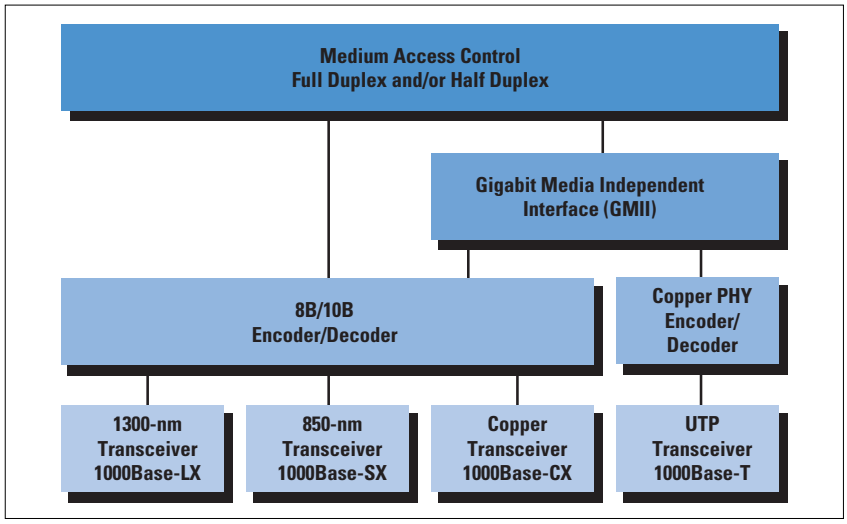
Protocol Architecture

Figure 2 shows the overall protocol architecture for Gigabit Ethernet. The Media Access Control (MAC) layer is an enhanced version of the basic 802.3 MAC algorithm. A separate gigabit medium-independent interface (GMII) has been defined and is optional for all the medium options except unshielded twisted-pair (UTP).

The GMII defines independent 8-bit-parallel transmit and receive synchronous data interfaces. It is intended as a chip-to-chip interface that lets system vendors mix MAC and physical sublayer (PHY) components from different manufacturers.

Two signal encoding schemes are defined at the physical layer. The 8B/10B scheme is used for optical fiber and shielded copper media, and the pulse amplitude modulation (PAM)-5 is used for UTP.

Figure 2: Gigabit Ethernet Layers



Media Access Layer

The 1000-Mbps specification calls for the same CSMA/CD frame format and MAC protocol as used in the 10- and 100-Mbps versions of IEEE 802.3. For traditional Ethernet hub operation, in which only one station can transmit at a time (half-duplex), the basic CSMA/CD scheme has two enhancements:

- *Carrier extension:* Carrier extension appends a set of special symbols to the end of short MAC frames so that the resulting block is at least 4096 bit-times in duration, up from the minimum 512 bit-times imposed at 10 and 100 Mbps. This extension makes the frame length of a transmission longer than the propagation time at 1 Gbps.
- *Frame bursting:* This feature allows for multiple short frames to be transmitted consecutively, up to a limit, without relinquishing control for CSMA/CD between frames. Frame bursting avoids the overhead of carrier extension when a single station has a number of small frames ready to send. extension when a single station has numerous small frames ready to send.

With a LAN switch (full-duplex operation), which provides dedicated rather than shared access to the medium, the carrier extension and frame bursting features are not needed. They are unnecessary because data transmission and reception at a station can occur simultaneously without interference and with no contention for a shared medium. All the gigabit products on the market use a switching technique, and so do not implement the carrier extension and frame bursting.

With a switching technique, full-duplex operation is employed, and the CSMA/CD protocol is not needed. The gigabit specification expands on the pause protocol that is defined for 100-Mbps Ethernet by allowing asymmetric flow control. Using the autonegotiation protocol, a device may indicate that it may send pause frames to its link partner but will not respond to pause frames from its partner.

Physical Layer

The current 1-Gbps specification for IEEE 802.3 includes the following physical-layer alternatives (Figure 3):

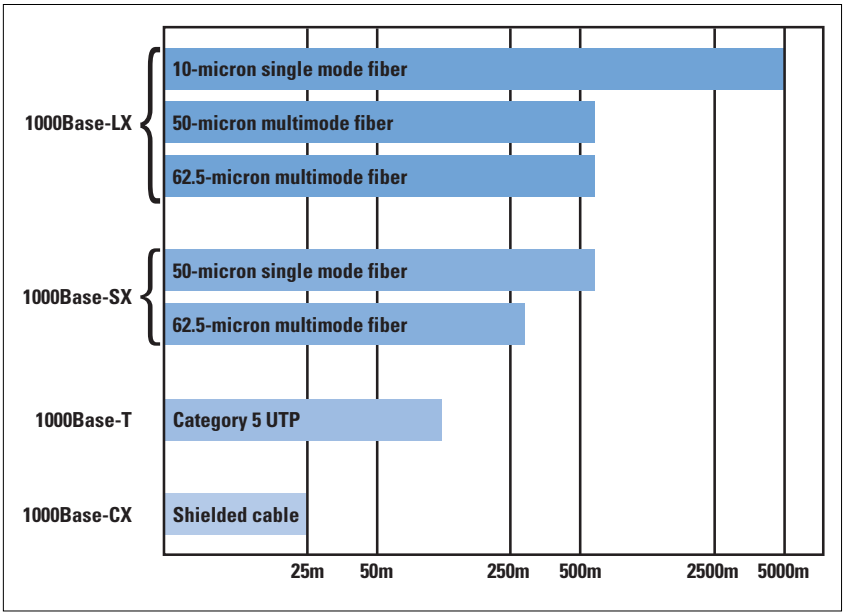


Figure 3: Gigabit Ethernet Media Options (log scale)

- *1000Base-LX:* This long-wavelength option supports duplex links of up to 550 m of 62.5-μm or 50-μm multimode fiber or up to 5 km of 10-μm single-mode fiber. Wavelengths are in the range of 1270 to 1355 nm.
- *1000Base-SX:* This short-wavelength option supports duplex links of up to 275 m using 62.5-μm multimode or up to 550 m using 50-μm multimode fiber. Wavelengths are in the range of 770 to 860 nm.
- *1000Base-CX:* This option supports 1-Gbps links among devices located within a single room or equipment rack, using copper jumpers (specialized shielded twisted-pair cable that spans no more than 25 m). Each link is composed of a separate shielded twisted-pair running in each direction.
- *1000Base-T:* This option makes use of four pairs of Category 5 unshielded twisted-pair copper wires to support devices over a range of up to 100 m.

Digital Signal Encoding Techniques for Gigabit Ethernet

The encoding scheme used for all the Gigabit Ethernet options except twisted-pair is 8B/10B. This scheme is also used in Fibre Channel. With 8B/10B, each 8 bits of data is converted into 10 bits for transmission. The 8B/10B scheme was developed and patented by IBM for use in its 200-megabaud ESCON interconnect system.

- The developers of this code list the following advantages:
- It can be implemented with relatively simple and reliable transceivers at low cost.
- It is well balanced, with minimal deviation from the occurrence of an equal number of 1 and 0 bits across any sequence.
- It provides good transition density for easier clock recovery.
- It provides useful error-detection capability.

The 8B/10B code is an example of the more general *m*B*n*B code, in which *m* binary source bits are mapped into *n* binary bits for transmission. Redundancy is built into the code to provide the desired transmission features by making $n > m$. Figure 4 illustrates the operation of this code. The 8B/10B code actually combines two other codes, a 5B/6B code and a 3B/4B code. The use of these two codes is simply an artifact that simplifies the definition of the mapping and the implementation; the mapping could have been defined directly as an 8B/10B code. In any case, a mapping is defined that maps each of the possible 8-bit source blocks into a 10-bit code block. There is also a function called *disparity control*. In essence, this function keeps track of the excess of zeros over ones or ones over zeros. An excess in either direction is referred to as a disparity. If there is a disparity, and if the current code block would add to that disparity, then the disparity control block complements the 10-bit code block. This complement has the effect of either eliminating the disparity or at least moving it in the opposite direction of the current disparity.

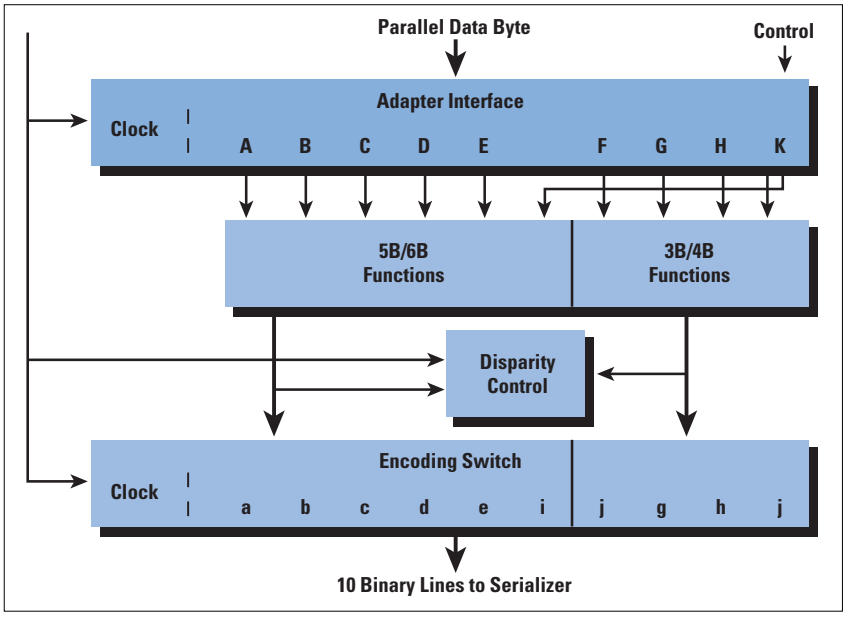


Figure 4: 8B/10B Encoding

The encoding mechanism also includes a control line input, K, which indicates whether the lines A through H are data or control bits. In the latter case, a special nondata 10-bit block is generated. A total of 12 of these nondata blocks are defined as valid in the standard. These blocks are used for synchronization and other control purposes.

For 1000Base-T, the encoding scheme used is PAM-5, over four twisted-pair links. Therefore, each link must provide a data rate of 250 Mbps. PAM-5 provides better bandwidth utilization than simple binary signaling by using five different signaling levels. Each signal element can represent two bits of information (using four signaling levels). In addition, a fifth signal level is used in a forward error correction scheme.

References

A good tutorial on Gigabit Ethernet is [1]. The Gigabit Ethernet Alliance is at **http://www.gigabit-ethernet.org**

[1] Frazier, H., and Johnson, H. "Gigabit Ethernet: From 100 to 1,000 Mbps." *IEEE Internet Computing*, January/February 1999.

[2] *Gigabit Ethernet: Technology and Applications for High-Speed LANs* by Rich Seifert, ISBN 0-201-18553-9, Addison-Wesley, 1998. (Reviewed in *The Internet Protocol Journal*, Volume 1, Number 2, September 1998.)

WILLIAM STALLINGS is a consultant, lecturer, and author of more than a dozen books on data communications and computer networking. He has a PhD in computer science from M.I.T. His latest book is *Data and Computer Communications, Sixth Edition* (Prentice Hall, 1999). His home in cyberspace is **http://www.shore.net/~ws** and he can be reached at **ws@shore.net**

One Byte at a Time: Is Your FTP Active or Passive?

by Thomas M. Thomas, NetCerts

What many people don't know is that the *File Transfer Protocol* (FTP) has multiple modes of operation that can dramatically affect its operation and, as a result, the security of your network. These modes of operation determine whether the FTP server or FTP client initiates the TCP connections that are used to send information from the server to the client. The FTP protocol supports two modes of operation, as follows:

- The first FTP mode of operation is known as *normal,* though it is often referred to as *active*. This mode of operation is typically the default.
- The second FTP mode of operation is known as *passive*.

In active (normal) FTP, the client opens a control connection on port 21 to the server, and whenever the client requests data from the server, the server opens a TCP session on port 20. In passive FTP, the client opens the data sessions, using a port number supplied by the server.

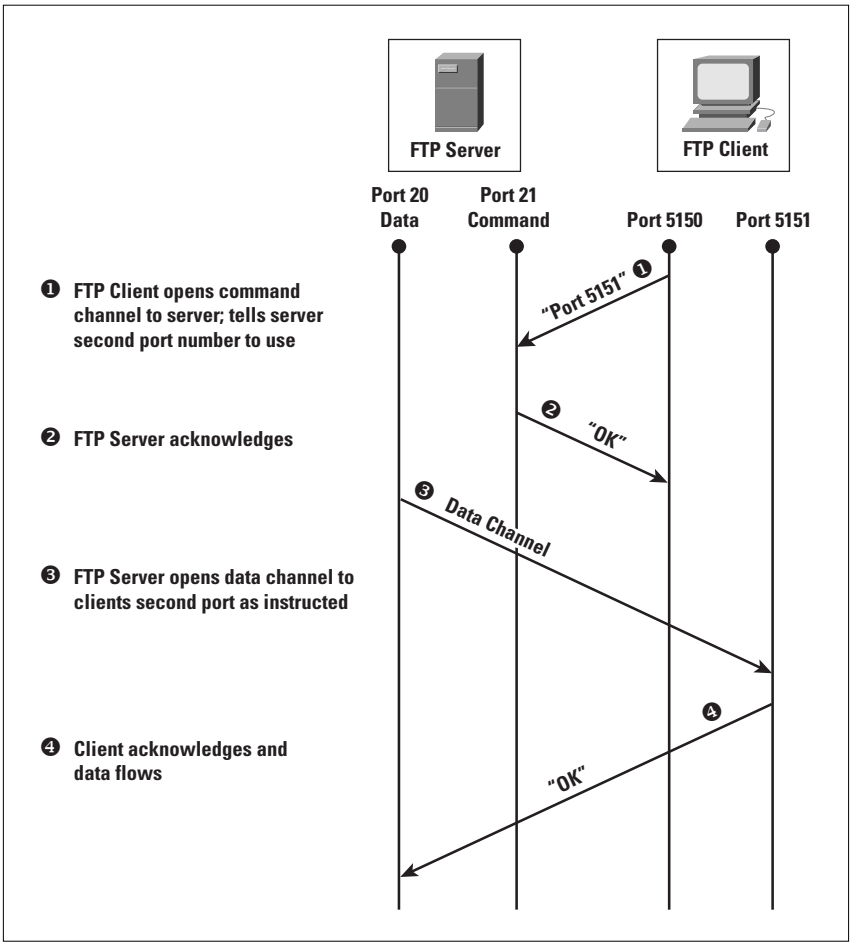
Active FTP Operation

The active mode of operation is less secure than the passive mode. This mode of operation complicates the construction of firewalls, because the firewall must anticipate the connection from the FTP server back to the client program. The steps of this mode of operation are discussed below and are shown in Figure 1.

- The client opens a control channel (port 21) to the server and tells the server the port number to respond on. This port number is a randomly determined port greater than 1023.
- The server receives this information and sends the client an acknowledgement "OK" (ack). The client and server exchange commands on this control connection.
- When the user requests a directory listing or initiates the sending or receiving of a file, the client software sends a "PORT" command that includes a port number > 1023 that the client wishes the server to use for the data connection.
- The server then opens a data connection from port 20 to the client's port number, as provided to it in the "PORT" command.

The client acknowledges and data flows.

Figure 1: Active-Mode FTP Connection



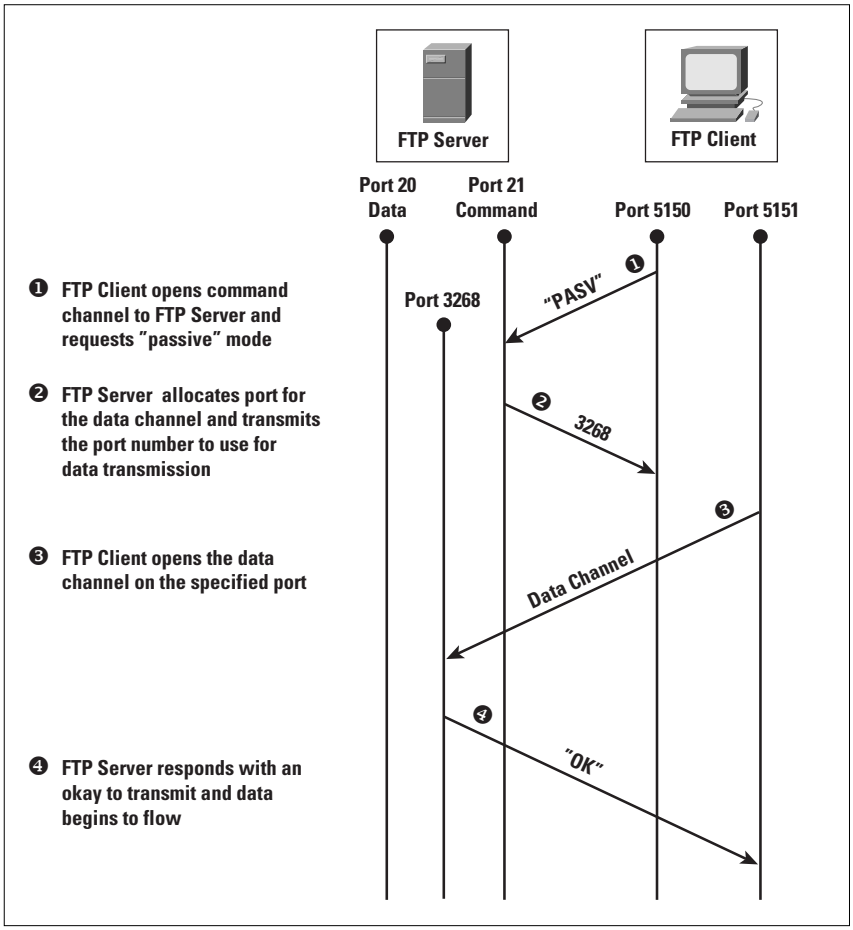
Passive FTP Operation

This mode of operation is assumed to be more secure because all the connections are being initiated from the client, so there is less chance that the connection will be compromised. The reason it is called passive is that the server performs a "*passive open.*" The steps of this mode of operation are discussed below and are shown in Figure 2.

- In passive FTP, the client opens a control connection on port 21 to the server, and then requests passive mode through the use of the "PASV" command.
- The server agrees to this mode, and then selects a random port number (>1023). It supplies this port number to the client for data transfer.
- The client receives this information and opens a data channel to the server-assigned port.

The server receives the data and sends an "OK" (ack).

Figure 2: Passive-Mode FTP Connection



References

[1] *Internetworking With TCP/IP, Volume 1: Principles, Protocols, Architecture, Third Edition,* by Douglas E. Comer, ISBN 0-13-216987-8, Prentice Hall, 1995.

[2] R. Braden, "Requirements for Internet hosts—application and support," RFC 1123, October 1989.

[3] S. Bellovin, "Firewall-Friendly FTP," RFC 1579, February 1994.

[4] P. Deutsch, A. Emtage, A. Marine, "How to Use Anonymous FTP," RFC 1635, May 1994.

THOMAS M. THOMAS II has recently founded his own company, NetCerts (**www.netcerts.com**), to assist network engineers working toward their Cisco Certifications. Before starting NetCerts, Tom worked as a Course Developer at Cisco Systems for the Worldwide Training division. He worked as part of a team on a new course on Multilayer Switching. He also wrote the book *OSPF Network Design Solutions* for Cisco Press. Tom has also worked as a senior network engineer and group leader of the Advanced Systems Solutions Engineering Team for MCI's Managed Network Services. In this capacity, he developed network maintenance Standard Operating Procedures and performed various in-house training duties. Before joining MCI, Tom worked as a technical team leader at AT&T Solutions, where he provided technical support and network management for Cisco routers over ATM and Frame Relay and configured various networking protocols. E-mail: **tothomas@netcerts.com**

Letter to the Editor

The article "Was the Melissa Virus so Different?" (*The Internet Protocol Journal*, Volume 2, Number 2, June 1999) by Barbara Y. Fraser et al makes an interesting comparison between events in our real and virtual lives, comparing e-mail borne viruses with commercial samples delivered to our physical mail boxes. While I think the comparison is a useful exercise, the authors fail to point out one of the fundamental differences between these two worlds.

An electronic message contains a finite amount of information: a careful sender can make sure his identity cannot be revealed. In contrast, a physical "message" (i.e., mail bomb, extortion letter, etc.) contains an essentially unlimited amount of information: from finger prints and material analysis to DNA traces, a potential perpetrator can never be certain that he can deny his involvement. For cyberspace crimes the chance to be caught is (and is perceived to be) much smaller. As a result, many virus authors have but the slimmest motive for their deed.

The fact that the Melissa author was quickly identified because of a hidden signature in Microsoft Word is little comfort. For reasons of privacy, this feature has been disabled: it was a bug, not a feature.

To extend the analogy: suppose a simple device would become available that can look up a person's full ID based on a DNA trace (a few molecules) on any object touched or handled. Move the scanner over the door handle and you know who's been visiting. The ramifications would be extensive. Most likely, the as-yet hypothetical device would be illegal except for police use.

—*Ernst Lopes Cardozo, Aranea Consult BV*
e.lopes.cardozo@aranea.nl

Send us your comments!

We look forward to hearing your comments and suggestions regarding anything you read in this publication. Send us e-mail at: **ipj@cisco.com**

Changes at the IPJ Web Site

Now you can find every issue of *The Internet Protocol Journal* in both PDF and HTML format at **www.cisco.com/ipj**. We are also pleased to announce that Nikkei Business Publications in Tokyo has provided an introduction to IPJ in Japanese, as well as translation of some of the titles from previous issues at: **http://nit.nikkei.co.jp/ipj.html**. We hope to set up similar links with other publications around the world.

Book Reviews

DHCP *DHCP—A Guide to Dynamic TCP/IP Network Configuration,* by Berry Kercheval, ISBN 0-13-099721-8, Prentice Hall PTR, 1998, **http://www.prenhall.com/ptrbooks/ptr_0130997218.html**

First, I should note that this book arrived at the perfect time for me: I am involved in adding *Dynamic Host Configuration Protocol* (DHCP) support to a software product and needed a quick, thorough understanding of DHCP that went into sufficient detail to support some key design decisions. The book provided me with exactly what I wanted. However, as to whether or not this is a book you should own or even want to read, that is a much more difficult question to answer.

Organization

The author begins with a chapter of general background information. Then, in a logical progression, he goes through an overview of DHCP and on to explicit details of both the client and server aspects of the protocol. In other sections he covers server administration, DHCP and IP Version 6 (IPv6), and the future of DHCP. He then briefly reviews a few available implementations. In supporting sections he covers the relationship between DHCP and the *Domain Name System* (DNS), specifically Dynamic DNS. In one chapter he discusses the relationship between directory services and DHCP, in particular, the *Lightweight Directory Access Protocol* (LDAP). He then concludes with three appendices: one lists DHCP vendors, another covers the available DHCP options, and a final appendix provides the DHCP RFCs, RFC 2131 and RFC 2132.

Presentation

Overall, the book is well planned and easy to read. The background information is clearly written and gives sufficient material to assure that even novice readers will not get left behind. The author clearly explains the origins of DHCP in BOOTP and the continuing relationship between the two protocols. He also provides many examples that help make the more difficult aspects of DHCP easier to grasp. The chapters tend to progress in a logical order, making absorption of the fairly technical subject almost easy.

The presentation, however, is somewhat marred by minor errors and omissions. None of these mistakes would confuse an expert, but they will make it harder for the novice to be sure what he or she is to understand. In one example, a client workstation on net 10.0.1.0 is offered, and selects, an address of 10.0.2.32. This scenario is, however, clearly unroutable, and the example only confuses the reader. The author also makes a good effort at defining terms the first time they are used, and then again in an extensive glossary. However, for some reason he never defines two key terms: *broadcast* and *multicast*. Since both techniques are core to understanding DHCP, this oversight is difficult to understand.

The chapters on DHCP are fairly exhaustive in their examination of the protocol from overview to minutiae. The roles of clients, servers, and relay agents are well described and documented with sample packets. Each packet field is thoroughly explained and easy to grasp. However, the sections of LDAP and Dynamic DNS could have been presented better. The reader is left with a glimpse of possible relationships between the protocols, but without enough information to really pull it all together. Notably missing is any mention of remote access and the *Remote Authentication Dial-In User Service* (RADIUS) protocol. DHCP and RADIUS perform similar functions in different situations, and there has been much discussion in the past year or two about use of DHCP to manage RADIUS IP address assignments.

Summary

This book sets out to accomplish a limited goal: informing the reader about the basics of DHCP. A couple of detours along the way provide useful information about related technologies (such as DNS and LDAP). The author makes no assumptions about the user's technical capability and level of knowledge. This is perhaps the book's major strength and its biggest weakness. Because of his assumptions about the reader's technical ability, a lot of space is devoted to giving background and reference information assuring that the reader has the necessary foundation to understand the more complex aspects of DHCP. If the background information and appendices (all of which are available on the net and consist mostly of the RFCs anyway) are removed from the book, little is left: without the appendices there are only 144 pages. Given that the book costs $45, and that the 144 pages are essentially a guided explanation of the RFCs anyway, the technically competent reader might do just as well to download the RFCs and slog through them.

However, for the non-technical reader, or someone who just wants it all in one convenient volume, the author's approach is well worth the cost of the book and the (short) time required to read it. Explanations are clear and concise, terms are well defined, and everything the reader needs to grasp about the complexities of DHCP is right there, in a logical order.

—Richard Perlman, Lucent Technologies
perl@lucent.com

Information Warfare

Information Warfare and Security, Dorothy E. Denning, ISBN 0-201-43303-6, Addison-Wesley, 1999, **http://www.awl.com/cseng/0-201-43303-6/**

It has been said that "information is power," and they who control the information control the power. Whether the information is broadcast on the evening news, printed in a newspaper, etched on stone tablets, or published on a USENET newsgroup or Internet Web page, we rely on information in our daily lives, and trust that most of the information we receive and process is accurate.

"Information warfare." What images does it conjure up for you? Propaganda wars via pamphlets dropped from airplanes, or "cyber-terrorists" versus the FBI on the Internet—or something else entirely? Dr. Denning covers all bases in this, her latest book. The "warfare" of the title is specifically the battle between the good guys and "information terrorists."

This book is a textbook for a course by the same name at Georgetown University. No one, however, should be scared off by this knowledge. This book is incredibly approachable, intended for a broad audience. It is an introduction to information warfare, but really concentrates on computer- and network-based information. Anyone involved or interested in computer and network security would benefit from this book. Many sections are self-contained, so a reader can jump back and forth among the sections. All the sections are interesting and informative, and should be to both the highly technical reader as well as those for whom technology is peripheral to their jobs, but who require or desire deeper and broader knowledge of information warfare.

About the Author

Dorothy E. Denning is Professor of Computer Science at Georgetown University. She is a well-known expert in the areas of computer security and cryptography, and has been called as an expert witness to testify before the U.S. Congress. She is the author of over 100 papers on computer and Internet security, and has written three other books in addition to this one: *Cryptography and Data Security* (a coeditor with Peter Denning), *Rights and Responsibilities of Participants in Networked Communities,* with Herbert S. Lin, and *Internet Besieged: Countering Cyberspace Scofflaws*. She is also a frequent contributor to security-related publications.

Organization

Information Warfare and Security has three parts. Part 1 starts with a very exciting (and still timely) discussion of the role information warfare played in the Gulf War in the early 1990s. The tone and flavor of this opening chapter continues throughout the book. Randomly put your finger in the book and you will be able to start an enjoyable and interesting read (though I recommend reading beginning to end). Part 1 introduces basic concepts upon which the work is built. Chapters 2 and 3 present a taxonomy of information warfare, relating it to information security and assurance, and suggesting four arenas of activity: play, crime, individual rights, and national security. The author discusses goals, motivations, culture, and concerns. Included is the no-doubt apocryphal, but always fun, quote attributed to Secretary of State for War Henry Stimson, upon the 1929 "discovery" of the Black Chamber codebreaking operation: "Gentlemen, do not read one another's mail."

Part 2 focuses on offense. This section covers topics that, for the most part, will be new to many readers. The chapters cover open source material and privacy (and piracy of information), "social engineering," and its kin. The threat from insiders—legitimate and those who have broken in, gets a thorough treatment. Eavesdropping also is examined, from cellular and pager intercepts, to the mysterious-to-most-people area of traffic analysis, to surveillance, packet-sniffing, and other electronic eavesdropping attacks.

Chapter 8 looks in detail at well-known computer hacking techniques and the tools that implement the attacks. Chapter 9 discusses identity theft, including forged e-mail and stolen accounts, IP-spoofing (stealing the identity of a computer), and Trojan Horse attacks. Finally, Part 2 ends with a chapter dedicated to computer viruses, both real and hoaxes.

Topics discussed in Part 3, "Defensive Information Warfare," will be familiar to most readers who understand computer and network security. Chapter 11 not only describes cryptographic techniques for protecting information, but also covers *steganography,* or "the practice of hiding a message in such a manner that its very existence is concealed"—and anonymity. Chapter 12, "How to Tell a Fake," deals with methods for determining identity or trustworthiness of entities or information. Chapter 13 talks about access control mechanisms, including firewalls, and intrusion detection. Covering vulnerability monitoring and analysis, risk analysis, risk management, and incident response, Chapter 14 possibly should have started Part 3. Devices, mechanisms, and methods should be deployed after an understanding of what is contained in this chapter. Part 3, and the book, end with a chapter dedicated to discussing the role of government in defensive information warfare. Also included are descriptions of recent (1990s) actions, laws, and initiatives of the U.S. Government in this area.

Throughout, the book is seasoned with stories—infowar stories, if you will—and background information, allowing the novice not only to understand, but also to enjoy learning what is contained within.

A Book for the Lecture Hall or Armchair

It is not surprising that *Information Warfare and Security* so thoroughly covers the space of information warfare theory, measures, and countermeasures, not because it weighs in at over 500 pages, but because it was written as a text for a course that had to cover all of this material. What may be surprising to readers unfamiliar with Dr. Denning is that such complete coverage could be done in such an easy-to-read way. I have no doubt that this book is and will continue to be useful and effective in the classroom. In addition, the reader studying for accreditation in a field requiring this knowledge, or the professional wanting to "brush up," "fill in," or just "kick back," will find much here to commend itself.

—Frederick M. Avolio, Avolio Consulting
fred@avolio.com

Cryptonomicon

Cryptonomicon, Neal Stephenson, ISBN 0-380-97346-4, Avon Books, 1999. **http://www.cryptonomicon.com/main.html**

It isn't often that you find reviews of works of fiction in these pages, but *Cryptonomicon* deserves special treatment. Neal Stephenson's latest work is a 918-page science fiction World War II thriller that I couldn't put down. You have to love a novel that has plot points that depend on the technical details of prime number theory, Pretty Good Privacy (PGP), public key infrastructure (PKI), Secure Shell (SSH), Global Positioning System (GPS), secure e-mail, and other Internet applications. Truly this is an epic novel of techno-epic proportions.

The story takes places during both World War II and modern times. The contemporary action revolves around an offshore data haven created by a Silicon Valley startup with the usual coterie of managers, venture capitalists, lawyers with class-action suits, marketeers, and nerds that you'll easily recognize. These entrepreneurs think nothing of flying across the Pacific to attend a meeting and then flying home to get in some quality family time.

The war setting revolves around a small group of code crackers who travel around the globe planting misinformation behind German and Japanese lines. The two groups are literally related: the modern generation is the progeny of the wartime crackers. Both groups are going after hidden caches of gold, among other things, buried near the Philippines.

Technology

There is much technology here for any self-respecting computer geek to digest. Think of Tom Clancy playing with the latest laptops and the Internet rather than with the latest guns. There is even an appendix describing the technical details of one of the crypto algorithms using synchronized decks of playing cards (a key plot point in the book). Stephenson blends in descriptions of undersea cable laying and salvage operations with the cracking of the *Enigma*[1] codes and hunting down German submarines. At one point, the code-cracking wartime division has to change its numerical designation because it can be factored into two prime numbers—too obvious.

One of my favorite scenes happens early in the book, when the modern-day principals of the crypto firm are meeting some of their backers and potential clients for the first time. The firm's engineer (using the built-in pinhole camera of the laptop) programs his UNIX laptop to surreptitiously capture a photo of whoever is using the keyboard during a demo of the firm's crypto technology, but hides his program in a way that any UNIX hacker would appreciate. He then e-mails the collected digital photos to a friend to try to confirm their identity.

Balance

Unlike Clancy, this book has characters with some depth to them and doesn't overdo the technology. The relationship of the war and modern-day periods is nicely tied together in the end, and the familiarity of the modern-day business relationships is sometimes almost too painful to read.

—David Strom, publisher of Web Informant
david@strom.com

References

[1] See **http://www.nsa.gov:8080/museum/enigma.html**

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and appli-cation management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

Fragments

More ICANN News

The *Internet Corporation for Assigned Names and Numbers* (ICANN) recently announced that seven additional applicant companies have met its registrar accreditation criteria.

As accredited registrars, these seven companies will compete in the market for domain name registration services in the **.com**, **.net**, and **.org** domains. In addition, they will be able to participate the ongoing testbed program for the *Shared Registry System,* which allows multiple ICANN-accredited registrars to provide domain name registration services in these domains. Under an agreement announced August 6 by the U.S. Department of Commerce and Network Solutions, Inc. (NSI—the developer of the Shared Registry System), new registrars that have signed an accreditation agreement with ICANN will be eligible to join the initial five testbed registrars as participants in the testbed operation. The testbed phase is currently scheduled to conclude on September 10, 1999.

The seven new companies join the 57 companies that have already been accredited by ICANN starting in April, 1999. Until the initial introduction of competition in June, registration services in the **.com**, **.net**, and **.org** domains were provided solely by NSI under a 1992 Cooperative Agreement with the U.S. Government.

The additional seven companies named are: CommuniTech.Net, Inc. (United States), GANDI (France), iDirections, Inc. (United States), InterNeXt (France), ProBoard Technologies (United States), PSI-USA (United States), and Signature Domains, Inc. (United States). Further information about these companies will be made available on the ICANN Web site:

http://www.icann.org/registrars/accreditation.html

Under an October 6, 1998 amendment to the Cooperative Agreement between NSI and the U.S. Government, the process of opening the Internet Domain Name System's three largest domains to competition was launched with a testbed phase that began on April 26. Five companies were initially accredited to use the NSI Shared Registry System in a test operation designed to ensure that the introduction of competition occurs in a smooth, coordinated manner.

By qualifying to be accredited as registrars, the seven new registrars join the five original testbed registrars, as well as the 52 other companies that have already qualified for ICANN accreditation. The Shared Registry System testbed program has been expanded to extend to all accredited registrars that sign the standard testbed registrar agreements with NSI and meet technical certification requirements.

ICANN is a non-profit, international corporation formed in September 1998 to oversee a select set of Internet technical management functions currently managed by the U.S. Government, or by its contractors and volunteers. Specifically, ICANN is assuming responsibility for coordinating the management of the *Domain Name System* (DNS), the allocation of IP address space, the assignment of protocol parameters, and the management of the root server system. For more information, see **`http://www.icann.org`**. Here you will also find information about ICANN's upcoming public meetings.

INET 2000

INET 2000: The Internet Global Summit, is a special INET. Hosted by the Internet Society, the Summit will be held 18–21 July 2000, in Yokohama, Japan. The place, the date, and the fact that it is the 10th anniversary of this important event all mark it as an exceptional year.

To be considered as a speaker, panelist, tutorial instructor, or poster presenter, please see **`http://www.isoc.org/inet2000/callforabstracts.shtml`** for submission instructions and to read about this year's theme, "Global Distributed Knowledge for Everyone."

INET is the premier international event for Internet and internetworking professionals. Nowhere can such a broad cross-section of important movers of the Internet be found in one single location.

We look forward to receiving your abstract and seeing you in Japan!

—Jean-Claude Guedon and Jun Murai
Co-Chairs, INET 2000 Program Committee

Y2K and The Internet

As the countdown to the Year 2000 continues, a number of efforts are underway to ensure that the Internet continues to operate normally on January 1, 2000. Here we include some pointers to recent activities.

On July 30, 1999, the *President's Council on Year 2000 Conversion,* convened a roundtable meeting to examine the readiness of the Internet for the Year 2000 date change, and to coordinate efforts to maintain Internet performance and reliability during the transition to the new millennium. The roundtable brought together roughly 100 prominent organizations and individuals from different parts of the Internet community to discuss the Internet's Y2K readiness. Meeting participants included small and large ISPs, equipment vendors, root name server and domain registries, exchange points, network time servers, industry associations, and government officials. For more information see:
`http://www.y2k.gov/` and **`http://www.mids.org/y2k/`**

For small- and medium-sized businesses in the U.S. and in key trading partner countries, the U.S. Department of Commerce (DoC) is providing a strategic management tool to help battle the millennium bug. The *Y2K Self-Help Tool/CD-ROM* contains a software program that enables users to complete an inventory of assets that may be susceptible to Y2K problems, gauge the criticality of business processes, develop contingency plans and conduct remediation activities.

This CD-ROM contains a 10-minute discussion video, the software program for managing your Y2K process, a self-assessment checklist, contingency planning template, user guide and hotlinks to many helpful Y2K sites. It has been produced in several languages including English, Spanish, Mandarin Chinese, Japanese, French, Portuguese, Arabic and Russian. The software was developed by the DoC's National Institute of Standards and Technology Manufacturing Extension Partnership (MEP) in cooperation with the U.S. Department of Agriculture and the U.S. Small Business Administration.

To receive just the software, visit: **www.nist.gov/y2k/software.htm** and download *Conversion 2000: Y2K Jumpstart Kit*. To receive the complete CD-ROM with video and hotlinks, you can call 1-800-Y2K-7557 and ask for the Self-Help Tool in any of the languages listed above. If you are an association or organization interested in multiple copies of the CD-ROM for your members and staff, click on order form, print the form, complete the requested information, and fax it to 202-482-0077. Please note that there is a minimal charge for orders over 100 copies for duplication and shipping.

The *Internet Engineering Task Force* (IETF) has examined all of the protocol standards and related documents to identify any potential inherent Y2K problems in the Internet Protocol Suite. The resulting report, RFC 2626, "The Internet and the Millennium Problem (Year 2000)" can be found at **http://www.ietf.org/rfc/rfc2626.txt**

See also:

http://www.apia.org
http://www.nety2k.org/
http://www.cert.org/y2k/indmessage.html
http://www.icann.org/committees/dns-root/y2k-statement.htm

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Cisco News Publications Group, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

December 1999 Volume 2, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

In June 1992 when I was editor and publisher of *ConneXions—The Interoperability Report*, we published an article entitled "First IETF Internet Audiocast." Steve Casner and Steve Deering wrote: "The March Internet Engineering Task Force (IETF) meeting in San Diego was an exciting one for those interested in teleconferencing. In addition to several sessions on teleconferencing topics, we managed to pull off a 'wild idea' suggested by Allison Mankin from MITRE: live audio from the IETF site was 'audiocast' using IP multicast packet audio over the Internet to participants at 20 sites on three continents spanning 16 timezones."

Multicast has come a long way since 1992. Today, every IETF meeting features several live streams of not only audio but also video and slide presentations. Multicast continues to be developed in the IETF, as protocols and tools are being revised and refined. In two articles, Jon Crowcroft and Mark Handley describe the technologies behind multicast. The first article, included in this issue, looks at the current state of multicast. The second article, to appear in a future issue of IPJ, will look at the problems that need to be solved before multicast can become a truly scalable service for the Internet.

Research into new, high-speed networking technologies and applications is taking place in many parts of the world. One example of such a research effort can be found in the Internet2 Project. Larry Dunn describes some of the technology and application development being conducted by Internet2 members.

Interest in *IP Version 6* (IPv6) is growing as organizations contemplate a world where millions of devices such as cellphones, PDAs, cable TV set-top boxes and so on are "Internet Ready." The formation of the *IPv6 Forum* (**www.ipv6forum.com**) is some indication of this interest. We will look at a particular IPv4-to-IPv6 transition strategy in our next issue. In the meantime, Peter Salus takes a historical look at Internet addressing in our series "One Byte at a Time."

And so we reach the end of 1999 and the end of Volume 2 of *The Internet Protocol Journal*. We wish you a pleasant holiday season and an uneventful transition to Y2K.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Internet Multicast Today

by Mark Handley, ACIRI and Jon Crowcroft, University College London

When you need to send data to many receivers simultaneously, you have two options: repeated transmission and broadcast. Repeated transmission may be acceptable if the cost is low enough and delivery can be spread out over time, as with junk mail or electronic mailing lists. Otherwise, a broadcast solution is required. With real-time multimedia, repeated delivery is feasible, but only at great expense to the sender, who must invest in large amounts of bandwidth. Similarly, traditional broadcast channels have been very expensive if they cover significant numbers of recipients or large geographic areas. However, the Internet offers an alternative solution: IP multicast effectively turns the Internet into a broadcast channel, but one that anyone can send to without having to spend huge amounts of money on transmitters and government licenses. It provides efficient, timely, and global many-to-many distribution of data, and as such may become the broadcast medium of choice in the future.

The Internet is a datagram network, meaning that anyone can send a packet to a destination without having to preestablish a path. Of course, the boxes along the way must have either precomputed a set of paths, or they must be relatively fast at calculating one as needed, and typically, the former approach is used. However, the sending host need not be aware of or participate in the complex route calculation; nor does it need to take part in a complex *signaling* or *call setup* protocol. It simply addresses the packet to the right place, and sends it. This procedure may be a more complex procedure if the sending or receiving systems need more than the default performance that a path or network might offer, but it is the *default* model.

Adding multicast to the Internet does not alter the basic model. A sending host can still simply send, but now there is a new form of address, the multicast or host group address. Unlike unicast addresses, hosts can dynamically subscribe to multicast addresses and by so doing cause multicast traffic to be delivered to them. Thus the IP multicast *service model* can be summarized:

- Senders send to a multicast address
- Receivers express an interest in a multicast address
- Routers conspire to deliver traffic from the senders to the receivers

Sending multicast traffic is no different from sending unicast traffic except that the destination address is slightly special. However, to receive multicast traffic, an interested host must tell its local router that it is interested in a particular multicast group address; the host accomplishes this task by using the *Internet Group Management Protocol* (IGMP).

Point-to-multipoint communication is nothing new. We are all used to the idea of broadcast TV and radio, where a shared medium (the radio frequency [RF] spectrum) is partitioned among users (transmitter or TV/radio station owners). It is a matter of regulation that there is typically only one unique sender of particular content on any given frequency, although other parts of the RF spectrum are given over to free use for multiparty communication (police radio, citizen band radio, and so on).

The Internet multicast *model*[3] is very similar. The idea is to convert the mesh wide-area network that is the Internet (whether the public Internet, a private enterprise net, or intranet makes no difference to the model), into a shared resource for senders to send to multiple participants, or groups.

To make this group communication work for large-scale systems—in the sense of a large number of recipients for a particular group, or in the sense of a large number of senders to a large number of recipients, or in the sense of a large number of different groups—it is necessary, both for senders and for the routing functions to support delivery, to have a system that can be largely independent of the particular recipients at any one time. In other words, just as a TV or radio station does *not know* who is listening when, an Internet multicast sender does not know who might receive packets it sends. If this scenario sends out alarm bells about security, it shouldn't. A unicast sender has no assurance about who receives its packets either. Assurances about disclosure (privacy) and authenticity of sender/recipient are largely separate matters from simple packet delivery models. Security is a topic of much research and the focus for the recently formed *Internet Research Task Force* (IRTF) research group, *Secure Multicast Group* (SMuG).

The Internet multicast model is an extension of the datagram model; it uses the fact that the datagram is a self-contained communications unit that not only conveys data from source to destination, but also conveys the source and destination address information. In other words, in some senses, datagrams *signal* their own path, both with a source and a destination address in every packet.

By adding a range of addresses dedicated for sending to groups, and providing independence between the address allocation and the rights to send to a group, the analogy between RF spectrum and the Internet multicast space is maintained. Some mechanism, as yet unspecified, is used to dynamically choose which address to send to. Suffice it to say that for now, the idea is that somehow, elsewhere, the address used for a multicast session or group communication activity is chosen so that it does not clash with other uses or users, and is advertised to potential senders and receivers.

Unlike the RF spectrum, an IP packet to be multicast carries a unique source identifier, in that such packets are sent with the normal unicast IP address of the interface of the sending host.

It is also worth noting that an address that is being used to signify a group of entities must surely be a logical address (or in some senses a name) rather than a topological or topographical identifier. We shall see that this means there must be some service that maps such a logical identifier to a specific set of locations in the same way that a local unicast address must be mapped (or bound) to a specific location. In the multicast case, this mapping is distributed. Note also that multicast Internet addresses are in some sense "host group" addresses, in that they indicate a set of hosts to deliver to. In the Internet model, there is a further level of multiplexing, that of transport-level ports, and there is room for some overlap of functionality, since a host may receive packets sent to multiple multicast addresses on the same port, or multiple ports on the same multicast address.

This model raises numerous questions about address and group management, such as how these addresses are allocated. The area requiring most change, though, is in the domain of the routing. Somehow the routers must be able to build a distribution tree from the senders to all the receivers for each multicast group. The senders don't know who the receivers are (they just send their data), and the receivers don't know who the senders are (they just ask for traffic destined for the group address), so the routers have to do something without help from the hosts. We will examine this scenario in detail in the section "Multicast Routing."

Roadmap

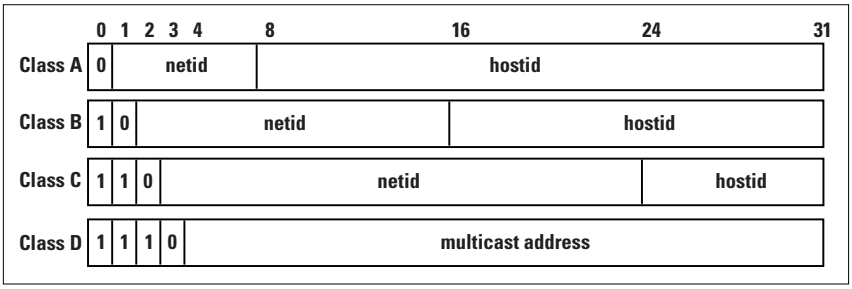
The functions that provide the Standard Internet Multicast Service can be separated into host and network components. The interface between these components is provided by IP multicast addressing and IGMP group membership functions, as well as standard IP packet transmission and reception. The network functions are principally concerned with multicast routing, while host functions also include higher-layer tasks such as the addition of reliability facilities in a transport-layer protocol. That's the order in which we cover each of these functions in the rest of this article. At the end of the article we list the current status of *Internet Engineering Task Force* (IETF) specification for the various components.

Host Functions

As we stated above, host functionality is extended through the use of the IGMP protocol. Hosts and routers, which we will look at later, must be able to deal with new forms of addressing was first designed, it was divided into classes as shown in Figure 1.

Figure 1: Internet Address Classes

A	1.0.0.0	to 126.255.255.255
B	128.0.0.0	to 191.255.255.255
C	192.0.0.0	to 223.255.255.255
D	224.0.0.0	to 239.255.255.255



Originally Class A was intended for large networks, B for midsize networks, and C for small networks. Class D was later allocated for multicast addresses. Since then, classless addressing has been introduced to solve Internet scaling problems, and the rules for Classes A, B, and C no longer hold, but Class D is still reserved for multicast, so all IPv4 multicast addresses start with the high-order 4-bit "nibble": 1110

In other words, from the 2^{32} possible addresses, 2^{28} are multicast, meaning that there can be up to about 270 million different groups, each with as many senders as can get unicast addresses! This number is many orders of magnitude more than the RF spectrum allows for typical analog frequency allocations.

For a host to support multicast, the host service interface to IP must be extended in three ways:

- A host must be able to join a group, meaning that it must be able to reprogram its network level, and possibly, consequentially, the lower levels, to be able to receive packets addressed to multicast group addresses.
- An application that has joined a multicast group and then sends to that group must be able to select whether it wants the host to loopback the packets it sent so that it receives its own packets.
- A host should be able to limit the *scope* with which multicast messages are sent. The Internet Protocol contains a *Time-To-Live* (TTL) field, used originally to limit the lifetime of packets on the network, both for safety of upper layers, and for prevention of traffic overload during temporary routing loops. It is used in multicast to limit how "far" a packet can go from the source. We will see below how scoping can interact with routing.

When an application tells the host networking software to join a group, the host software checks to see if the host is a member of the group. If not, it makes a note of the fact, and sends out an IGMP membership report message. It also maps the IP address to a lower-level address and reprograms its network interface to accept packets sent to that address. There is a refinement here: a host can join "on an interface;" that is, hosts that have more than one network card can decide which one (or more than one) they wish to receive multicast packets via. The implication of the multicast model is that it is "pervasive," so it is usually necessary to join on only one interface.

Taking a particular example to illustrate the IP-level to link-level mapping process, if a host joins an IP multicast group using an Ethernet interface, there is a mapping from the low 24 bits of the multicast address into the low 24 (out of 48) bits of the Ethernet address. Since this mapping is a many-to-one mapping, there may be multiple IP multicast groups occupying the same Ethernet address on a given wire, though it may be made unlikely by the address allocation scheme. An Ethernet LAN is a shared-medium network, thus local addressing of packets to an Ethernet group means that the packets are received by Ethernet hardware and delivered to the host software of *only* those hosts with members of the relevant IP group. Therefore, host software is generally saved the burden of filtering out irrelevant packets. Where there is an Ethernet address clash, software can filter the packets efficiently.

Operation of the IGMP protocol can be summarized as follows:

- When a host first joins a group, it programs its Ethernet interface to accept the relevant traffic, and it sends an IGMP Join message on its local network. This message informs any local routers that there is a receiver for this group now on this subnet.
- The local routers remember this information, and arrange for traffic destined for this address to be delivered to the subnet.
- After a while, the routers wonder if there is still any member on the subnet, and send an IGMP query message to the multicast group. If the host is still a member, it replies with a new message unless it hears someone else do so first. Multicast traffic continues to be delivered.
- Eventually the application finishes, and the host no longer wants the traffic. It reprograms its Ethernet interface to reject the traffic, but the packets are still sent until the router times the group out and sends a query to which no one responds. The router then stops delivering the traffic.

Thus joining a multicast group is quick, but leaving can be slow with IGMP Version 1. IGMP Version 2 reduces the leave latency by introducing a "Leave" message and a set of rules to prevent one receiver from disconnecting others when it leaves. IGMP Version 3 (not yet deployed) introduces the idea of *source-specific* joining and leaving, whereby a host can subscribe (or reject) traffic from individual senders rather than the group as a whole, at the expense of more complexity and extra state in routers.

Multicast Routing

Given the multicast service model described above, and the restrictions that senders and receivers don't know each others' location or anything about the topology, how do routers conspire to deliver traffic from the senders to the receivers?

We shall assume that if a sender and a receiver did know about each other, they could each send unicast packets to the other. In other words, there is a network with bidirectional paths and an underlying unicast routing mechanism already running. Given this network, there is a spectrum of possible solutions. At one extreme, we can flood data from the sender to all possible receivers and have the routers for networks where there are no receivers prune off their branches of the distribution tree. At the other extreme, we can communicate information in a multicast routing protocol conveying the location of all the receivers to the routers on the paths to all possible senders. Neither method is particularly desirable on a global scale, so the most interesting solutions tend to be hybrid solutions that lie between these extremes.

In the real world, there are many different multicast routing protocols, each with its own advantages and disadvantages. We shall explain each of the common ones briefly, because a working knowledge of their pros and cons helps us understand the practical limits to the uses of multicast.

Flood and Prune Protocols

Flood and Prune Protocols are more correctly known as *reverse-path multicast* algorithms. When a sender first starts sending, traffic is flooded out through the network. A router may receive the traffic along multiple paths on different interfaces, in which case it rejects any packet that arrives on any interface other than the one it would use to send a unicast packet back to the source. It then sends a copy of each packet out of each interface other than the one back to the source. In this way, each link in the whole network is traversed at most once in each direction, and the data is received by all routers in the network.

So far, this process describes *reverse-path broadcast*. Many parts of the network will be receiving traffic, even though there are no receivers there. These routers know they have no receivers (otherwise IGMP would have told them) and they can then send prune messages back toward the source to stop unnecessary traffic from flowing. Thus the delivery tree is pruned back to the minimal tree that reaches all the receivers. The final distribution tree is what would be formed by the union of shortest paths from each receiver to the sender, so this type of distribution tree is known as a *shortest-path tree* (strictly speaking, it's a reverse shortest path tree—typically the routers don't have enough information to build a true forward shortest-path tree).

Two commonly used multicast routing protocols fall in the class: the *Distance Vector Multicast Routing Protocol* (DVMRP)[4] and *Protocol Independent Multicast Dense-Mode* (PIM-DM)[5]. The primary difference between these protocols is that DVMRP computes its own routing table to determine the best path back to the source, whereas PIM Dense-Mode uses the routing table of the underlying unicast routing system, hence the term "Protocol Independent."

It should be fairly obvious that sending traffic *everywhere* and getting people to tell you what they don't want is not a particularly scalable mechanism. Sites get traffic they don't want (albeit very briefly), and routers not on the delivery tree need to store prune state. For example, if a group has one member in the UK and two in France, routers in Australia still get some of the packets, and they need to hold prune state to prevent more packets from arriving! However, for groups where most places actually do have receivers (receivers are "densely" distributed), this sort of protocol works well. So although these protocols are poor choices for a global scheme, they might be appropriate within some organizations.

MOSPF

Multicast Open Shortest Path first (MOSPF[12]) isn't really a category, but a specific instance of a protocol. MOSPF is the multicast extension to *Open Shortest Path First* (OSPF[11]), which is a unicast link-state routing protocol.

Link-state routing protocols work by having each router send a routing message periodically listing its neighbors and how far away they are. These routing messages are flooded throughout the entire network, so every router can build up a map of the network. This map is then used to build forwarding tables (using a Dijkstra algorithm) so that the router can decide quickly which is the correct next hop for a particular packet.

Extending this concept to multicast is achieved simply by having each router also list in a routing message the groups for which it has local receivers. Thus given the map and the locations of the receivers, a router can also build a multicast forwarding table for each group.

MOSPF also suffers from poor scaling. With flood-and-prune protocols, data traffic is an *implicit* message about where there are senders, so routers need to store unwanted state where there are no receivers. With MOSPF, there are *explicit* messages about where all the receivers are, so routers need to store unwanted state where there are no senders. However, both types of protocol build very efficient distribution trees.

Center-Based Trees

Rather than flooding the data everywhere, or flooding the membership information everywhere, algorithms in the center-based trees category map the multicast group address to a particular unicast address of a router, and they build explicit distribution trees centered around this particular router. Three main problems need to be solved to get this approach to work:

- How is the mapping from group address to center address performed?
- How is the center location chosen so that the distribution trees are efficient?
- How is the tree actually constructed given the center address?

Different protocols have come up with different solutions to these problems. Three center-based tree protocols are worth exploring because they illustrate different approaches: *Core-Based Trees* (CBT), *PIM Sparse-Mode* (PIM-SM), and the *Border Gateway Multicast Protocol* (BGMP). However, we will leave discussion of BGMP until our second article because it is not currently deployed.

Core-Based Trees

Core-Based Trees (CBT[1]) was the earliest center-based tree protocol, and it is the simplest.

When a receiver joins a multicast group, its local CBT router looks up the multicast address and obtains the address of the Core router for the group. It then sends a Join message for the group toward the Core. At each router on the way to the Core, forwarding state is instantiated for the group, and an acknowledgment is sent back to the previous router. In this way, a multicast tree is built, as shown in Figure 2.

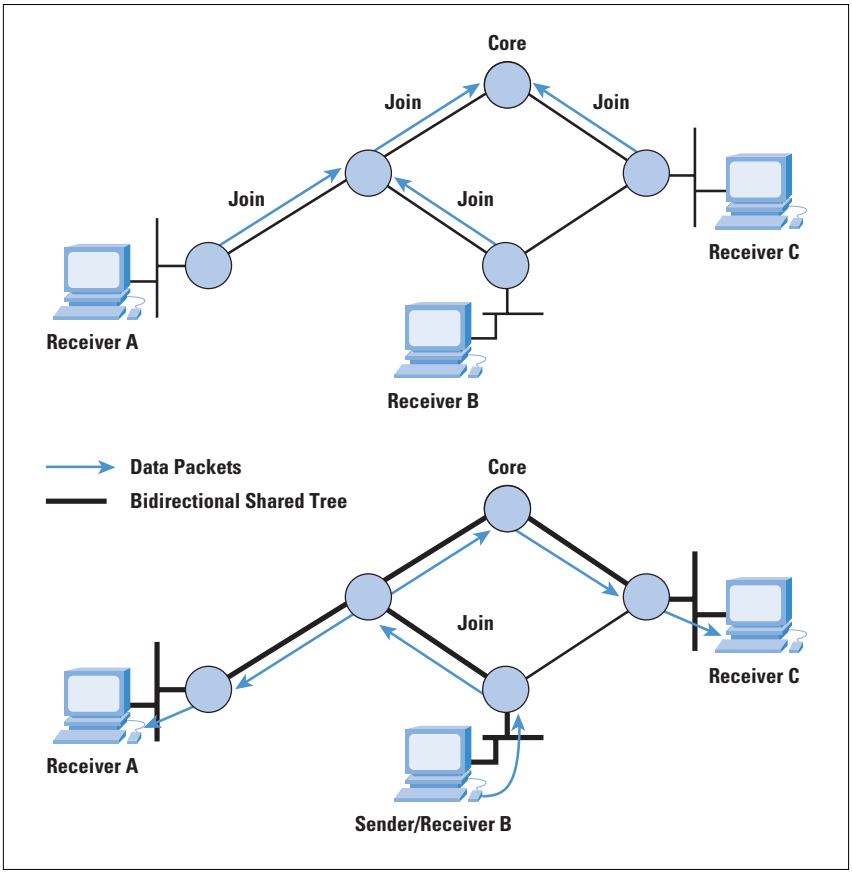


Figure 2: Formation of a CBT Bidirectional Shared Tree

If a sender (that is, a group member) sends data to the group, the packets reach its local router, which forwards them to any of its neighbors that are on the multicast tree. Each router that receives a packet forwards it out of all its interfaces that are on the tree except the one the packet came from. The style of tree CBT builds is called a "bidirectional shared tree," because the routing state is "bidirectional"—packets can

flow both up the tree toward the Core and down the tree away from the Core, depending on the location of the source, and packets are "shared" by all sources to the group. This scenario is in contrast to "unidirectional shared trees" built by PIM-SM as we shall see later.

IP multicast does not require senders to a group to be members of the group, so it is possible that a sender's local router is not on the tree. In this case, the packet is forwarded to the next hop toward the Core. Eventually the packet will either reach a router that is on the tree, or it will reach the Core, and it is then distributed along the multicast tree.

CBT also allows multiple Core routers to be specified, adding a little redundancy in case the Core becomes unreachable. CBT never properly solved the problem of how to map a group address to the address of a Core. In addition, good Core placement is a difficult problem. Without good Core placement, CBT trees can be quite inefficient, and so CBT is unlikely to be used as a global multicast routing protocol.

However, within a limited domain, CBT is very efficient in terms of the amount of state that routers need to keep. Only routers on the distribution tree for a group keep forwarding state for that group, and no router needs to keep information about any source; thus CBT scales much better than flood-and-prune protocols, especially for sparse groups where only a small proportion of subnetworks have members.

PIM Sparse-Mode

The work on CBT encouraged others to try to improve on its limitations while keeping the good properties of shared trees, and *PIM Sparse-Mode*[7] was one result. The equivalent of a CBT Core is called a *Rendezvous Point* (RP) in PIM, but it largely serves the same purpose.

When a sender starts sending, whether it is a member or not, its local router receives the packets and maps the group address to the address of the RP. It then encapsulates each packet in another IP packet (imagine putting one letter inside another, differently addressed, envelope) and sends it unicast directly to the RP.

When a receiver joins the group, its local router initiates a Join message that travels hop-by-hop to the RP instantiating forwarding state for the group. However, this state is unidirectional state—it can be used only by packets flowing from the RP toward the receiver, and not for packets flowing back up the tree toward the RP. Data from senders is de-encapsulated at the RP and flows down the shared tree to all the receivers.

PIM-SM is an improvement on CBT in that discovery of senders and and tree building from senders to receivers are separate functions.

Thus PIM-SM unidirectional trees are not particularly good distribution trees, but they do start data flowing to the receivers. Once this data is flowing, the local router of a receiver can then initiate a transfer from the shared tree to a shortest-path tree by sending a source-specific Join message toward the source, as shown in Figure 3. When data starts to arrive along the shortest-path tree, a prune message can be sent back up the shared tree toward the source to avoid getting the traffic twice.

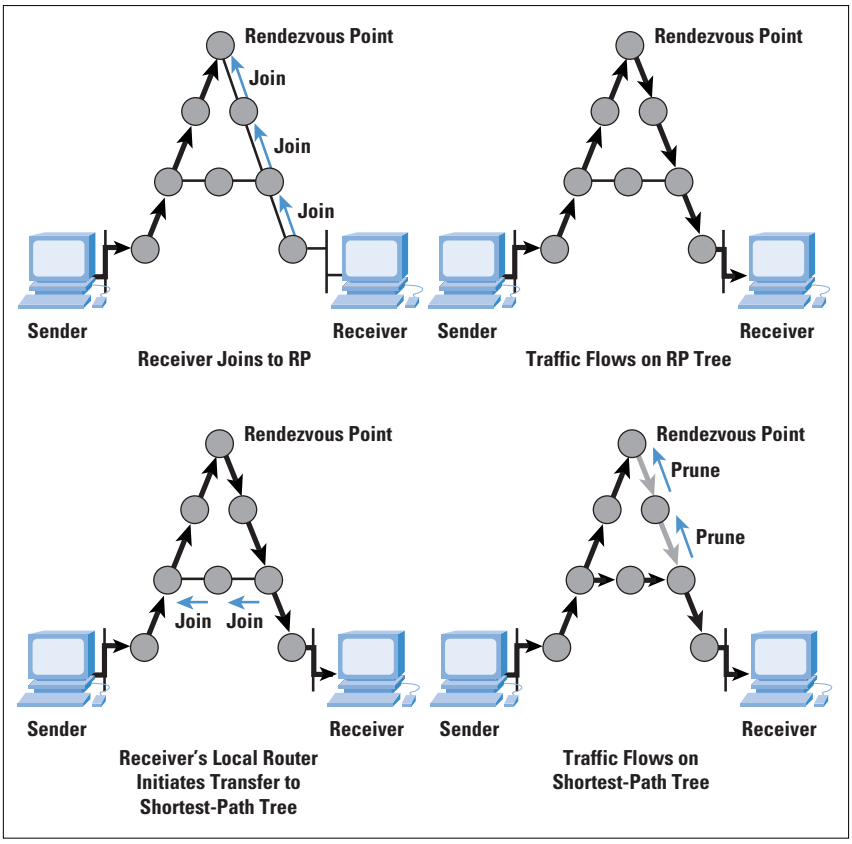


Figure 3: Formation of a PIM Sparse-Mode Tree

Unlike other shortest-path tree protocols such as DVMRP and PIM-DM, where prune state exists everywhere there are no receivers, with PIM-SM, source-specific state exists only on the shortest-path tree. Also, low-bandwidth sources such as those sending *Real-Time Control Protocol* (RTCP) receiver reports do not trigger the transfer to a shortest-path tree, a scenario that further helps scaling by eliminating unnecessary source-specific state.

Because PIM-SM can optimize its distribution trees after formation, it is less critically dependent on the RP location than CBT is on the Core location. Hence the primary requirement for choosing an RP is load balancing. To perform multicast-group-to-RP mapping, PIM-SM predistributes a list of candidates to be RPs to all routers. When a router needs to perform this mapping, it uses a special hash function to hash the group address into the list of candidate RPs to decide the actual RP to join.

Except in rare failure circumstances, all the routers within the domain will perform the same hash, and come up with the same choice of RP. The RP may or may not be in an optimal location, but this situation is offset by the ability to switch to a shortest-path tree.

The dependence on this hash function and the requirement to achieve convergence on a list of candidate RPs does, however, limit the scaling of PIM-SM. As a result, it is also best deployed within a domain, although the size of such a domain may be quite large.

Interdomain Multicast Routing

All the multicast routing schemes described so far suffer from scaling problems of one form or another:

- DVMRP and PIM-DM initially send data everywhere, and require routers to hold prune state to prevent this flooding from persisting.
- MOSPF requires all routers to know where all receivers are.
- PIM-SM needs predistribution of information about the set of RPs. Because traffic needs to flow to the RP, an RP cannot handle too many groups simultaneously, so many RPs are needed globally.

Thus each of these schemes is likely to be best deployed within a domain. How then does interdomain multicast routing take place?

Long-term solutions to this problem will be discussed in the second of these articles. In the meantime, the interim solution currently being deployed consists of multiprotocol extensions to the unicast *Border Gateway Protocol* (BGP) interdomain routing protocol, and a protocol called MSDP to glue PIM-SM domains together.

Multiprotocol BGP

For either technical or policy reasons, not all routers or peerings between Internet Service Providers (ISPs) are multicast capable. This situation complicates the use of PIM-SM for operation between domains because PIM assumes that the route obtained by unicast routing is good for multicast routing (strictly speaking, PIM assumes the reverse unicast path is good for forward-path multicast routing). If, in fact, the reverse unicast path is *not* good for forward-path multicast, then Join messages will often reach routers that do not support multicast, resulting in a lack of multicast connectivity. How then do we solve this problem?

BGP is the unicast interdomain routing protocol that is very widely used to connect unicast routing domains together. The multiprotocol extensions to BGP allow multiple routing tables to be maintained for different protocols. Thus with the *Multiprotocol Extensions for BGP-4* (MBGP)[2], you can build one routing table for unicast-capable routes and one for multicast-capable routes using the same protocol. PIM can then use the multicast-capable routes to forward Join messages and can, therefore, detour around parts of the network that don't support multicast.

Multicast Source Discovery Protocol

In addition to the problem of designing a scalable mechanism for mapping multicast groups to RPs, attempts to use PIM-SM as an interdomain protocol are hindered by ISPs' desire not to be dependent on other ISPs' facilities. For example, consider a multicast group consisting of senders and receivers in two domains, A and B, run by two different ISPs. If the RP is in domain A, and there is some problem in domain A, then senders and receivers in domain B might still be unable to communicate with each other using multicast, even though they are in the same domain, because initial PIM register messages must go via the RP. ISPs do not want to be dependent on other ISPs for connectivity within their own domain, so it appears that using PIM-SM as an interdomain protocol would be unacceptable, even if there were no scalability problems.

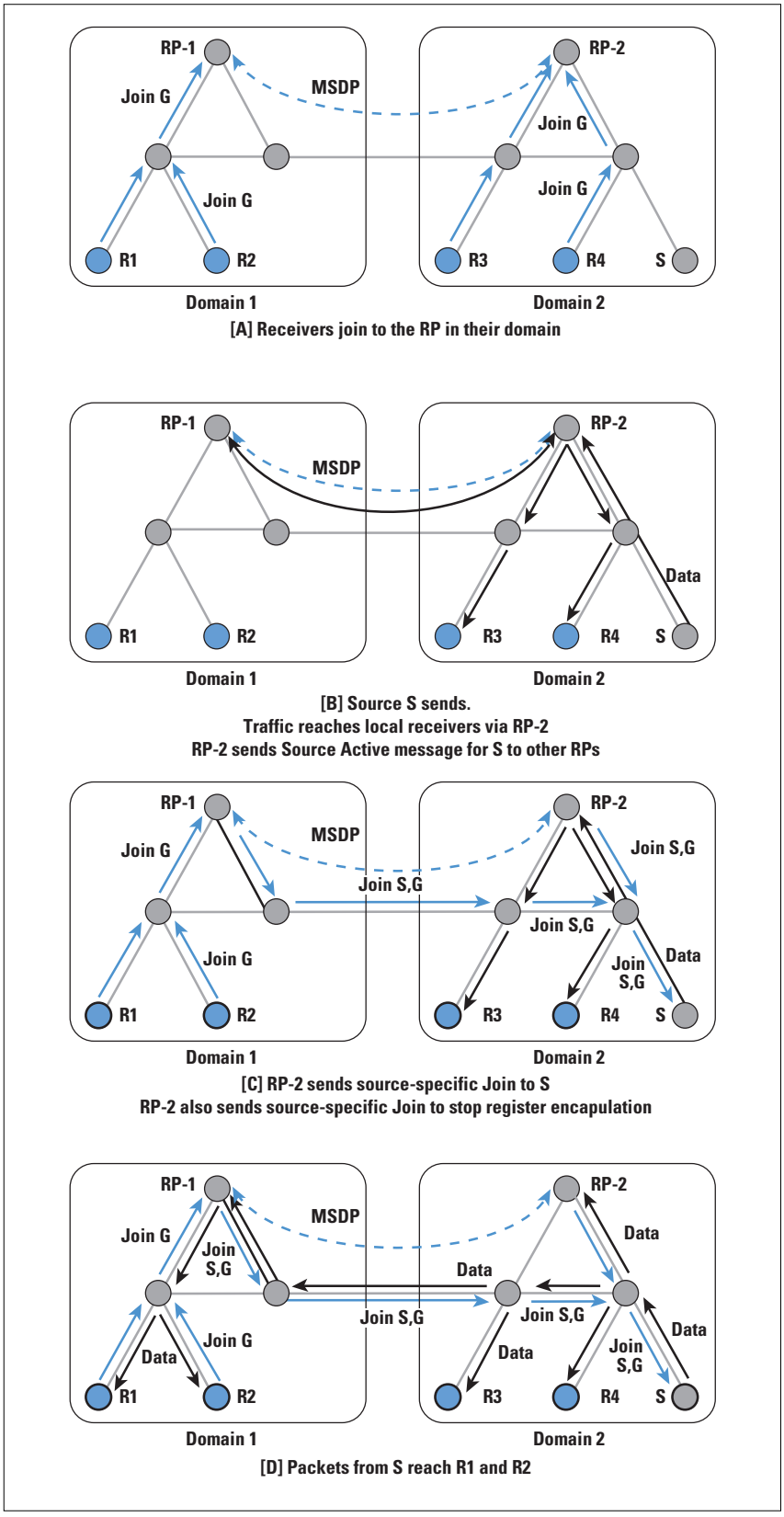
The *Multicast Source Discovery Protocol* (MSDP)[8] is an attempt to work around this problem. It does not provide a long-term scalable solution, but does provide a solution that solves the ISP interdependence problem.

With MSDP, ISPs run PIM-SM within their own domain, and they have their own set of RPs for all groups within that domain. Additionally, the RPs within the domain are interconnected with each other and with RPs in neighboring domains using MSDP control connections to form a loose mesh.

The process is shown in Figure 4. Within domain 1, R1 and R2 send Join messages from group G to RP-1. Similarly, R3 and R4 send Join messages to RP-2. When S starts sending, its packets are encapsulated to RP-2 by its local router in the normal PIM-SM manner. RP-2 decapsulates the packets and forwards them down the group-shared tree within domain 2 to reach R3 and R4. In addition, it sends a *Source Active* message over the MSDP mesh to all other RPs. RPs like RP-1 that have active joiners for this group then send a source-specific Join back across the interdomain boundary toward S. Traffic is then delivered interdomain following the source-specific state laid down by the Join messages, and it is eventually delivered to R1 and R2.

MSDP uses the normal PIM-SM source-specific join mechanism interdomain following the MBGP multicast routes back to the source, but it sets up only a group-shared tree within each domain, avoiding the need to depend on remote RPs in different domains for the delivery of traffic between local members in a domain.

Figure 4: MSDP in Operation



As an interdomain routing protocol, however, MSDP has many shortcomings. In particular, every RP in every domain must be told about every source that starts sending, and a significant subset of the RPs must cache all this information so that receivers that join late can cause source-specific Joins to be sent by their local RP. Thus MSDP does not scale well if there are a large number of senders worldwide.

In addition, to ensure that the first few packets sent by a source do not get lost, they must be encapsulated and sent alongside the *Source Active* message to all the RPs that might possibly have receivers. If they are not encapsulated, then sources that send only a few packets every few minutes might never get any data through to receivers because the source-specific state has timed out after each time they send.

In summary, MSDP is not a scalable long-term solution to interdomain multicast routing. However, it does solve a real short-term problem faced by ISPs, and so it is currently seeing significant deployment.

Multicast Address Allocation

A local protocol for requesting multicast addresses from multicast address allocation servers has recently been standardized. This protocol is called *Multicast Address Dynamic Client Allocation Protocol*, or MADCAP[10]. It is a relatively simple request-response protocol loosely modeled after the *Dynamic Host Configuration Protocol* (DHCP)[6].

MADCAP is intended to be used with interdomain protocols that perform dynamic allocation of parts of the multicast address space between domains, but because these protocols are not yet deployed, they will be discussed in the second of these articles.

As an interim solution for interdomain address allocation, a simple static mechanism has been defined. This mechanism involves embedding the *Autonomous System* (AS) number of the domain as the middle 16 bits of a multicast address. Thus the domain with AS number 16007 would get multicast addresses in the range 233.64.7.0 to 233.64.7.255 (64 and 7 being the upper and lower bytes, respectively, of 16007). Known as *glop addressing*, this mechanism is experimental. It may be superseded by a dynamic mechanism in the longer term.

Multicast Scoping

When applications operate in the global Multicast backbone (MBone), it is clear that not all groups should have global scope. Not only is this constraint especially important for performance reasons with flood and prune multicast routing protocols, but it also is true with other routing protocols for application security reasons and because multicast addresses are a scarce resource. Being able to constrain the scope of a session allows the same multicast address to be in use at more than one place as long as the scopes of the sessions do not overlap. This is analogous to the same radio frequency being used by two radio stations operating far apart from one another—each will only be heard locally.

Multicast scoping can currently be performed in two ways, known as *TTL Scoping* and *Administrative Scoping*. Currently TTL scoping is most widely used, with only a very few sites making use of administrative scoping.

TTL Scoping

When an IP packet is sent, an IP header field called *Time To Live* (TTL) is set to a value between zero and 255. Every time a router forwards the packet, it decrements the TTL field in the packet header, and if the value reaches zero, the packet is dropped. The IP specification also states that the TTL should be decremented if a packet is queued for more than a certain amount of time, but this decrement is rarely implemented these days. With unicast, the TTL is normally set to a fixed value by the sending host (64 and 255 are commonly used) and is intended to prevent packets from looping forever.

With IP multicast, the TTL field can be used to constrain how far a multicast packet can travel across the MBone by carefully choosing the value put into packets as they are sent. However, because the relationship between hop count and suitable scope regions is poor at best, the basic TTL mechanism is supplemented by configured thresholds on multicast tunnels and multicast-capable links. Where such a threshold is configured, the router will decrement the TTL, as with unicast packets, but then will drop the packet if the TTL is less than the configured threshold. When these thresholds are chosen consistently at all of the borders to a region, they allow a host within that region to send traffic with a TTL less than the threshold, and to know that the traffic will not escape that region.

An example is the multicast tunnels and links to and from Europe, which are all configured with a TTL threshold of 64. Any site within Europe that wishes to send traffic that does not escape Europe can send with a TTL of less than 64 and be sure that its traffic does not escape.

However, there are also likely to be thresholds configured within a particular scope zone—for example, most European countries use a threshold of 48 on international links within Europe, and because TTL is still decremented each time the packet is forwarded, it is good practice to send European traffic with a TTL of 63, a scenario that allows the packet to travel 15 hops before it would fail to cross a European international link.

Administrative Scoping

In some circumstances it is difficult to consistently choose TTL thresholds to perform the desired scoping. In particular, it is impossible to configure overlapping scope regions as shown in Figure 5, and TTL scoping has numerous other problems, so more recently, administrative scoping has been added to the multicast forwarding code in *mrouted* and in most router implementations.

Administrative scoping allows the configuration of a boundary by specifying a range of multicast addresses that will not be forwarded across that boundary in either direction.

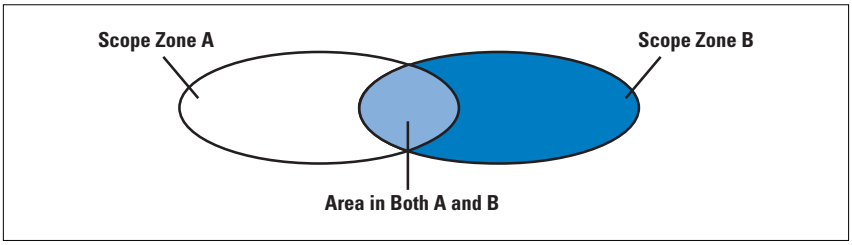


Figure 5: Overlapping Scope Zones possible with Administrative Scoping

Scoping Deployment

Administrative scoping is much more flexible than TTL scoping, but it has many disadvantages. In particular, it is not possible to tell from the address of a packet where it will go unless all the scope zones that the sender is within are known. Also, because administrative boundaries are bidirectional, one scope zone nested within or overlapping another must have totally separate address ranges. This makes address allocation difficult from an administrative point of view, because the ranges ought to be allocated on a top-down basis (largest zone first) in a network where there is no appropriate top-level allocation authority. Finally, it is easy to misconfigure a boundary by omitting or incorrectly configuring one of the routers. With TTL scoping it is likely that in many cases a more distant threshold will perform a similar task, lessening the consequences, but with administrative scoping, there is less likelihood that this scenario will occur.

For these reasons, administrative scoping has been viewed by many network administrators as a speciality solution to difficult configuration problems, rather than as a replacement for TTL scoping, and the Mbone still very much relies on TTL scoping. However, this situation is set to change as a protocol for automatically discovering scope zones (and scope zone misconfigurations) starts to be deployed. This protocol is called the *Multicast Zone Announcement Protocol* (MZAP)[9], and it will shortly become an IETF Proposed Standard. Eventually the use of configured TTL scopes to restrict traffic will cease to be used as a primary scoping mechanism.

Summary

In this article we have looked at the various routing systems that are used to devise delivery trees over which multimedia data can be sent for the purposes of group communication, and at address allocation and scoping mechanisms for this traffic.

After ten years of experimentation, IP multicast is not currently a ubiquitous service on the public Internet, but significant deployment has taken place on private intranets. The existing multicast routing and address allocation mechanisms work well at the scale of domains. However, as we have seen, there are still significant technical problems

concerning scaling to be overcome before multicast can be a ubiquitous interdomain service. In addition to the routing problems, we also still lack deployed congestion control mechanisms for multicast traffic, which are essential if multicast applications are to be safely deployed.

Despite these issues, IP multicast still shows great promise for many applications. Solutions have been devised to many of the remaining problems, although they have not yet been deployed. In the second of these articles, we will look at the proposed solutions for scalable interdomain routing and address allocation. We will also touch on multicast congestion control and the solutions that are currently emerging from the research community.

Document Status

A list of IETF specifications for the protocols discussed in this article is given below. We include the status for each document as of this writing (November 1999). For more information, check the IETF Web pages at **`www.ietf.org`**

Document	Status
IGMP v1	IETF Standard (RFC 1112)
IGMP v2	IETF Proposed Standard (RFC 2236)
IGMP v3	IETF work in progress
DVMRP	IETF Experimental Standard (RFC 1075)
PIM-Dense Mode	IETF work in progress
Multicast OSPF	IETF Proposed Standard (RFC 1584)
Core Based Trees	IETF Experimental Standard (RFC 2201)
PIM Sparse-Mode	IETF Experimental Standard (RFC 2362)
Multiprotocol BGP	IETF Proposed Standard (RFC 2283)
MSDP	IETF work in progress
MADCAP	IETF Proposed Standard (RFC 2730)
Glop Addressing	IETF work in progress

References

[1] Ballardie, A., "Core Based Trees (CBT version 2) Multicast Routing," RFC 2189, September 1997.

[2] Bates, T., Chandra, R., Katz, D., and Rekhter, Y., "Multiprotocol Extensions for BGP-4," RFC 2283, February 1998.

[3] Deering, S., "Host Extensions for IP Multicasting," RFC 1112, August 1989.

[4] Deering, S., Partridge, C., and Waitzman, D., "Distance Vector Multicast Routing Protocol," RFC 1075, November 1988.

[5] Deering, S., Estrin, D., Farinacci, D., Jacobson, V., Helmy, A., Meyer, D., and Wei, L., "Protocol Independent Multicast Version 2 Dense Mode Specification," Internet Draft, work in progress.

[6] Droms, R., "Dynamic Host Configuration Protocol," RFC 1531, October 1993.

[7] Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., Liu, C., Sharma, P., and Wei, L., "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification," RFC 2362, June 1998.

[8] Farinacci D. et al. "Multicast Source Discovery Protocol (MSDP)," Internet Draft, work in progress, June 1998.

[9] Handley, M., Thaler, D., and Kermode, R., "Multicast-Scope Zone Announcement Protocol (MZAP)," Internet Draft, work in progress.

[10] Hanna, S., Patel, M., and Shah, M., "Multicast Address Dynamic Client Allocation Protocol (MADCAP)," RFC 2730, December 1999.

[11] Moy, J., "OSPF Version 2," RFC 2328, April 1998.

[12] Moy, J., "Multicast Extensions to OSPF," RFC 1584, March 1994.

[13] Miller, C. K., "Reliable Multicast Protocols and Applications," *The Internet Protocol Journal,* Volume 1, No. 2, September 1998.

JON CROWCROFT is a professor of networked systems in the Department of Computer Science, University College London, where he is responsible for a number of European and U.S. funded research projects in Multi-media Communications. He has been working in these areas for over 18 years. He graduated in Physics from Trinity College, Cambridge University, in 1979, and gained his MSc in Computing in 1981, and PhD in 1993. He is a member of the ACM, the British Computer Society, and is a Fellow of the IEE and a senior member of the IEEE. He is a member of the Internet Architecture Board (IAB) and was general chair for the ACM SIGCOMM from 1995 to 1999. He is also on the editorial team for the ACM/IEEE *Transactions on Networks*. With Mark Handley, he is the co-author of *WWW: Beneath the Surf* (UCL Press); he also authored *Open Distributed Systems* (UCL Press/Artech House), and with Mark Handley and Ian Wakeman, a third book, *Internetworking Multimedia* (Morgan Kaufmann Publishers), published in October 1999.
E-mail: **J.Crowcroft@cs.ucl.ac.uk**

MARK HANDLEY received his BSc in Computer Science with Electrical Engineering from University College London in 1988 and his PhD from UCL in 1997. For his PhD he studied multicast-based multimedia conferencing systems, and was technical director of the European Union funded MICE and MERCI multimedia conferencing projects. After two years working for the University of Southern California's Information Sciences Institute, he moved to Berkeley to join the new AT&T Center for Internet Research at ICSI (ACIRI). Most of his work is in the areas of scalable multimedia conferencing systems, reliable multicast protocols, multicast routing and address allocation, and network simulation and visualisation. He is co-chair of the IETF Multiparty Multimedia Session Control working group and the IRTF Reliable Multicast Research Group.
E-mail: **mjh@aciri.org**

[This article is based in part on material in *Internetworking Multimedia* by Jon Crowcroft, Mark Handley, and Ian Wakeman, ISBN 1-55860-584-3, published by Morgan Kaufmann in 1999. Used with permission].

The Internet2 Project

by Larry Dunn, Cisco Systems

Communication, connectivity, education, entertainment, e-commerce—across a broad spectrum of activities, the commodity Internet has made a strong impact on the way we live, work, and play. Nevertheless, many classes of applications do not yet run well, and some don't run at all, over the commodity net. As new applications are developed in disciplines from medicine to engineering to the arts and sciences, their success increasingly depends on an ability to use networks effectively. In research and education collaborations all over the world, efforts are under way to make use of new network technologies and develop network services that will facilitate these advanced applications. One such effort in the United States is called the *Internet2* Project[1].

The Internet2 Project was started in 1996 by 34 U.S. research universities. It has since grown to over 140 universities, and includes several corporate members and international partners. This article examines network technology used in Internet2, and looks at some of the engineering challenges involved in facilitating applications being developed by Internet2 members.

Background

In 1995, the U.S. National Science Foundation (NSF) funded a program to create the *very-high-performance Backbone Network Service* (vBNS)[2]. The NSF provided funding to MCI, who interconnected five U.S. supercomputer centers and 3 *Network Access Points* (NAPs), where it was envisioned that supercomputer clients and other vBNS users would connect.

By 1996, congestion stemming from academic traffic to the commodity Internet had seriously congested the NAPs; it was accordingly recognized that clients of the supercomputer centers might be better served if the Research Universities, where Principal Investigators often resided, were themselves *directly* connected to the vBNS. So in 1996, the NSF accepted proposals as part of the *High-Performance Connections* (HPC) program[3]. Schools applying for an HPC grant might receive $350,000 over a 2-year period, provided their proposals met various criteria, including meritorious research that would benefit from the high-performance connection, a solid network plan, intention to investigate capabilities enabled by such a connection, commitment to share results with the community, matching funds from the University, and so on.

In October 1996, representatives from 34 universities met, and concluded that, while not all the schools had projects involving "meritorious research" that would meet the NSF criteria, they all *did* have a critical interest in deploying the kind of applications that such high-performance connections could enable.

Thus, to facilitate development and deployment of applications that would further the research and education mission of member universities, the Internet2 Project was formed.

From the beginning, the stated intention was to enable applications that could not run, or could not run well, on the "Commodity Internet." Networks would be utilized or constructed only so as to facilitate this applications-enabling goal, and results/methods would be applied to the broader community as rapidly as possible.

Applications Focus

The list of applications being used or developed by Internet2 members is extensive. Several fall in the category of "meritorious research" as mentioned in the NSF HPC criteria. Examples include: remote instrument control (for instance, telescopes, microscopes), high-performance distributed computation, and large-scale database navigation. Other applications that further the education mission of member universities include tools to facilitate multisite collaboration, and asynchronous learning. Many examples in areas from science, engineering, art, language, music, and more can be found at the Internet2 applications Web site[4]. In addition to individual applications, a couple of broad initiatives have a relationship with Internet2, including *The Internet2 Digital Video Initiative,* housed at the *International Center for Advanced Internet Research* (iCAIR)[5], and the *Internet2 Distributed Storage Infrastructure Initiative* (I2-DSI)[6].

The above applications share several challenging requirements, many of which translate to resource commitments that must be met by the network in an end-to-end fashion, including bandwidth and jitter. Additionally, the applications can become scalable only if more-mature middleware and control-plane infrastructure is developed. Necessary components include features such as *Authentication, Authorization, and Accounting* (AAA), scheduling, and coordination of resources managed by multiple administrative domains.

One compelling example of the network challenges present in a virtual collaborative environment is exemplified by a CAVE (*Cave Automated VR Environment*). See [7] for more details, but in brief, a single CAVE is a (10 x 10 x 10)-foot cube, with one wall removed. Users enter through the open wall, and using lightweight stereo-three-dimensional (3D) glasses, and a radio frequency (RF) mouse, can interact with an immersive environment created by rear-screen and direct projection on multiple walls and the floor. As an example, the interconnection of multiple CAVEs allows design teams in remote locations to jointly experience the operating "feel" of a new vehicle, and to dynamically adjust, design, or control parameters to see how the modified vehicle behaves.

The developers of CAVE software at Argonne National Labs have noted that the data flows in a CAVE consist of at least: control, text, audio, video, tracking, database, simulation, haptic, and rendering flows. Additionally, they have estimated the latency, jitter, and bandwidth requirements for these flows. Some of the flows represent a challenge in a single resource dimension, others have strict requirements in multiple resource dimensions.

Backbone Networks

At this time, Internet2 members may connect to either of two backbone networks, or both.

The vBNS is operated by MCI/Worldcom. It consists primarily of an IP-over-ATM network. Most schools connect at DS3 or OC-3c via ATM to a vBNS ATM switch. Interior vBNS links are OC-12c ATM. The schools peer with a Layer 3 router; a router is attached to each of the vBNS ATM switches. The vBNS routers are logically connected to each other via a full mesh of *Unspecified Bit Rate* (UBR) *Virtual Circuits* (VCs). The ATM switches are connected to each other via a second layer of ATM switches, which are part of MCI's commercial Hyperstream offering. While schools pass the vast majority of their traffic via peering at Layer 3 with the nearest vBNS border router, other services are available, including the option to establish *Variable Bit Rate* (VBR) VCs as needed, and the possibility to place some of the ATM-attached hosts of the school directly in a vBNS Classical IP *Logical IP Subnet* (LIS). This setup allows such hosts to send bytes directly to other ATM-attached hosts, bypassing the routers of both the school and the vBNS. The vBNS also carries native IP multicast traffic among members. In addition, the vBNS has a native IPv6 offering, which is achieved by deploying routers that run IPv6, and provisioning VCs to schools also running IPv6. The vBNS has also begun to offer an *IP-over-Synchronous Optical Network* (SONET) service, the first instance of which is an OC-48 *Packet-over-SONET* (POS) link from Northern to Southern California. Because the nominal partnership arrangement with the NSF expires in the year 2000, the vBNS has established a new network offering [called *Next Generation Network* (NGN)], to which schools and other entities may connect if the vBNS/NSF partnership is not renewed.

Measurement Tools in vBNS

One of the outcomes of the vBNS program has been the development of a variety of high-performance measurement tools. One such tool, called *OC-3mon* (and now, *OC-xMon*), was developed to allow passive capture (using optical splitters) of ATM cell and IP header information, to facilitate high-speed flow characterization. More detail is available at the vBNS Web site[2]. Recently, further development of OC-xMon has been undertaken by the *Cooperative Association for Internet Data Analysis* (CAIDA)[8]. CAIDA has perhaps the best collection of high-performance public-domain measurement and analysis tools in the world, and its Web site is definitely worth browsing.

The second backbone network to which Internet2 members can connect is called *Abilene*[9]. Abilene was constructed by the *University Corporation for Advanced Internet Development* (UCAID) in collaboration with three industrial partners and Indiana University (IU). Partner contributions include fiber capacity from Qwest, SONET gear from Nortel, and routers from Cisco. The Abilene Network Operations Center (NOC) is staffed and operated by Indiana University. The network uses OC-48c POS interior links that initially connect ten routers in a partial mesh (a few interior links started as OC-12c, but are being upgraded). Abilene participants can connect at OC-3c or OC-12c, using either POS or ATM. See [10] for details on the router hardware architecture. For an insightful look at a research project that shows how this architecture can scale, see the second link in[10] and also see Stanford Professor Nick McKeown's Tiny Tera homepage at[11].

Measurement Tools in Abilene

It's worth spending a bit of time at the Abilene NOC Web site[12]. One of the interesting tools developed there is the "Abilene Weather Map"[13]. Abilene NOC has indicated that it will make source code for this tool available to Internet2 schools.

Gigapop Technology Survey

Internet2 schools can connect to either vBNS or Abilene directly. However, it is also common for several schools to converge their links at a "gigapop." This gigapop then connects to Abilene and/or vBNS, and possibly to commodity Internet Service Providers (ISPs) (to carry the "Commodity Internet" traffic of the school). Additionally, non-Internet2 schools, libraries, K-12, and state government networks also often converge at gigapops. Non-Internet2 schools typically don't forward traffic over Abilene or vBNS. But the common meeting point allows local exchange of local traffic, often affords larger aggregate commodity Internet connectivity for the gigapop participants, and allows direct access to other services that might be offered at the gigapop (Web caching, and so on).

The connectivity architecture used at gigapops varies widely. Detailed documentation for several gigapops can be found at[14]. Some Gigapops are "Layer 2," meaning that each participant is responsible for exchanging routes and traffic among themselves directly. More often, gigapops are "Layer 3," meaning that the gigapop provides a router with which gigapop participants peer. The gigapop router then typically exchanges traffic with vBNS and/or Abilene, and possible commodity ISPs.

Some gigapops are implemented at a single site (for instance, *Metropolitan Research and Education Network* [MREN], *Southern Crossroads* [SoX]), while others are "distributed gigapops," meaning gigapop equipment exists at multiple locations (for instance, the *California Research and Educational Network* [CalREN-2], and *The Great Plains Network* [GPN]). Following are a couple of specific gigapop examples.

MREN

The MREN[15] is built on a Layer 2 gigapop near Chicago that joins schools and research facilities from Illinois and several states in the Midwest. MREN members typically connect with OC-3c ATM links. Since MREN is a Layer 2 gigapop, the border router of each member peers directly with the border routers of other members. Additionally, each member's border router might peer with the Chicago-area vBNS or Abilene border router. vBNS and Abilene routers (as well as several other national research and international networks) peer here. Physically, the facility is built upon the Network Access Point (NAP) facility provided by Ameritech Advanced Data Services (AADS)[16]. Routers typically peer with each other via ATM UBR *Permanent Virtual Paths* (PVPs), although other arrangements are possible.

CENIC/CalREN-2

The Corporation for Education Network Initiatives in California (CENIC)[17] has constructed CalREN-2. The CalREN-2 distributed gigapop is interesting in several respects. First, as the name implies, it represents a distributed gigapop. In this case, three separate SONET ring facilities provide connectivity for Northern, Central (Los Angeles area), and Southern California schools. These three regions are linked to each other, and also to external networks.

Second, in each ring, there are two sets of OC-12c connections to each adjacent school. CalREN-2 has currently utilized these connections to construct both a ring of ATM connectivity, and a separate, parallel ring of POS connectivity. As a result, CalREN-2 is uniquely positioned to experiment simultaneously with both ATM and POS connectivity, performance, and QoS characteristics.

Third, to take the Northern schools as an example, the ring structure allows for a variety of Layer 3 topologies to be explored. For example, in a ring with these size and bandwidth characteristics, what are the tradeoffs on application-level performance of inducing more hops while keeping the per-hop bandwidth high, versus dividing the bandwidth into smaller slices but creating a partial mesh that reduces the average Layer 3 hop count?

Engineering Challenges

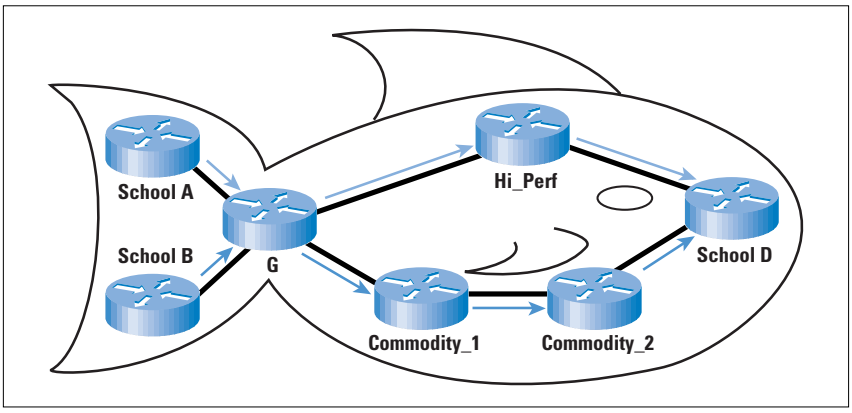
This section looks at some of the engineering challenges present in Internet2. They revolve around enabling applications with new network services, implementing appropriate policy, and doing all of this at high speed. Specifically, we'll look at Explicit Routing, Multicast, and Quality of Service.

Explicit Routing—The Fish Problem

The condition that several schools often converge at a gigapop, combined with the constraint that sometimes the funders of high-performance connectivity require that only the funded schools are allowed to use the high-performance connection, gives rise to a need for

"explicit routing" at the gigapop. The gigapop can forward packets through either a high-performance connection, or through the commodity Internet. Usually, for a single destination, traditional routing would have the gigapop use the "best" path to forward all packets to a particular destination. But when multiple policies must be implemented at the gigapop, the gigapop router must be able to "override" normal routing and forward packets on a path that's not the "best." A concrete example is shown in Figure 1.

Figure 1: The "Fish Problem"



Consider packets from schools A and B, both headed for destination D. Assume both schools are connected to gigapop G, and that G has two paths to D; one along *G-Hi_perf-D*, and the other along *G-Commodity_1-Commodity_2-D*. Further assume that A is allowed to use either path (and would perfer *G-Hi_perf-D*), but that B is prohibited from using *G-Hi_perf-D*. This scenario describes the "Explicit Routing Problem," and since it is often drawn in a shape resembling a fish, is also known as the "Fish Problem." The essence is that a routing decision at G must be made on some other criteria than just the destination IP address.

A couple of solutions to the fish problem have been used in the past, but they tend to have problems with either speed or scalability. For example, "policy routing," which usually includes a method to look at both source and destination address, has historically shown low performance. Inserting ATM switches and using virtual circuits has been used in some cases, but this solution has scaling problems and requires extra equipment. Today, many Internet2 gigapops use a separate router per policy. In the case of needing two policies in the example above, this means two routers. This solution is expensive, but does have high performance.

One promising idea is to implement enough of the "policy routing" process in hardware to allow high-speed *source+destination+other_bits* lookups. While straightforward in concept, some point out that even with line-rate source-address routing capability, the method is flawed because it requires significant manual configuration, and is prone to creating black holes for traffic upon link failure. Proponents suggest that these shortcomings can be overcome.

Another promising mechanism is becoming available as a result of work done to facilitate *Multiprotocol Label Switching* (MPLS) in routers and switches. The idea here is that one of the underlying pieces of technology required for MPLS is "multi-FIB" (multiple *Forwarding Information Bases*). Instead of the traditional "single-FIB," which always uses "the best" route to a destination, multi-FIB allows multiple forwarding tables to exist in a single router. This setup will allow a gigapop to implement multiple policies in one router, rather than the "one box per policy" that several gigapops have used previously. Note that in the case of a gigapop with a single router on which all members converge, multiple policies can be achieved with multi-FIB without actually using MPLS *Label-Switched Paths* (LSPs). For more complex gigapops, where members themselves may converge high-performance-eligible and ineligible traffic before forwarding on a single link to the gigapop, one might consider using simple LSPs to present the gigapop with traffic that is predifferentiated.

Multicast

Many of the applications in Internet2 schools use multicast. In addition to flows for videoconferencing or distance learning that use MPEG-1 (or slower) rates, a wide variety of applications require high-performance, scalable multicast. Examples include high-resolution immersive environments, collaborative real-time medical image diagnosis, and high-fidelity conferencing or distance learning (for instant, digital video camera rates of 30 mbps). When the Internet2 project began, many schools were on the *Multicast Backbone* (Mbone), and used *Distance Vector Multicast Routing Protocol* (DVMRP) tunnels to participate in multicast. Over the past year, one of the strong areas of collaboration between the Internet2 schools and the vendor community has been to develop and implement a migration strategy that allows Internet2 backbones, gigapops, and schools to move toward high-performance, scalable, native multicast support.

At the Internet2 conference in San Francisco in September 1998, the vBNS backbone was exposed to unprecedented levels of multicast stress. In a somewhat painful, but worthwhile, learning experience, it was concluded that *Protocol Independent Multicast-Dense Mode* (PIM-DM) did not scale well in highly meshed, high bitrate backbones. As a result, the vBNS has shifted to *PIM-Sparse Mode* (PIM-SM), and Abilene is being constructed with PIM-SM.

The current set of multicast components being applied in Internet2 (and leading ISPs) include: PIM-SM, *Multicast Border Gateway Protocol* (MBGP), and the *Multicast Source Discovery Protocol* (MSDP). MBGP allows distribution of routing information such that unicast and multicast routing can use noncongruent topologies.

MSDP allows independent domains to exchange information about multicast sources without creating interdomain Rendezvous Point (RP) dependencies. As they become standardized, it is expected that the *Border Gateway Multicast Protocol* (BGMP) and *Multicast Address Set Claim* (MASC) will be added to this infrastructure set.

Quality of Service

An area of broad interest in the Internet2 community centers on *Quality of Service* (QoS). The heart of QoS involves establishing strategies through which applications can be assured access to appropriate network resources when required. Typical examples of resources include end-to-end bandwidth, latency, or jitter. Of course, collateral issues and dimensions abound, including end-to-end vs. segment-only QoS; signaled vs. static provisioning; amount of state required by various approaches; level of granularity, precision, and strength of QoS "guarantee;" AAA issues; and reliability and recovery dynamics.

In an effort to start small, but make concrete progress, the Internet2 QoS working group[18] has launched an experiment called the *Qbone*[19]. Participants include backbone networks, gigapops, and individual schools and research labs worldwide. The Qbone will focus on deploying and using components developed by the Internet Engineering Task Force's (IETF) *Differentiated Services* working group (Diffserv)[20].

The initial Qbone plan is to deploy an approximation to the Expedited Forwarding (EF)[21] forwarding behavior. The Qbone will start by statically allocating a small amount of EF bandwidth across boundaries between Autonomous Systems (ASs) to allow small EF flows among arbitrary combinations of schools/labs. Large flows, in these early stages, will have to be handled manually (much as they are today). In later stages the plan is to use *Bandwidth Brokers* (BBs) currently under development[22] to aid in the automation of adjusting resource commitments between ASs (using interdomain BBs), and to aid in accepting application resource requests (using intradomain BBs, combined with policy servers and AAA mechanisms). The precise mechanics for BB interaction, trade-offs among signaling frequency, amount of state, scalability, and so on are certainly topics of research, but that's part of what makes Qbone participation fun!

Summary

There is no single application or technology that makes Internet2 unique or exciting. Rather, the effort required to enable new applications that have strong bandwidth, latency, jitter, and coordination requirements has resulted in an infusion of energy from a variety of disciplines. Internet2 requires stretching existing technologies (ATM, POS, multicast, measurement), nurturing developing technologies (Quality of Service, explicit routing, Dense Wave-Division Multiplexing [DWDM], mobility), and participating in the invention of new technologies (all-optical infrastructures, extending AAA, and other resource allocation and

scheduling middleware). Internet2 requires attention to maturing components in backbone, gigapop, and campus environments in order to deliver on the promise of speedy transference of lessons learned to the commodity Internet. The effort so far has resulted in demonstration of truly stunning, impactful, and useful applications. It is the convergence of effort and rapid rate of change that makes Internet2 a challenging and rewarding endeavor.

Other Initiatives

Although this article has focused on aspects of Internet2 in the United States, there are many advanced Internet activities around the world. A partial list includes:

http://www.dante.net/ten-155.html (Europe)
http://www.ukerna.ac.uk (UK)
http://www.dfn.de (Germany)
http://www.renater.fr (France)
http://www.surfnet.nl (The Netherlands)
http://apan.or.kr (Asia/Pacific)
http://www.singaren.net.sg (Singapore)
http://www.canet3.net (Canada)
http://www.cudi.edu.mx (Mexico)
http://www.reuna.cl (Chile)
http://www.ngi.gov (U.S. Federal)
http://www.startap.net (International peering)

A more complete list of advanced Internet initiatives is maintained at:
http://www.cisco.com/aii

References

[1] **http://www.internet2.edu**

[2] **http://www.vbns.net**

[3] See latest press release at:
http://www.nsf.gov/od/lpa/news/press/99/pr9915.htm
...and updated program announcement at:
http://www.nsf.gov/pubs/1998/nsf98102/nsf98102.txt

[4] **http://apps.internet2.edu**

[5] **http://i2dv.nwu.icair.org/** and **http://www.icair.org/**

[6] **http://dsi.internet2.edu/**

[7] **http://evlweb.eecs.uic.edu/pape/CAVE**
...has a great introduction to CAVE technology.
Also see the *Electronic Visualization Laboratory* homepage at:
http://www.evl.uic.edu/EVL/index.html

[8] **http://www.caida.org**

[9] **http://www.ucaid.org**, and Abilene specifics at:
http://www.internet2.edu/abilene

Abilene router details are at:
[10] **http://www.cisco.com/warp/public/cc/cisco/mkt/core/12000/index.shtml**
...and Nick McKeown's paper is at:
http://www.cisco.com/warp/public/cc/cisco/mkt/core/12000/tech/fasts_wp.pdf

[11] **http://tiny-tera.stanford.edu/tiny-tera/index.html**

[12] **http://www.abilene.iu.edu**

[13] **http://hydra.uits.iu.edu/~abilene/traffic**

[14] Following are several gigapop sites:
California's CENIC/CalREN2: **http://www.cenic.org,**
The Pacific/Northwest gigapop: **http://www.pnw-gigapop.net**
The Great Plains Network: **http://www.greatplains.net**
The Southern Crossroads, with members from Southeastern Universities Research Association: **http://www.sox.net**
MidAtlantic Crossroads: **http://www.networkvirginia.net/MAX**
MREN: **http://www.mren.org**
WestNet: **http://www.scd.ucar.edu/nets/Projects/Westnet**
North Carolina Gigapop: **http://www.ncni.net**
The Texas Gigapop: **http://noc.gigapop.gen.tx.us**
Northern Crossroads: **http://www.nox.org**
Philadelphia area Magpi: **http://www.magpi.net**
Pittsburgh-based NCNE: **http://www.ncne.net**
New York: **http://www.nysernet.org**

[15] **http://www.mren.org**

[16] **http://www.aads.net,** and **http://nap.aads.net/main.html**

[17] **http://www.cenic.org**

[18] **http://www.internet2.edu/qos/wg**

[19] **http://www.internet2.edu/qos/qbone**

[20] **http://www.ietf.org/html.charters/diffserv-charter.html**

[21] **http://www.ietf.org/rfc/rfc2598.txt**

[22] **http://www.merit.edu/working.groups/i2-qbone-bb**

LARRY DUNN is the Technology Development Manager in the Advanced Internet Initiatives Division at Cisco Systems. He serves on the Internet2 Quality of Service and Routing working groups. After receiving his PhD from the University of Minnesota (Electrical Engineering '92), he served as Director of Networking there, and subsequently as Director of Strategic Markets and Applications (Education) for FORE Systems. He periodically teaches Advanced Networking courses at the University of Minnesota. Research interests include test vector generation for combinational logic, network design and analysis, and Quality of Service techniques and deployment strategies.
E-mail: **ldunn@cisco.com**

One Byte at a Time: Internet Addressing

by Peter H. Salus

The source of all knowledge where the Internet is concerned is the set of *Requests for Comments* (RFCs). Because there are now well over 2,700 RFCs, however, only a few people track history, evolution, and outright paradigm shift.

Each node on the Internet—router or end system (often called "host" or "server")—has a unique identifier attached to it; this identifier is its *address*. Any packet sent between nodes must use the destination address to tell the intervening routers where it should go.

In RFC 1 (April 1969), Steve Crocker laid out a scheme that allotted five bits to address space: enough for 32 addresses. By September 1969, when *Interface Message Processor* (IMP) No. 1 was installed in Kleinrock's lab at UCLA, this number had grown to six bits (63 addresses). By 1972, it had become apparent that this number would be insufficient, and the address space was enlarged to eight bits (255 addresses). In fact, the *Advanced Research Projects Agency Network* (ARPANET) hit only 63 hosts in January 1976. This number was, however, already a lot in terms of the `HOSTS.TXT` tables that were distributed to every site. By August 1983, there were 213 hosts, and the eight-bit address barrier was being pushed.

Cerf's original version of TCP (RFC 675; December 1974) and Postel's of IP (RFC 760; January 1980) increased this "address space" to 32 bits, but the structure of the ARPANET was "flat," that is, the hierarchical distributed name-to-address database we are familiar with only came about with Mills' conceptualization of the *Domain Name System* (DNS) (RFC 799; September 1981), and its implementation by Paul Mockapetris (RFCs 882 and 883; November 1983).

Address Classes

The Internet Protocol uses a 32-bit addressing scheme and originally four classes of networks: A, B, C, D. (See Figure 1 on page 5). There are only 128 Class A networks, but each can have 16,777,216 unique host identifiers. Next, there are 16,384 Class B networks, with 65,535 unique identifiers; 2,097,192 Class C networks, with 255 hosts; and over 268 million Class D multicast groups. (A fifth class, Class E, is reserved and not available for general use).

Address Depletion

Using the 32-bit IP addressing scheme allowed for about 4 billion hosts on 16.7 million networks. Although this number of various kinds of addresses seemed like a lot, the expansion of the use of the Internet over the past decade has been explosive, and the original address classes did not allow for a flexible address assignment based on an organization's particular need.

In August 1990 during the Vancouver *Internet Engineering Task Force* (IETF) meeting, Frank Solensky, Phill Gross, and Sue Hares projected that the current rate of assignment would exhaust the Class B space by March 1994.

CIDR

Classless Inter-Domain Routing or CIDR (RFCs 1518 and 1519; September 1993) was introduced to improve both routing scalability and address space utilization in the Internet. By eliminating the notion of "network classes," CIDR allows for a better match between address requirements and address allocation. This results in expansion of the scope of hierarchical routing, which in turn improves scaling properties of the Internet routing system. CIDR has proven to be the palliative that has enabled the Internet to continue functioning while growth continues.

Even with this palliative, it was predicted in 1994 that, using the current allocation statistics, the Internet will exhaust the IPv4 address space between 2005 and 2011. With five more years of experience, which has also brought greater uncertainty as to gross numbers, we can push these dates out a bit, but exhaustion will come eventually.

Another factor that has slowed down the address depletion rate is the use of *Network Address Translation* (NAT). NAT devices allows an organization to have one external ("public") address and many private (net 10 is often used) addresses internally. Since the internal addresses are not "seen" from the outside, they do not need to be globally unique. This approach has downsides (some protocols weren't designed with NATs in mind), but from the address depletion point of view, it is a win. RFC 1597 describes "Address Allocation for Private Internets."

If you are interested in current Internet addressing, an excellent book is available: *TCP/IP Addressing,* by Buck Graham, AP Professional, 1997. Graham does an excellent job on addressing, routing, and the various bizarries involved in optimal routing, efficient use of address space, and making network management less onerous. This book is, however, not intended to be for elementary instruction; Graham primarily speaks to the professional market.

IPng aka IPv6

In the summer of 1994, the IETF set up an Internet Protocol next generation (IPng) task force, cochaired by Scott Bradner and Allison Mankin. (IPng later became known as IPv6 for "IP version 6"). Recommendations from that task force were released in October 1994 for discussion at the December 1994 IETF meeting. The basic goal was to have something in place before 2000, so that the time limit would not be pushed.

Unfortunately, as Bradner and Mankin stated in their recommendation: "Some people pointed out that this type of projection makes an assumption of no paradigm shifts in IP usage. If someone were to develop a new 'killer application,' (for example, cable-TV set top boxes), the resultant rise in the demand for IP addresses could make this an over-estimate of the time available."

IPv6 provides for 128-bit addressing. This number is gigantic: larger than the estimated total number of molecules in the universe.

Books

Two noteworthy books are available on IPv6 itself: Christian Huitema's *IPv6: The New Internet Protocol* (ISBN 0-13-241936-X, Prentice Hall, 1996) and Scott Bradner and Allison Mankin's anthology *IPng* (ISBN 0-201-63395-7, Addison-Wesley, 1996), which provides an explanation of the task force's process and explicates the services that are provided for (as, for example, ATM support). These books are both dated, but they are the best available now. Keeping up with what's going on is easy, thanks to the IETF's Web site **`http://www.ietf.org`**.

An excellent business and technical case for IPv6 is found in the Internet Architecture Board draft by Steve King and several colleagues (**`draft-iab-case-for-ipv6-05.txt`**). Other works in progress deal with the adjustments to Open Shortest Path First (OSPF), multicasting, mobility, and so on.

Transition

The period from 1981 through 1983—the time of conversion to DNS—was painful to all concerned. Over the past 15 years we have learned a lot, but the switch from IPv4 to IPv6 may be yet more painful. The drafts tell the tale of those who are striving to make things easier.

There has been much discussion about various kinds of transition mechanisms, and some of these may be less painful (more automated) than we might at first think. Remember, this pain is not because of the innate difficulty, but veering a ship that carries fewer than 250 passengers is far easier than veering a ship that carries 60 million. Some members of the community think that the pain may not justify the gain. The author is not one of them. It has been nearly 20 years since TCP/IP was made official, yet there are still UUCP networks.

In the author's opinion, IPv6 will be here in a few years, if not sooner.

Reference

[1] Fink, R., "IPv6—What and Where It Is," *The Internet Protocol Journal,* Volume 2, No. 1, March 1999.

PETER H. SALUS is the author of *A Quarter Century of UNIX* (1994) and *Casting the Net: From ARPANET to Internet and Beyond* (1995). He is the Editor in Chief of *The Handbook of Programming Languages* (1998). His e-mail address is:
`peter@pedant.com`

Book Review

An Engineering Approach to Computer Networking

An Engineering Approach to Computer Networking: ATM Networks, the Internet and the Telephone Network, Srinivasan Keshav, ISBN 0-201-63442-2, Addison-Wesley, 1997, **http://www.awl.com/cseng/titles/0-201-63442-2/**

The rapid convergence of telephone and data networks brings with it a collision of two diverse approaches to fundamental network design. This "New World," as it is often called, requires us to understand both the analog-to-digital evolution of the voice network, with its redundant search for faultless reliability, and the persistent tolerance of the data network. Mirroring the industry trend, this book explores the three major networking technologies: ATM, the Internet, and telephone networks, with the idea that the design of any modern network requires consideration of the influence of at least two of the three technologies.

This book is a textbook. Keshav himself declares in the preface that "textbooks, almost by definition, tend to be boring," and the reader will recall this subtle warning shortly into Chapter 2. This is definitely a book for those who have at least an intermediate knowledge of data networking and a need to understand the component parts of network implementations. Keshav takes a true engineering approach, in that he attempts to teach the building blocks of the major networking technologies—and this approach is what makes the book one of my all-time favorites. By examining the component parts and why they are required, Keshav leaves you prepared to engineer a network that meets any number of diverse criteria.

Organization

The book is organized into three sections. Section 1 gives an introduction to the future of data and voice networks and then introduces three of the major networking technologies. This section also gives an overview of the historic construction of networks, along with some fundamental definitions of some of the engineering principles by which networks function. As early as Chapter 1, Keshav explores the engineering philosophy behind common network technologies, illustrating the theories that underlie their design. My favorite example is his suggestion that the telephone network was engineered to be intelligent because its endpoints, the telephones, are simply dumb. While this sounds obvious, it provides a fundamental perspective on the design of the system that proves invaluable to understanding the origin of the various "components" of the network.

Section 2 begins with a short but requisite review of protocol layering and, after a brief discussion of common design constraints, begins to dissect the major components required of almost any network implementation. Chapter 8 is a fairly comprehensive review of switching and, as the book's title suggests, the chapter is full of comparative anatomy. Read this chapter for its valuable insight into why various switching mechanisms have emerged and for its comparison of how various switching functions are handled on three major networking technologies. Chapter 9 deals with scheduling network resources, with an excellent comparison of the variety of scheduling mechanisms and their effect on connections and packets. It covers policy considerations that are also required of scheduling disciplines, giving the reader a set of strategies for network design. Chapter 11 covers routing of packets as well as routing in the telephone network. In my opinion, this discussion alone makes this book a required part of any networking professional's library. Admittedly, there are books that better explain routing in both of these environments, but because of the proximity of the topics, this presentation helps the reader to understand the mechanics of both systems in a way that provides insight into the inherent issues posed by both technologies.

Section 3 pulls together the various component functions discussed in Section 2 and explains some of their implementation in the form of protocols. Section 3 is a short section, probably not intended as a thorough survey of networking protocols. Keshav documents an excellent set of references for Section 3, however, and leaves it up to the reader to pursue those that are relevant to his or her professional development.

Required Reading

An Engineering Approach to Computer Networking is definitely an A+ book, and should be required reading for anyone interested in the inner workings of data and voice networks. Although the author expects the reader to absorb quite a bit in every chapter, the time spent is well invested. The book is a refreshing alternative in that it provides an answer to the question of "why" the network works rather than being another treatise on "how" the network works.

—Jim LeValley, Cisco Press
levalley@cisco.com

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

Fragments

Internet Policy Institute Launched

On November 9th, 1999 a group of distinguished Internet visionaries and scholars announced the creation of the *Internet Policy Institute,* the nation's first independent, nonpartisan think tank devoted exclusively to providing research and hard data on the Internet and society. The group also announced its first research project and an initiative aimed at educating the presidential contenders.

The creation of the new think tank was announced by Jim Barksdale, former CEO of Netscape, Vint Cerf, Senior Vice President of Internet Architecture of MCI WorldCom, Esther Dyson, author and Chairman of EDventure Holdings, Inc., Mario Morino, Chairman of The Morino Institute, and Kimberly Jenkins, President of the Internet Policy Institute.

The new, nonprofit think tank will employ well-known experts and scholars to research subjects ranging from the role of the Internet in privacy to the Internet's impact on taxation and health care.

"The Internet is surrounded by noise, hype, rumors, marketing, IPOs and the hopes of starry-eyed start-ups, but there is very little hard data on which policymakers can base critical decisions that will determine the future of the new medium and how it affects society," said Barksdale, co-chairman of the Internet Policy Institute's Board of Directors. Wayne Clough, President of Georgia Tech, is his co-chairman.

"The speed at which society has adopted the Internet is unprecedented," said Cerf, who was Chairman and founding president of the Internet Society, as well as one of the designers of the TCP/IP protocol. "If, as we expect, half the world will be online within the next four years, we must make sure that the policy decisions we make now are based on solid, well-researched data."

The Institute announced its first research project, to be undertaken in collaboration with The Brookings Institution, on "The Economic Payoff from the Internet Revolution." The research will be led by Alice Rivlin, former vice chair of the Federal Reserve System's Board of Directors and former Office of Management and Budget director, now with the Brookings Institution, and Robert E. Litan, Vice President and Director of Economic Studies at The Brookings Institution and former associate director of the Office of Management and Budget. The research will produce the first comprehensive, systematic economic study by an independent research group of the subject.

The nature and extent of the impact is of special importance to macroeconomic policy—specifically monetary policy—to the extent that the Net is having or will have a material and sustained impact on the growth rate of productivity. The impact the Net has on specific industries, and the way it affects barriers to entry, has important implications for antitrust and regulatory policy.

Exactly one year before the next presidential election, the Internet Policy Institute also announced its first publications project, "Briefing the President: What the Next President of the United States Needs to Know About the Internet and Its Transformative Impact on Society." The Institute also released the introduction to the project by Barksdale, while Cerf outlined the contents of the next paper, "What is the Internet (and What Makes It Work)" that will be released December 1. Over the course of the coming months, the Institute will release 13 papers to be presented in briefings to all the leading presidential contenders and later compiled into a book.

"We didn't know five years ago the direction that the Internet would take," Barksdale said. "I'll bet that five years from now, we'll be surprised by its new directions. We need to assure that an honest, objective approach is taken on Internet issues, to prevent decision making that hinders the potential of this amazing medium," he said. For more information see: **http://www.internetpolicy.org**

APRICOT 2000

The *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will be held in at the Intercontinental Hotel in Seoul, Korea from February 28th to March 2nd, 2000. APRICOT provides a forum for key Internet builders in the region to learn from their peers and other leaders in the Internet community from around the world. The week-long summit consists of seminars, workshops, tutorials, conference sessions, and birds-of-a-feather sessions—all with the goal of spreading and sharing the knowledge required to operate the Internet within the Asia Pacific region. For more information see:
http://www.apricot.net

More on Web Caching

If you enjoyed the article on Web Caching in our September 1999 issue, you might find the following paper of interest: "A Survey of Web Caching Schemes for the Internet," by Jia Wang. You can find this article in the October 1999 issue of ACM SIGCOMM's *Computer Communications Review* (Volume 29, Number 5). The paper is also available on line in either PostScript or PDF format:
http://www.acm.org/sigcomm/ccr/archive/1999/oct99/ccr9910-jia-wang.html

ICANN Update

On September 28, 1999, the United States Department of Commerce, Network Solutions, Inc. (NSI), and The Internet Corporation for Assigned Names and Numbers (ICANN) announced a series of agreements they had tentatively reached to resolve outstanding differences among the three parties. On November 4, 1999, based on public comment in writing and at a public forum held at the 1999 ICANN annual meeting, the ICANN Board approved revised versions of these agreements. The agreements were signed by the three parties on November 10, 1999. The full text of the agreements can be found on the ICANN Web site at **`www.icann.org`**. Here we include some highlights:

- NSI will operate the registry for the **`.com`**, **`.net`**, and **`.org`** top-level domains according to requirements stated in the agreement and developed in the future through the ICANN consensus-based process. All accredited registrars will have equal access to this registry.
- A revised registrar accreditation agreement between ICANN and registrars was adopted. To continue to register names with the **`.com`**, **`.net`**, and **`.org`** registry operated by NSI after November 30, 1999, registrars must have entered a new Registrar License and Agreement with NSI and the revised ICANN accreditation agreement.
- A revised NSI-Registrar License and Agreement was created under which competitive ICANN-accredited registrars are permitted to place and renew registrations in the registry.
- An amendment was made to Cooperative Agreement #NCR 92-18742 originally entered between NSI and the National Science Foundation (NSF) in 1992. On October 7, 1998, NSI and the United States Department of Commerce (which by then had assumed the NSF's role as lead agency of the U.S. Government) entered an Amendment 11 to that Cooperative Agreement under which NSI agreed to implement a shared registration system in which competitive registrars would enter registrations into the **`.com`**, **`.net`**, and **`.org`** registry on an equitable basis. Amendment 19 solidifies those arrangements and provides that in operating the registry NSI will abide by consensus policies adopted in the ICANN process.

At the annual meeting in early November, nine new directors joined the ICANN Board of Directors. They are Robert Blokzij, Ken Fockler and Pindar Wong named by the The Address Supporting Organization (ASO); Amadeu Abril i Abril, Jonathan Cohen and Alejandro Pisanty named by the Domain Name Supporting Organization (DNSO); Jean-François Abramatic, Vinton G. Cerf and Philip Davidson named by the Protocol Supporting Organization (PSO).

The newly expanded ICANN Board will take on a major challenge in 2000 in its consideration of contending proposals for the future of Top Level Domains. After years of vociferous argument, the DNS community is no closer than it ever has been to a consensus on whether new name registries should be created, and if so, with what structure and registration rules.

Interplanetary Internet Special Interest Group Formed

The Internet Society (ISOC) recently announced the formation of the Interplanetary Internet Special Interest Group (IPNSIG). The IPNSIG exists to allow public participation in the evolution of the Interplanetary Internet. The technical research into how the Earth's Internet may be extended into interplanetary space has been underway for several years as part of an international communications standardization body known as the Consultative Committee on Space Data Systems (CCSDS). (See **http://www.ccsds.org/**)

The CCSDS organization is primarily concerned with communications standardization for scientific satellites, with a primary focus on the needs of near-term missions. In order to extend this horizon out several decades, and to begin to involve the terrestrial internet research and engineering communities, a special Interplanetary Internet Study was proposed and subsequently funded in the United States.

The Interplanetary Internet Study is funded by the Defense Advanced Research Projects Agency's Next Generation Internet Initiative, and presently consists of a core team of researchers from the NASA Jet Propulsion Laboratory, MITRE Corporation, SPARTA, Global Science & Technology and consulting researchers from The University of Southern California Information Sciences Institute, University of California Los Angeles and the California Institute of Technology. The primary goal of the study is to investigate how terrestrial internet protocols and techniques may be extended and/or used as-is in the exploration of deep space. The study team has also founded the IPNSIG and has formed the core of an Interplanetary Internet Research Group under the sponsorship of the Internet Research Task Force (IRTF).

The NASA IPN Study Team will act as liaison between the satellite and space communities and the ISOC/IRTF communities. The NASA IPN Study Team will assist with requirements and understanding of the deep space environment and missions, while the primary research on new or modified protocols will be conducted by the IRTF. In addition, the NASA Study Team will also act as liaison with the CCSDS.

The NASA Study Team will also enable simulated and actual opportunities to test protocols and the use of internet techniques in the space environment. For more information, visit: **ipn.jpl.nasa.gov/**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Cisco News Publications Group, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

March 2000 Volume 3, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Work on a new version of the Internet Protocol, known as IPv6, has been under way for several years in the IETF. There is still some debate about when and how IPv6 will be deployed. Proponents of IPv6 argue that the demand for new IP addresses will continue to rise to a point where we will simply run out of available IPv4 addresses and that we should, therefore, start deploying IPv6 *today*. Opponents argue that such a protocol transition will be too costly and painful for most organizations. They also argue that careful address management and the use of *Network Address Translation* (NAT) will allow continued use of the IPv4 address space for a very long time. Regardless of the timeframe, a major factor in the deployment of IPv6 is an appropriate transition strategy that allows existing IPv4 systems to communicate with new IPv6 systems. A transition mechanism, known as "6to4," is described in our first article by Brian Carpenter, Keith Moore, and Bob Fink.

In previous editions of this journal, we have looked at various security technologies for use in the Internet. Security mechanisms have been added at every layer of the protocol stack, and IP itself is no exception. IP Security, commonly known as "IPSec," is being deployed in many public and private networks. In our second article, William Stallings describes the main features of IPSec and looks at how IPSec can be used to build Virtual Private Networks.

Our final article is a critical look at *Quality of Service* (QoS) in the Internet. The need to provide different priorities to different kinds of traffic in a network is well understood and the technical community has been hard at work developing numerous systems to address this need. Geoff Huston looks at the prospects of deploying QoS solutions that will operate across the Internet as a whole.

The Y2K transition has been described as a "nonevent" by many. However, the lessons learned and the collaborative coordination efforts that were put in place for this transition can hopefully be used in the future. A colleague of mine had to call a plumber to his house on New Year's Eve. When he tried to pay for the repair with a credit card which had "00" as the expiration year, the plumber insisted that this meant the card was invalid. So while most systems were "Y2K compliant," this particular plumber was clearly not. Do you have a Y2K story to share? Drop us a line at **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Connecting IPv6 Routing Domains Over the IPv4 Internet

by Brian E. Carpenter, IBM & iCAIR
Keith Moore, University of Tennessee
Bob Fink, Energy Sciences Network

A next-generation Internet Protocol[1], known first as IPng and then as IPv6, has been under development by the *Internet Engineering Task Force* (IETF) for several years to replace the current Internet Protocol known as IPv4. The reasons behind the need for IPv6 are not covered here, but interested readers are encouraged to read "The Case for IPv6"[2] for this background.

Of major importance during the development of IPv6 has been how to do the transition away from IPv4, and towards IPv6. The work on transition strategies, tools, and mechanisms has been part of the basic IPv6 design effort from the beginning. The current transition efforts, taking place at the *IETF IPng Transition Working Group* (ngtrans)[3], will continue until it is clear that the transition will be successful.

These transition design efforts resulted in a basic Transition Mechanisms specification for IPv6 hosts and routers[4] that specifies the use of a Dual IP layer providing complete support for both IPv4 and IPv6 in hosts and routers, and IPv6-over-IPv4 *tunneling*, encapsulating IPv6 packets within IPv4 headers to carry them over IPv4 routing infrastructures.

These concepts are heavily relied on for transition from the traditional IPv4-based Internet as we know it today, to an IPv6-based Internet. It is expected that IPv4 and IPv6 will coexist for many years during this transition.

Of great concern to transition strategy planners is how to provide connectivity between IPv6-enabled end-user sites (also known as *routing domains*) when they do not yet have a reasonable (or any) choice of *Internet Service Provider* (ISP) that provides native IPv6 transport services. One way to provide IPv6 connectivity between end-user sites (when native IPv6 service does not exist) is to use IPv6-over-IPv4 encapsulation (tunneling) between them, similar to the technique currently used in the 6bone[5] IPv6 testbed network. This requires complexity for both end-user sites, and the networks providing the tunneling service (for instance, the 6bone backbone ISPs), in creating, managing, and operating manually configured tunnels.

The "6to4" transition mechanism, "Connection of IPv6 Domains via IPv4 Clouds without Explicit Tunnels"[6], provides a solution to the complexity problem of using manually configured tunnels by specifying a unique routing prefix for each end-user site that carries an IPv4 tunnel endpoint address.

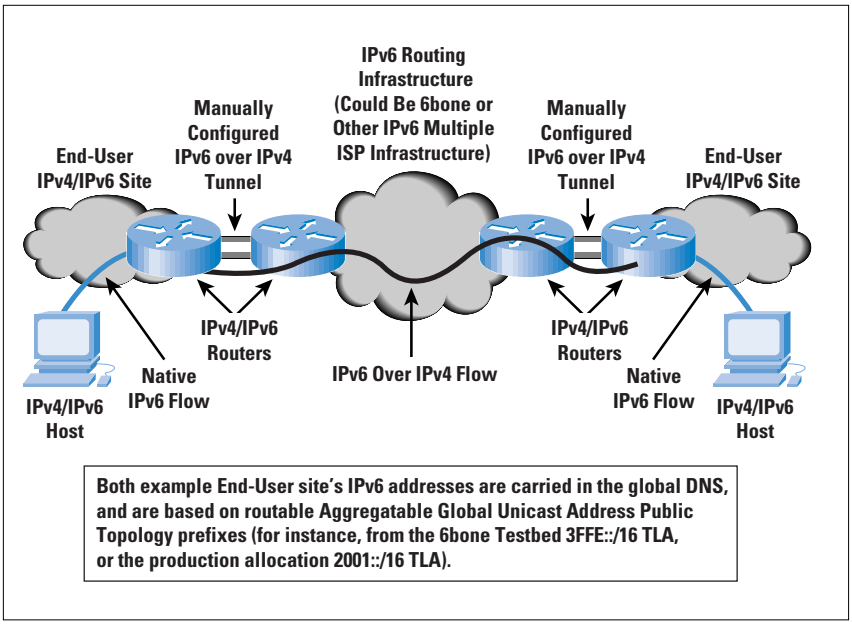
It should also be noted that each end-user site with as little as a single IPv4 address has a unique, routable, IPv6 site routing prefix thanks to the 6to4 transition mechanism.

Connecting IPv6 Routing Domains

When end-user site networks enable IPv6 in their local host and router systems, but have no native IPv6 Internet service, connectivity to other IPv6 routing domains across a worldwide Internet must be accomplished another way, or the value of a connected Internet is lost. Prior to the 6to4 transition mechanism, a site's network staff would have to rely on the manual configuration of IPv6-over-IPv4 tunnels to accomplish this connectivity.

This connectivity could be accomplished by arranging tunnels directly with each IPv6 site to which connectivity is needed, but more typically is done by arranging a tunnel into a larger IPv6 routing infrastructure that could guarantee connectivity to all IPv6 end-user site networks. (See Figure 1.) The 6bone IPv6 testbed was the first IPv6 routing infrastructure to provide worldwide IPv6 connectivity (starting in 1996), while more recently (late 1999) networks providing production IPv6 Internet service have also interconnected to provide this connectivity. In fact, the 6bone and production IPv6 routing infrastructures are well interconnected to guarantee worldwide IPv6 connectivity.

Figure 1: Configured Tunnel Overview



However, even given a solid, reliable, worldwide IPv6 routing infrastructure (similar to the IPv4-based Internet today), if an end-user site does not have a reasonable (or any) local choice for native IPv6 Internet service, a tunnel must be used.

The 6to4 mechanism addresses many of the practical difficulties with manually configured tunneling:

- The end-user site network staff must choose an IPv6 Internet service to tunnel to. This entails a process of at least three parts:
  - Finding candidate networks when the site's choice of IPv4 service does not provide IPv6 service (either tunneling or native),
  - Determining which ones are the best IPv4 path to use so that an IPv6-over-IPv4 tunnel doesn't inadvertently follow a very unreliable or low-performance path,
  - Making arrangements with the desired IPv6 service provider for tunneling service, a scenario that may at times be difficult if the selected provider is not willing to provide the service, or if for other administrative/cost reasons it is difficult to establish a business relationship.

Clearly it is easiest to use the site's own service provider, but in the early days of IPv6 transition this will often not be an option.

- An IPv6-over-IPv4 tunnel must be built to the selected provider, and a peering relationship must be established with the selected provider. This requires establishing a technical relationship with the provider and working through the various low-level details of how to configure tunnels between two routers, including answering the following questions:
  - Are the site and provider routers compatible early on in this process?
  - What peering protocol will be used (presumably an IPv6-capable version of the *Border Gateway Protocol Version 4* [BGP4]), and are the versions compatible and well debugged?
  - Have all the technical tunnel configuration issues between the site and provider been addressed?

Again, it is clearly easiest to perform all these steps if they are taken with the site's own IPv4 service provider.

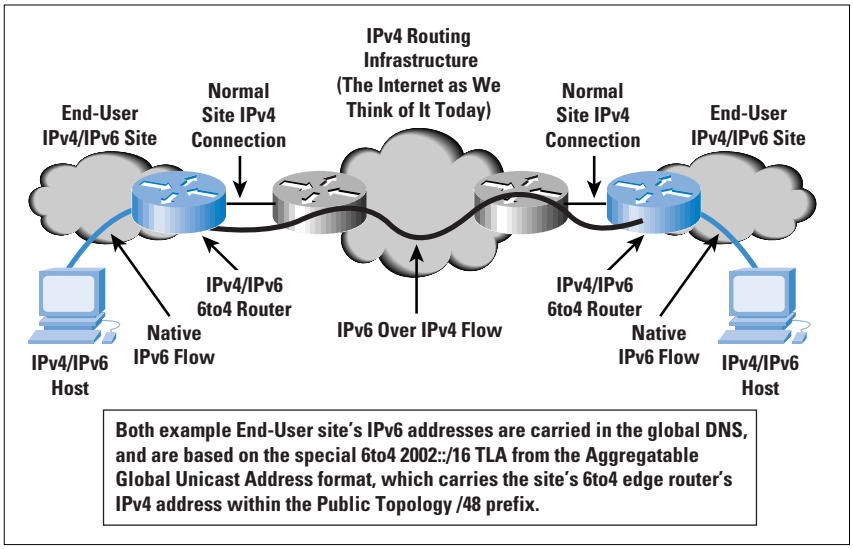


Figure 2: 6to4 Tunnel Overview

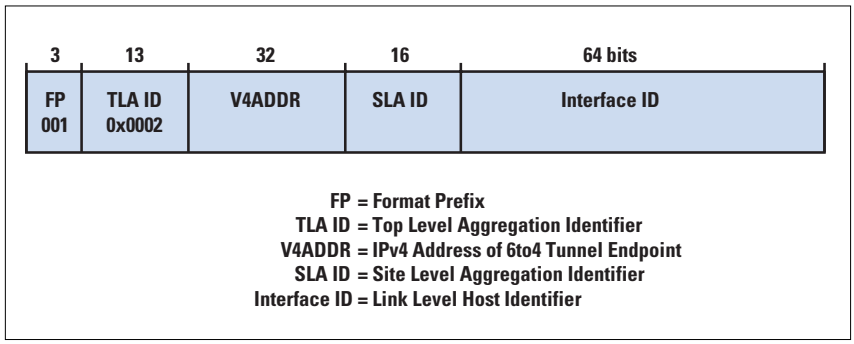
6to4 Eliminates Complex Tunnel Management

The 6to4 transition mechanism provides a solution to the complexity problem of building manually configured tunnels to an ISP by advertising a site's IPv4 tunnel endpoint (to be used for a dynamic tunnel) in a special external routing prefix for that site. Thus one site trying to reach another will discover the 6to4 tunnel endpoint from a *Domain Name System* (DNS) name to address lookup and use a dynamically built tunnel from site to site for the communication. (See Figure 2.) The tunnels are transient in that there is no state maintained for them, lasting only as long as a specific transaction uses the path. A 6to4 tunnel also bypasses the need to establish a tunnel to a wide-area IPv6 routing infrastructure, such as the 6bone.

The specification of a 48-bit external routing prefix in the IPv6 *Aggregatable Global Unicast Address Format* (AGGR)[7] (see Figure 3) that provides just enough space to hold the 32 bits required for the 32-bit IPv4 tunnel endpoint address (called V4ADDR in Figure 3) makes this setup possible.

Thus, this prefix has exactly the same format as normal prefixes assigned according to the AGGR. Within the subscriber site it can be used exactly like any other valid IPv6 prefix, for instance, for automated address assignment and discovery according to the normal IPv6 mechanisms for this.

Figure 3: 6to4 Prefix Format



The Simplest Use of 6to4

The simplest scenario for 6to4 is when several sites start to use IPv6 alongside IPv4, and have no native IPv6 ISP service available. Thus each site identifies a router to run dual stack (that is, IPv4 and IPv6 together) and 6to4 tunneling, ensuring that this router has a globally routable IPv4 address (that is, not in private IPv4 address space).

It is assumed that this new 6to4 router is reachable by IPv6-capable hosts within the site. Although the various ways in which these hosts may be reached are not discussed in detail here, they include using IPv6-enabled site IPv4 routers, operating special IPv6-only routers in parallel with site IPv4 routers, using the "6over4" mechanism[8], and employing other tunneling methods.

A new 6to4 site advertises the 6to4 prefix to its site via the *Neighbor Discovery* (ND) protocol[9], which will cause IPv6 hosts at this site to have their DNS name/address entries to include the 6to4 prefix for the site in them.

In operation, when one IPv6-enabled host at a 6to4 site tries to access an IPv6-enabled host by domain name at another 6to4 site, the DNS will return both an IPv4 and an IPv6 IP address for that host, indicating that it is reachable by both IPv4 and IPv6. The requesting host selects the IPv6 address, which will have a 6to4 prefix, and sends a packet off to its nearest router, eventually reaching its site boundary router, which we assume has 6to4 service as well.

Sending and Receiving Rules for 6to4 Routers

When the requesting site's 6to4 router sees that it must send a packet to another site (that is, there is a nonlocal destination), and that the next hop destination prefix contains the special 6to4 *Top Level Aggregation* (TLA) value of 2002::/16, the IPv6 packet is encapsulated in an IPv4 packet using an IPv4 protocol type of 41, as defined in the *Transition Mechanisms* RFC[4]. The source IPv4 address will be the one in the requesting site's 6to4 prefix (which is the IPv4 address of the outgoing interface to the Internet on the 6to4 router, and contained in the source 6to4 prefix of the IPv6 packet), and the destination IPv4 address will be the one in the next hop destination 6to4 prefix of the IPv6 packet.

When the destination site's 6to4 router receives the IPv4 packet, and recognizes that it has an IPv4 protocol type of 41, IPv4 security checks are made and the IPv4 header is removed, leaving the original IPv6 packet for local forwarding.

The sending rule above is the only modification to IPv6 forwarding, because the receiving rule was already specified for the basic IPv6 Transition Mechanism mentioned earlier[4]. Along with advertisement of the 6to4 prefix by appropriate entries in the DNS, any number of sites can interoperate without manual tunnel configuration.

It is not necessary to operate an exterior routing protocol (for instance, BGP4+) for 6to4 simple scenarios because the IPv4 exterior routing protocol is handling this function. Also, no new entries in IPv4 routing tables result from the use of 6to4.

The Return Path and Source Address Selection

Packets must flow in both directions to be useful; thus it is essential that IPv6 packets sent use a packet with a 6to4 prefix as a source address when talking to a site with a 6to4 prefix; in other words, the destination must have a 6to4 prefix. In the simple example given above, this is not an issue because both sites have only IPv4 connectivity, so they have 6to4 prefixes for their site to communicate with. DNS lookups for host systems at these sites will return only one IPv6 address, which will be the one with a 6to4 prefix. Source address selection is thus not an issue.

As we will soon see, source address selection is an issue for more complex 6to4 usage scenarios; therefore, some source address selection algorithm is necessary in IPv6 hosts. The exact form and method of the algorithm to use is under active study at the IETF IPv6 (ipng) working group[10], and an algorithm is likely to be chosen in early 2000. Meanwhile, for the purposes of understanding 6to4, it is sufficient to realize that when a 6to4 connected sending site is sending to a destination site using that site's 6to4 prefix, the sending host must guarantee that the source IPv6 address uses the sending site's 6to4 prefix.

More Complex 6to4 Usage Scenarios

Several more interesting 6to4 usage scenarios exist when a site has both 6to4 connectivity and native IPv6 connectivity. The simplest of these is when such a site is trying to reach another site that has only 6to4 connectivity, in which case the source address selection algorithm mentioned above is essential to ensure that the site's 6to4 IPv6 address is chosen. No destination selection is required because there is only one choice, that is, 6to4.

Similarly, when a site that has only 6to4 connectivity tries to reach a site with both 6to4 and native IPv6 connectivity, some host rule for choosing among multiple destination addresses must result in the 6to4 address being chosen, because only a local 6to4 IPv6 source address is available. Of course source selection is not an issue in this case because there is only the 6to4 IPv6 address to use.

Another variation of these scenarios is when a site with 6to4 and native IPv6 connectivity is trying to reach another site that has only native IPv6 connectivity, making a source address selection algorithm essential to make sure the site's native IPv6 address is chosen. No destination selection is required, because there is only one choice, that is, the native IPv6 address.

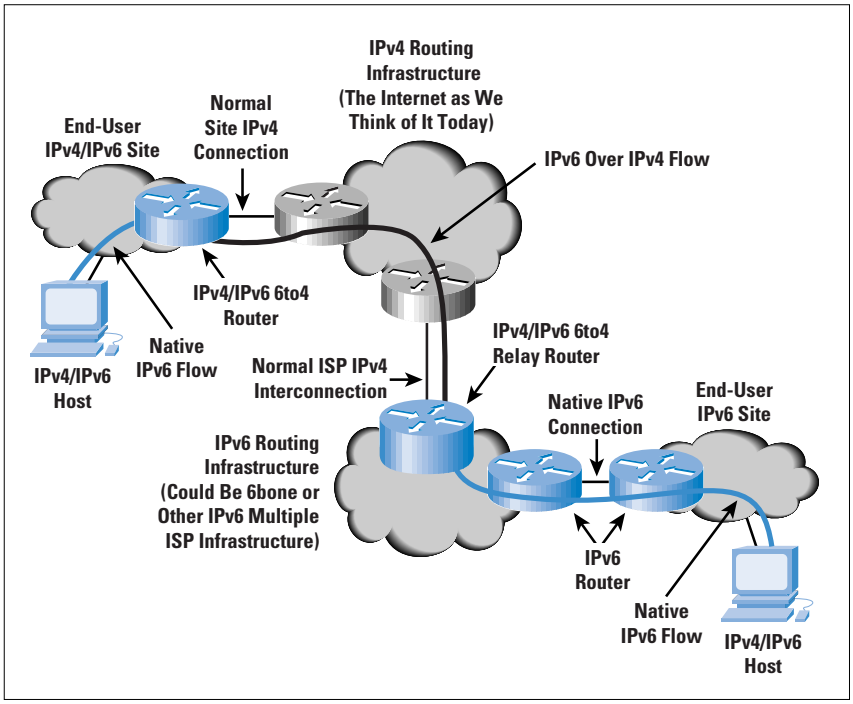
Similarly, when a site that has only native IPv6 connectivity tries to reach a site with 6to4 and native IPv6 connectivity, a host rule is essential for choosing among multiple addresses to ensure that a native IPv6 address is chosen, because only a local native IPv6 source address is available. Again, source selection is not an issue in this case because only the native IPv6 address can be used.

An interesting choice develops in the situation when both sites have 6to4 and native IPv6 connectivity as both 6to4-to-6to4 and native IPv6-to-native-IPv6 connections are a possibility. Current thinking as of the writing of this article is to prefer the native IPv6 connection.

The 6to4 Relay

The most interesting, and most complex, 6to4 scenario is that of sites with only 6to4 connectivity communicating with sites with only native IPv6 connectivity. This is accomplished by the use of a 6to4 relay that supports both 6to4 and native IPv6 connectivity (Figure 4). The 6to4 relay is nothing more than an IPv4/IPv6 dual-stack router.

Figure 4: The 6to4 Relay



The 6to4 relay advertises a route to 2002::/16 for itself into the native IPv6 infrastructure it is attached to. The native IPv6 network operators must filter out and discard any 6to4 (2002:...) prefix advertisements longer than /16. In addition, the 6to4 relay may advertise into its 6to4 connection whatever native IPv6 routes its policies allow, which the 6to4 router at the 6to4-only site picks up with either a BGP4+ peering session, or with a default route, to the 6to4 relay.

Thus the 6to4-only site will try to send a packet to the native IPv6-only site by forwarding an encapsulated (tunneled) IPv6 packet to the 6to4 relay, which removes the IPv4 header (decapsulates) and forwards the packet on to the IPv6-only site.

Potentially, multiple 6to4 relays are needed, one for each separate IPv6 routing realm (collection of IPv6 routing ISPs). In practice, it is expected that all native IPv6 ISP services will be interconnected even if the use of inter-IPv6-ISP manually configured tunnels are required to do so. This is currently the case as of early 2000, because all 6bone 3FFE::/16 TLA networks and all production 2001::/16 subTLA networks are interconnected with each other.

It is expected that native IPv6 service providers will choose to operate 6to4 relays as a simple extension of their service. There are no special rules or exceptions to 6to4 as described here for this to happen because the 6to4 relay is simply operated as part of an end-user site that belongs to the IPv6 ISP.

Other Issues

Several other 6to4 issues are presented below for completeness.

- The IPv6 *Maximum Transmission Unit* (MTU) size could prove too large for some intermediate IPv4 link when a 6to4 tunnel is in use, thus IPv4 fragmentation will occur. Though undesirable, fragmentation is not disastrous, so the IPv4 "Do Not Fragment" bit should not be set in the IPv4 packet carrying the 6to4 tunnel.
- How sites move IPv6 packets internal to a site is not important to the 6to4 process. For illustrative purposes in this article, it is generally assumed that native IPv6 transmission exists within a site. This may not be strictly true because "6over4," manual tunnels, and other methods of moving IPv6 packets could be in use. Nonetheless, it is not important to the 6to4 processes described here.
- Security issues with the 6to4 mechanism are not discussed here. The reader is referred to the current 6to4 draft for an explanation of these issues[6].
- 6to4 sites with IPv6 connectivity must not inject their 6to4 prefix into the IPv6 routing infrastructure via the native IPv6 connection.
- It is not possible to assume the general availability of wide-area IPv4 multicast, so the 6to4 mechanism must assume only unicast capability in its underlying IPv4 carrier network. However, it is expected that IPv6 multicast packets may be sent to, or sourced from, a 6to4 router in the IPv4 encapsulated form, as described above. When IPv6 multicast is supported, an IPv6 multicast routing protocol must be used.
- The use of IPv6 Anycast is compatible with 6to4 prefixes.
- 6to4 for hosts only, as opposed to sites, is possible and will likely be developed in the future. However, details of this feature are not discussed in this article.
- The 6to4 mechanism is unaffected by the presence of a firewall at the border router.
- When using IPv4 *Network Address Translation* (NAT), 6to4 mechanisms remain valid, and the NAT device includes a fully functional IPv6 router with the 6to4 mechanism included. Combining 6to4 and NAT in this way offers the advantages of NAT for IPv4 use, and the additional address space of IPv6.
- There is no significant impact to either IPv4 or IPv6 routing table size caused by the proper implementation of 6to4.

Summarizing 6to4

The 6to4 mechanism allows isolated IPv6 routing domains to communicate with other IPv6 routing domains, even in the total absence of native IPv6 service providers. It is a powerful IPv6 transition tool that will allow both traditional IPv4-based Internet end-user sites and new IPv6-only Internet sites to utilize IPv6 and operate successfully over the existing IPv4-based Internet routing infrastructure.

For Further Reading

[0] Fink, R., "IPv6—What and Where It Is," *The Internet Protocol Journal,* Volume 2, No. 1, March 1999.

[1] IPng and IPv6 information, including formal specifications, can be found at: **http://playground.sun.com/pub/ipng/html**

[2] "The Case for IPv6," an Internet Draft of the IAB, can be found at: **http://www.6bone.net/misc/case-for-ipv6.html**

[3] IETF IPv6 Transition Working Group (ngtrans) information, including status of all its current projects, can be found at: **http://www.6bone.net/ngtrans/**

[4] "Transition Mechanisms for IPv6 Hosts and Routers," RFC 1933, can be found at: **http://www.ietf.org/rfc/rfc1933.txt**

[5] The 6bone IPv6 Testbed Network is explained at: **http://www.6bone.net**

[6] "Connection of IPv6 Domains via IPv4 Clouds without Explicit Tunnels" ("6to4"), an Internet Draft of the IETF ngtrans WG, can be found at: **http://www.6bone.net/misc/6to4.txt**

[7] "IPv6 Aggregatable Global Unicast Address Format," RFC 2374, can be found at: **http://www.ietf.org/rfc/rfc2374.txt**

[8] "Transmission of IPv6 Packets over IPv4 Domains without Explicit Tunnels" ("6over4"), RFC 2529, can be found at: **http://www.ietf.org/rfc/rfc2529.txt**

[9] "Neighbor Discovery for IP Version 6 (IPv6)," RFC 2461, can be found at: **http://www.ietf.org/rfc/rfc2461.txt**

[10] IETF IPv6 Working Group (ipngwg) information, can be found at: **http://www.ietf.org/html.charters/ipngwg-charter.html**

BRIAN E. CARPENTER is a network researcher with the IBM Internet Division at iCAIR in Evanston Illinois. He is currently the Chair of the Internet Architecture Board (IAB) of the IETF. You can reach him at: **brian@icair.org**

KEITH MOORE is a network researcher at the Innovative Computing Laboratory of the Computer Science Department at the University of Tennessee. He is currently a Co-Director of the IETF Applications Area in the Internet Engineering Steering Group. You can reach him at: **moore@cs.utk.edu**

ROBERT FINK is a network researcher with the U.S. Dept. of Energy's Energy Sciences Network (ESnet) at the Lawrence Berkeley National Laboratory. He is currently a co-chair of the IETF ngtrans (IPng Transition) Working Group, and leads the 6bone project. You can reach him at: **fink@es.net**

IP Security

by William Stallings

In 1994, the *Internet Architecture Board* (IAB) issued a report entitled "Security in the Internet Architecture" (RFC 1636). The report stated the general consensus that the Internet needs more and better security, and it identified key areas for security mechanisms. Among these were the need to secure the network infrastructure from unauthorized monitoring and control of network traffic and the need to secure end-user-to-end-user traffic using authentication and encryption mechanisms.

These concerns are fully justified. As confirmation, the 1998 annual report from the *Computer Emergency Response Team* (CERT) lists over 1,300 reported security incidents affecting nearly 20,000 sites. The most serious types of attacks included IP spoofing, in which intruders create packets with false IP addresses and exploit applications that use authentication based on IP address; and various forms of eavesdropping and packet sniffing, in which attackers read transmitted information, including logon information and database contents.

In response to these issues, the IAB included authentication and encryption as necessary security features in the next-generation IP, which has been issued as IPv6. Fortunately, these security capabilities were designed to be usable both with the current IP (IPv4) and IPv6, meaning that vendors can begin offering these features now, and many vendors do now have some *IP Security Protocol* (IPSec) capability in their products.

Applications of IPSec

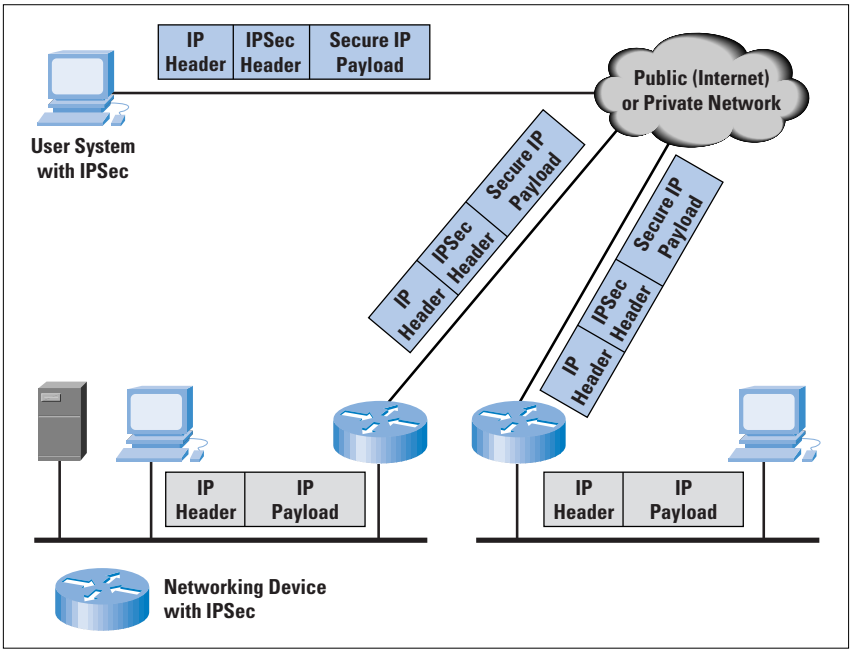
The Internet community has developed application-specific security mechanisms in numerous application areas, including electronic mail (*Privacy Enhanced Mail*, *Pretty Good Privacy* [PGP]), network management (*Simple Network Management Protocol Version 3* [SNMPv3]), Web access (Secure HTTP, *Secure Sockets Layer* [SSL]), and others. However, users have some security concerns that cut across protocol layers. For example, an enterprise can run a secure, private TCP/IP network by disallowing links to untrusted sites, encrypting packets that leave the premises, and authenticating packets that enter the premises. By implementing security at the IP level, an organization can ensure secure networking not only for applications that have security mechanisms but also for the many security-ignorant applications.

IPSec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include:

- Secure branch office connectivity over the Internet: A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.
- Secure remote access over the Internet: An end user whose system is equipped with IP security protocols can make a local call to an *Internet Service Provider* (ISP) and gain secure access to a company network. This reduces the cost of toll charges for traveling employees and telecommuters.
- Establishment of extranet and intranet connectivity with partners: IPSec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.
- Enhancement of electronic commerce security: Most efforts to date to secure electronic commerce on the Internet have relied upon securing Web traffic with SSL since that is commonly found in Web browsers and is easy to set up and run. There are new proposals that may utilize IPSec for electronic commerce.

The principal feature of IPSec that enables it to support these varied applications is that it can encrypt or authenticate *all* traffic at the IP level. Thus, all distributed applications, including remote logon, client/server, e-mail, file transfer, Web access, and so on, can be secured. Figure 1 shows a typical scenario of IPSec usage. An organization maintains LANs at dispersed locations. Traffic on each LAN does not need any special protection, but the devices on the LAN can be protected from the untrusted network with firewalls. Since we live in a distributed and mobile world, the people who need to access the services on each of the LANs may be at sites across the Internet. These people can use IPSec protocols to protect their access. These protocols can operate in networking devices, such as a router or firewall that connects each LAN to the outside world, or they may operate directly on the workstation or server. In the diagram, the user workstation can establish an IPSec tunnel with the network devices to protect all the subsequent sessions. After this tunnel is established, the workstation can have many different sessions with the devices behind these IPSec gateways. The packets going across the Internet will be protected by IPSec but will be delivered onto each LAN as a normal IP packet.

Figure 1: An IP Security Scenario



Benefits of IPSec

The benefits of IPSec include:

- When IPSec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. Traffic within a company or workgroup does not incur the overhead of security-related processing.
- IPSec is below the transport layer (TCP, UDP), so is transparent to applications. There is no need to change software on a user or server system when IPSec is implemented in the firewall or router. Even if IPSec is implemented in end systems, upper layer software, including applications, is not affected.
- IPSec can be transparent to end users. There is no need to train users on security mechanisms, issue keying material on a per-user basis, or revoke keying material when users leave the organization.
- IPSec can provide security for individual users if needed. This feature is useful for offsite workers and also for setting up a secure virtual subnetwork within an organization for sensitive applications.

Is IPSec the Right Choice?

There are already numerous products that implement IPSec, but it is not necessarily the security solution of choice for a network administrator. Christian Huitema, who at the time of the development of the initial IPSec documents was the head of the IAB, reports that the debates over how to provide Internet-based security were among the most heated that he ever observed. One issue concerns whether security is being provided at the right protocol layer. To provide security at the IP level, it is necessary for IPSec to be a part of the network code deployed on all participating platforms, including Windows NT, UNIX, and Macintosh systems. Unless a desired feature is available on all the deployed platforms, a given application may not be able to use that feature.

On the other hand, if the application, such as a Web browser/server combination, incorporates the function, the developer can guarantee that the features are available on all platforms for which the application is available. A related point is that many Internet applications are now being released with embedded security features. For example, Netscape and Internet Explorer support SSL, which protects Web traffic. Also, many vendors are planning to support *Secure Electronic Transaction* (SET), which protects credit-card transactions over the Internet. However, for a virtual private network, a network-level facility is needed, and this is what IPSec provides.

The Scope of IPSec

IPSec provides three main facilities: an authentication-only function, referred to as *Authentication Header* (AH), a combined authentication/encryption function called *Encapsulating Security Payload* (ESP), and a key exchange function. For virtual private networks, both authentication and encryption are generally desired, because it is important both to (1) assure that unauthorized users do not penetrate the virtual private network and (2) assure that eavesdroppers on the Internet cannot read messages sent over the virtual private network. Because both features are generally desirable, most implementations are likely to use ESP rather than AH. The key exchange function allows for manual exchange of keys as well as an automated scheme.

The IPSec specification is quite complex and covers numerous documents. The most important of these, issued in November 1998, are RFCs 2401, 2402, 2406, and 2408.

Security Associations

A key concept that appears in both the authentication and confidentiality mechanisms for IP is the *Security Association* (SA). An association is a one-way relationship between a sender and a receiver that affords security services to the traffic carried on it. If a peer relationship is needed, for two-way secure exchange, then two security associations are required. Security services are afforded to an SA for the use of AH or ESP, but not both. A security association is uniquely identified by three parameters:

- *Security Parameters Index* (SPI): The SPI assigns a bit string to this SA that has local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.
- *IP destination address:* Currently, only unicast addresses are allowed; this is the address of the destination endpoint of the SA, which may be an end-user system or a network system such as a firewall or router.
- *Security protocol identifier:* This indicates whether the association is an AH or ESP security association.

Hence, in any IP packet, the security association is uniquely identified by the destination address in the IPv4 or IPv6 header and the SPI in the enclosed extension header (AH or ESP).

An IPSec implementation includes a security association database that defines the parameters associated with each SA. A security association is defined by the following parameters:

- *Sequence number counter:* A 32-bit value used to generate the sequence number field in AH or ESP headers
- *Sequence counter overflow:* A flag indicating whether overflow of the sequence number counter should generate an auditable event and prevent further transmission of packets on this SA
- *Anti-replay window:* Used to determine whether an inbound AH or ESP packet is a replay, by defining a sliding window within which the sequence number must fall
- *AH information:* Authentication algorithm, keys, key lifetimes, and related parameters being used with AH
- *ESP information:* Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP
- *Lifetime of this security association:* A time interval or byte count after which an SA must be replaced with a new SA (and new SPI) or terminated, plus an indication of which of these actions should occur
- *IPSec protocol mode:* Tunnel, transport, or wildcard (required for all implementations); these modes are discussed later
- *Path MTU:* Any observed path maximum transmission unit (maximum size of a packet that can be transmitted without fragmentation) and aging variables (required for all implementations)

The key management mechanism that is used to distribute keys is coupled to the authentication and privacy mechanisms only by way of the security parameters index. Hence, authentication and privacy have been specified independent of any specific key management mechanism.

SA Selectors

IPSec provides the user with considerable flexibility in the way in which IPSec services are applied to IP traffic. IPSec provides a high degree of granularity in discriminating between traffic that is afforded IPSec protection and traffic that is allowed to bypass IPSec, in the former case relating IP traffic to specific SAs.

The means by which IP traffic is related to specific SAs (or no SA in the case of traffic allowed to bypass IPSec) is the nominal *Security Policy Database* (SPD). In its simplest form, an SPD contains entries, each of which defines a subset of IP traffic and points to an SA for that traffic. In more complex environments, there may be multiple entries that potentially relate to a single SA or multiple SAs associated with a single SPD entry.

Each SPD entry is defined by a set of IP and upper-layer protocol field values, called *selectors*. In effect, these selectors are used to filter outgoing traffic in order to map it into a particular SA. Outbound processing obeys the following general sequence for each IP packet:

- Compare the values of the appropriate fields in the packet (the selector fields) against the SPD to find a matching SPD entry, which will point to zero or more SAs.
- Determine the SA (if any) for this packet and its associated SPI.
- Do the required IPSec processing (that is, AH or ESP processing).

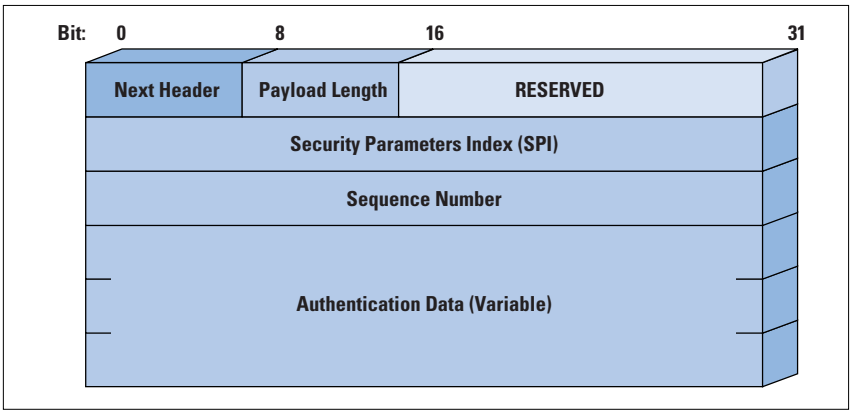
The following selectors determine an SPD entry:

- *Destination IP address:* This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one destination system sharing the same SA (for instance, behind a firewall).
- *Source IP address:* This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one source system sharing the same SA (for instance, behind a firewall).
- *UserID:* UserID is used to identify a policy tied to a valid user or system name.
- *Data sensitivity level:* The data sensitivity level is used for systems providing information flow security (for instance, "Secret" or "Unclassified").
- *Transport Layer protocol:* This value is obtained from the IPv4 protocol or IPv6 *Next Header* field. This may be an individual protocol number, a list of protocol numbers, or a range of protocol numbers.
- *IPSec protoco*l (AH or ESP or AH/ESP): If present, this is obtained from the IPv4 Protocol or IPv6 Next Header field.
- *Source and destination ports:* These may be individual TCP or *User Datagram Protocol* (UDP) port values, an enumerated list of ports, or a wildcard port.
- *IPv6 class:* This class is obtained from the IPv6 header. It may be a specific IPv6 Class value or a wildcard value.
- *IPv6 flow label:* This label is obtained from the IPv6 header. It may be a specific IPv6 flow label value or a wildcard value.
- *IPv4 Type of Service* (TOS): The TOS is obtained from the IPv4 header. It may be a specific IPv4 TOS value or a wildcard value.

Authentication Header

The authentication header provides support for data integrity and authentication of IP packets. The data integrity feature ensures that undetected modification to the content of a packet in transit is not possible. The authentication feature enables an end system or network device to authenticate the user or application and filter traffic accordingly; it also prevents the address spoofing attacks observed in today's Internet. The AH also guards against the replay attack described later.

Figure 2: IPSec Authentication Header



Authentication is based on the use of a *Message Authentication Code* (MAC); hence the two parties must share a secret key. The authentication header consists of the following fields (Figure 2):

- *Next Header* (8 bits): This field identifies the type of header immediately following this header.
- *Payload Length* (8 bits): This field gives the length of the authentication header in 32-bit words, minus 2. For example, the default length of the authentication data field is 96 bits, or three 32-bit words. With a three-word fixed header, there are a total of six words in the header, and the Payload Length field has a value of 4.
- *Reserved* (16 bits): This field is reserved for future use.
- *Security Parameters Index* (32 bits): This field identifies a security association.
- *Sequence Number* (32 bits): This field contains a monotonically increasing counter value.
- *Authentication Data* (variable): This variable-length field (must be an integral number of 32-bit words) contains the *Integrity Check Value* (ICV), or MAC, for this packet.

Anti-Replay Service

A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination. The receipt of duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence. The *Sequence Number* field is designed to thwart such attacks.

When a new SA is established, the *sender* initializes a sequence number counter to 0. Each time that a packet is sent on this SA, the sender increments the counter and places the value in the Sequence Number field. Thus, the first value to be used is 1. If anti-replay is enabled (the default), the sender must not allow the sequence number to cycle past $2^{32} - 1$ back to zero. Otherwise, there would be multiple valid packets with the same sequence number. If the limit of $2^{32} - 1$ is reached, the sender should terminate this SA, and negotiate a new SA with a new key.

Because IP is a connectionless, unreliable service, the protocol does not guarantee that packets will be delivered in order and does not guarantee that all packets will be delivered. Therefore, the IPSec authentication document dictates that the *receiver* should implement a window of size *W*, with a default of $W = 64$. The right edge of the window represents the highest sequence number, *N*, so far received for a valid packet. For any packet with a sequence number in the range from $N - W + 1$ to N that has been correctly received (that is, properly authenticated), the corresponding slot in the window is marked. Inbound processing proceeds as follows when a packet is received:

- If the received packet falls within the window and is new, the MAC is checked. If the packet is authenticated, the corresponding slot in the window is marked.
- If the received packet is to the right of the window and is new, the MAC is checked. If the packet is authenticated, the window is advanced so that this sequence number is the right edge of the window, and the corresponding slot in the window is marked.
- If the received packet is to the left of the window, or if authentication fails, the packet is discarded; this is an auditable event.

Message Authentication Code

The message authentication algorithm is used to calculate a message authentication code, using an algorithm known as *HMAC*. HMAC takes as input a portion of the message and a secret key and produces a MAC as output. This MAC value is stored in the Authentication Data field of the AH header. The calculation takes place over the entire enclosed TCP segment plus the authentication header. When this IP packet is received at the destination, the same calculation is performed using the same key. If the calculated MAC equals the value of the received MAC, then the packet is assumed to be authentic. The authentication data field is calculated over:

- IP header fields that either do not change in transit (immutable) or that are predictable in value upon arrival at the endpoint for the AH SA. Fields that may change in transit and whose value on arrival are unpredictable are set to zero for purposes of calculation at both source and destination.
- The AH header other than the Authentication Data field. The Authentication Data field is set to zero for purposes of calculation at both source and destination.
- The entire upper-level protocol data, which is assumed to be immutable in transit (for instance, a TCP segment or an inner IP packet in tunnel mode).

For IPv4, examples of immutable fields are *Internet Header Length* and *Source Address*. An example of a mutable but predictable field is the Destination Address (with loose or strict source routing). Examples of mutable fields that are zeroed prior to ICV calculation are the *Time to Live* (TTL) and *Header Checksum* fields.

Note that both source and destination address fields are protected, so that address spoofing is prevented. For IPv6, examples in the base header are *Version* (immutable), *Destination Address* (mutable but predictable), and *Flow Label* (mutable and zeroed for calculation).

Encapsulating Security Payload

The encapsulating security payload provides confidentiality services, including confidentiality of message contents and limited traffic flow confidentiality. As an optional feature, ESP can also provide the same authentication services as AH.

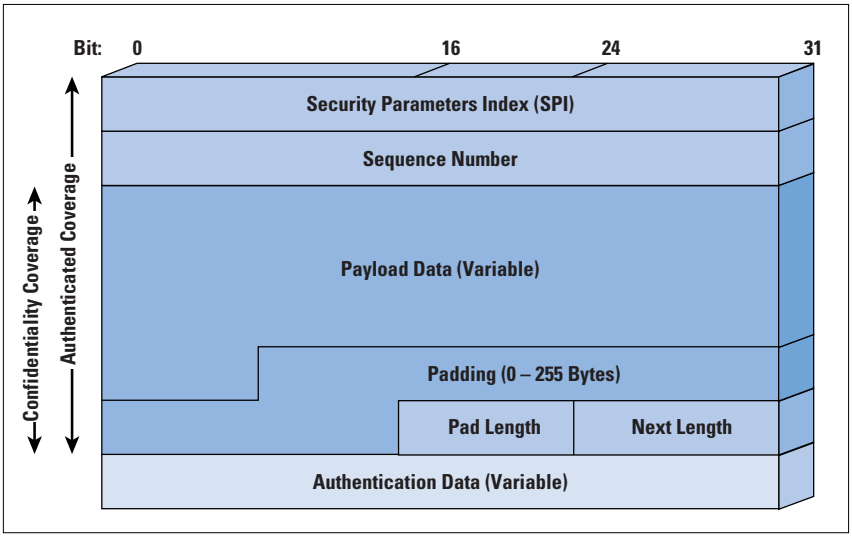


Figure 3: IPSec ESP Format

Figure 3 shows the format of an ESP packet. It contains the following fields:

- *Security Parameters Index* (32 bits): Identifies a security association
- *Sequence Number* (32 bits): A monotonically increasing counter value
- *Payload Data* (variable): A transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption
- *Padding* (0–255 bytes): Extra bytes that may be required if the encryption algorithm requires the plaintext to be a multiple of some number of octets
- *Pad Length* (8 bits): Indicates the number of pad bytes immediately preceding this field
- *Next Header* (8 bits): Identifies the type of data contained in the payload data field by identifying the first header in that payload (for example, an extension header in IPv6, or an upper-layer protocol such as TCP)
- *Authentication Data* (variable): A variable-length field (must be an integral number of 32-bit words) that contains the integrity check value computed over the ESP packet minus the Authentication Data field

Encryption and Authentication Algorithms

The Payload Data, Padding, Pad Length, and Next Header fields are encrypted by the ESP service. If the algorithm used to encrypt the payload requires cryptographic synchronization data, such as an *Initialization Vector* (IV), then this data may be carried explicitly at the beginning of the Payload Data field. If included, an IV is usually not encrypted, although it is often referred to as being part of the ciphertext. The current specification dictates that a compliant implementation must support the *Data Encryption Standard* (DES). A number of other algorithms have been assigned identifiers and could, therefore, be used for encryption; these include:

- Three-key triple DES
- RC5
- International Data Encryption Algorithm (IDEA)
- Three-key triple IDEA
- CAST
- Blowfish

It is now well known that DES is inadequate for secure encryption, so it is likely that many future implementations will use triple DES and eventually the *Advanced Encryption Standard* (AES). As with AH, ESP supports the use of a MAC, using HMAC.

Padding

- The Padding field serves several purposes: If an encryption algorithm requires the plaintext to be a multiple of some number of bytes (for instance, the multiple of a single block for a block cipher), the Padding field is used to expand the plaintext (consisting of the Payload Data, Padding, Pad Length, and Next Header fields) to the required length.
- The ESP format requires that the Pad Length and Next Header fields be right aligned within a 32-bit word. Equivalently, the ciphertext must be an integer multiple of 32 bits. The Padding field is used to assure this alignment.
- Additional padding may be added to provide partial traffic flow confidentiality by concealing the actual length of the payload.

Figure 4 indicates the scope of ESP encryption and authentication in both transport and tunnel modes.

Transport and Tunnel Modes

Both AH and ESP support two modes of use: *transport* and *tunnel* mode.

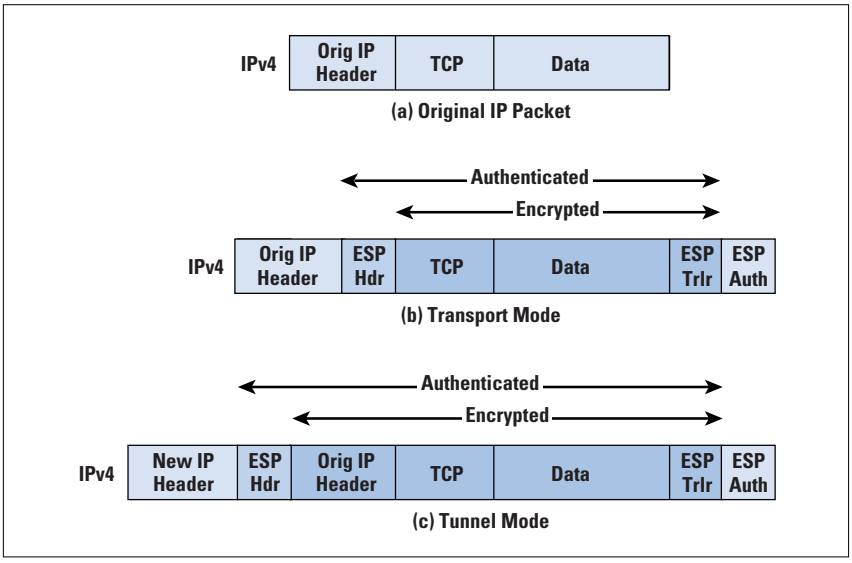


Figure 4: Scope of ESP Encryption and Authentication

Transport Mode

Transport mode provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet. Examples include a TCP or UDP segment, or an *Internet Control Message Protocol* (ICMP) packet, all of which operate directly above IP in a host protocol stack. For this mode using IPv4, the ESP header is inserted into the IP packet immediately prior to the transport-layer header (for instance, TCP, UDP, ICMP) and an ESP trailer (Padding, Pad Length, and Next Header fields) is placed after the IP packet. This setup is shown in Figure 4b. If authentication is selected, the ESP Authentication Data field is added after the ESP trailer. The entire transport-level segment plus the ESP trailer are encrypted. Authentication covers all of the ciphertext plus the ESP header.

Typically, transport mode is used for end-to-end communication between two hosts (for instance, communications between a workstation and a server, or two servers). When a host runs AH or ESP over IPv4, the payload is the data that normally follows the IP header. For IPv6, the payload is the data that normally follows both the IP header and any IPv6 extensions headers that are present, with the possible exception of the destination options header, which may be included in the protection.

ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header. AH in transport mode authenticates the IP payload and selected portions of the IP header. All IPv4 packets have a *Next Header* field. This field contains a number for the payload protocol, such as 6 for TCP and 17 for UDP. For transport mode, the IP Next Header field is decimal 51 for AH, or 50 for ESP. This tells the receiving machine to interpret the remainder of the packet after the IP header as either AH or ESP. Both the AH and ESP headers also have a Next Header field.

As an example, let's examine a Telnet session within an ESP packet in transport mode. The IP header would contain 51 in the Next Header field. In the ESP header, the Next Header field would be 6 for TCP. Within the TCP header, Telnet would be identified as port 23.

Transport mode operation may be summarized for ESP as follows:

- At the source, the block of data consisting of the ESP trailer plus the entire transport-layer segment is encrypted and the plaintext of this block is replaced with its ciphertext to form the IP packet for transmission. Authentication is added if this option is selected.
- The packet is then routed to the destination. Each intermediate router needs to examine and process the IP header plus any plaintext IP extension headers but will not need to examine the ciphertext.
- The destination node examines and processes the IP header plus any plaintext IP extension headers. Then, on the basis of the SPI in the ESP header, the destination node decrypts the remainder of the packet to recover the plaintext transport-layer segment. This process is similar for AH, however the payload (upper layer protocol) is not encrypted.

Transport mode operation provides confidentiality for any application that uses it, thus avoiding the need to implement confidentiality in every individual application. This mode of operation is also reasonably efficient, adding little to the total length of the IP packet. One drawback to this mode is that it is possible to do traffic analysis on the transmitted packets.

Tunnel Mode

Tunnel mode encapsulates an entire IP packet within an IP packet to ensure that no part of the original packet is changed as it is moved through a network. The entire original, or inner, packet travels through a "tunnel" from one point of an IP network to another; no routers along the way need to examine the inner IP header. For ESP, this is shown in Figure 4c. Because the IP header contains the destination address and possibly source routing directives and hop-by-hop option information, it is not possible simply to transmit the encrypted IP packet prefixed by the ESP header. Intermediate routers would be unable to process such a packet. Therefore, it is necessary to encapsulate the entire block (ESP header plus ciphertext plus Authentication Data, if present) with a new IP header that will contain sufficient information for routing but not for traffic analysis. Tunnel mode is used when one or both ends of an SA is a security gateway, such as a firewall or router that implements IPSec. With tunnel mode, a number of hosts on networks behind firewalls may engage in secure communications without implementing IPSec. The unprotected packets generated by such hosts are tunneled through external networks by tunnel mode SAs set up by the IPSec process in the firewall or secure router at the boundary of the local network.

Whereas the transport mode is suitable for protecting connections between hosts that support the ESP feature, the tunnel mode is useful in a configuration that includes a firewall or other sort of security gateway that protects a trusted network from external networks. In this latter case, encryption occurs only between an external host and the security gateway or between two security gateways. This setup relieves hosts on the internal network of the processing burden of encryption and simplifies the key distribution task by reducing the number of needed keys. Further, it thwarts traffic analysis based on ultimate destination.

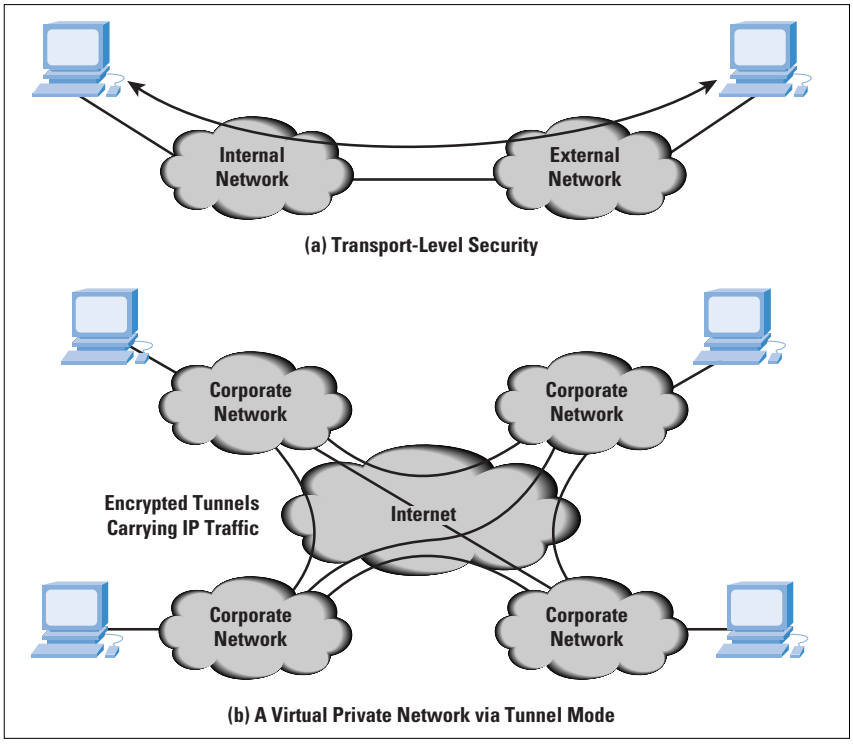
Let's use the diagram in Figure 1 as an example of how tunnel mode IPSec operates. The following steps occur for transfer of a transport-layer segment from the user system to one of the servers on one of the protected LANs.

- The user system prepares an inner IP packet with a destination address of the target host on the internal LAN. For a Telnet session, this packet would be a TCP packet with the original SYN flag set with a destination port set to 23. This entire IP packet is prefixed by an ESP header; then the packet and ESP trailer are encrypted and Authentication Data may be added. The Next Header field of the ESP header would be decimal 4 for IP-in-IP, indicating that the entire original IP packet is contained as the "payload." The resulting block is encapsulated with a new IP header (base header plus optional extensions such as routing and hop-by-hop options for IPv6) whose destination address is the firewall; this forms the outer IP packet. The Next Header field for this IP packet is 50 for ESP.
- The outer packet is routed to the destination firewall. Each intermediate router needs to examine and process the outer IP header plus any outer IP extension headers but does not need to examine the ciphertext.
- The destination firewall examines and processes the outer IP header plus any outer IP extension headers. Then, on the basis of the SPI in the ESP header, the gateway decrypts the remainder of the packet to recover the plaintext inner IP packet. This packet is then transmitted in the internal network.
- The inner packet is routed through zero or more routers in the internal network to the destination host. The receiver would have no indication that the packet had been encapsulated and protected by the "tunnel" between the user system and the gateway. It would see the packet as a request to start a Telnet session and would respond back with a TCP SYN/ACK, which would go back to the gateway. The gateway would encapsulate that packet into an IPSec packet and transport it back to the user system through this "tunnel." That return packet would be processed to find the original packet, which would contain the SYN/ACK for the Telnet session.

Common Uses of IPSec in Real Networks

Figure 5 shows two ways in which the IPSec ESP service can be used. In the upper part of the figure, encryption (and optionally authentication) is provided directly between two hosts. Figure 5b shows how tunnel mode operation can be used to set up a *Virtual Private Network* (VPN). In this example, an organization has four private networks interconnected across the Internet. Hosts on the internal networks use the Internet for transport of data but do not interact with other Internet-based hosts. By terminating the tunnels at the security gateway to each internal network, the configuration allows the hosts to avoid implementing the security capability. The former technique is supported by a transport mode SA, while the latter technique uses a tunnel mode SA.

Figure 5: Transport-Mode versus Tunnel-Mode Encryption



Key Management

The key management portion of IPSec involves the determination and distribution of secret keys. The IPSec Architecture document mandates support for two types of key management:

- *Manual:* A system administrator manually configures each system with its own keys and with the keys of other communicating systems. This is practical for small, relatively static environments.
- *Automated:* An automated system enables the on-demand creation of keys for SAs and facilitates the use of keys in a large distributed system with an evolving configuration. An automated system is the most flexible but requires more effort to configure and requires more software, so smaller installations are likely to opt for manual key management.

The default automated key management protocol for IPSec is referred to as *Internet Key Exchange* (IKE). IKE provides a standardized method for dynamically authenticating IPSec peers, negotiating security services, and generating shared keys. IKE has evolved from many different protocols and can be thought of as having two distinct capabilities. One of these capabilities is based on the *Internet Security Association and Key Management Protocol* (ISAKMP). ISAKMP provides a framework for Internet key management and provides the specific protocol support, including formats, for negotiation of security attributes. ISAKMP by itself does not dictate a specific key exchange algorithm; rather, ISAKMP consists of a set of message types that enable the use of a variety of key exchange algorithms. The actual key exchange mechanism in IKE is derived from Oakley and several other key exchange protocols that had been proposed for IPSec. Key exchange is based on the use of the Diffie-Hellman algorithm, but provides added security. In particular, Diffie-Hellman alone does not authenticate the two users that are exchanging keys, making the protocol vulnerable to impersonation. IKE includes mechanisms to authenticate the users.

Public Key Certificates

An important element of IPSec key management is the use of public key certificates. In essence, a public key certificate is provided by a trusted *Certificate Authority* (CA) to authenticate a user's public key. The essential elements include:

- Client software creates a pair of keys, one public and one private. The client prepares an unsigned certificate that includes a user ID and the user's public key. The client then sends the unsigned certificate to a CA in a secure manner.
- A CA creates a signature by calculating the hash code of the unsigned certificate and encrypting the hash code with the CA's private key; the encrypted hash code is the signature. The CA attaches the signature to the unsigned certificate and returns the now signed certificate to the client.
- The client may send its signed certificate to any other user. That user may verify that the certificate is valid by calculating the hash code of the certificate (not including the signature), decrypting the signature using the CA's public key, and comparing the hash code to the decrypted signature.

If all users subscribe to the same CA, then there is a common trust of that CA. All user certificates can be placed in the directory for access by all users. In addition, a user can transmit his or her certificate directly to other users. In either case, once B is in possession of A's certificate, B has confidence that messages it encrypts with A's public key will be secure from eavesdropping and that messages signed with A's private key are unforgeable.

If there is a large community of users, it may not be practical for all users to subscribe to the same CA. Because it is the CA that signs certificates, each participating user must have a copy of the CA's own public key to verify signatures. This public key must be provided to each user in an absolutely secure (with respect to integrity and authenticity) way so that the user has confidence in the associated certificates. Thus, with many users, it may be more practical for there to be many CAs, each of which securely provides its public key to some fraction of the users. In practice, there is not a single CA but rather a hierarchy of CAs. This complicates the problems of key distribution and of trust, but the basic principles are the same.

Whither IP Security

The driving force for the acceptance and deployment of secure IP is the need for business and government users to connect their private WAN/LAN infrastructure to the Internet for (1) access to Internet services and (2) use of the Internet as a component of the WAN transport system. Users need to isolate their networks and at the same time send and receive traffic over the Internet. The authentication and privacy mechanisms of secure IP provide the basis for a security strategy.

Because IP security mechanisms have been defined independent of their use with either the current IP or IPv6, deployment of these mechanisms does not depend on deployment of IPv6. Indeed, it is likely that we will see widespread use of secure IP features long before IPv6 becomes popular.

Recommended Web Sites

- The IPSec Working Group of the IETF. Charter for the group and latest RFCs and Internet Drafts for IPSec:
  `http://ietf.org/html.charters/ipsec-charter.html`
- IPSec Resources: List of companies implementing IPSec, implementation survey, and other useful material:
  `http://web.mit.edu/tytso/www/ipsec/index.html`

WILLIAM STALLINGS is a consultant, lecturer, and author of over a dozen books on data communications and computer networking. He has a Ph.D. in computer science from M.I.T. His latest book is *Local and Metropolitan Area Networks, Sixth Edition* (Prentice Hall, 2000). His home in cyberspace is **`WilliamStallings.com`** and he can be reached at **`ws@shore.net`**

Quality of Service—Fact or Fiction?

by Geoff Huston, Telstra

Much has been written about the potential of *Quality of Service* (QoS) and the Internet. However, much of the material is strong on promise, but falls short in critical analysis. In an effort to balance the picture, we present here a brief status report on the QoS effort, exposing some of the weaknesses in the current QoS architectures.

The QoS Service

The default service offering associated with the Internet is a *best-effort* service, where the network treats all traffic in exactly the same way. There is no consistent service outcome from the Internet best-effort service model. When the load level is low, the network delivers a high-quality service. The best-effort Internet does not deny entry to traffic, so as the load levels increase, the network congestion levels increase, and service-quality levels decline uniformly. This decline in service is experienced by all traffic passing through a congestion point, and is not limited to the most recently admitted traffic flows. For many applications, this best-effort response is perfectly acceptable. When network capacity is available, the application can make use of the resource, whereas when the level of contention for network bandwidth is high, each application will experience similar levels of congestion. A best-effort network service is a good match to opportunistic applications that can vary their data transfer rate in response to signaled network load.

The objective of various Internet QoS efforts is to augment this service with a number of selectable service responses. These service responses may be different from the best-effort service by some form of superior service response, such as lower delay, lower jitter, or greater bandwidth. These responses are relative, where the service outcome is claimed to be no worse than best effort at any time, and superior to best-effort under congestion load. Alternatively, QoS service responses may be distinguished by providing a consistent, and therefore predictable, service response that is unaffected by network congestion levels. These are quantitative service responses, where the characteristics of the service can be measured against a constant outcome. A quantitative service many be one that constrains jitter to a maximum level, or one that makes a certain bandwidth available, within parameters of bounded jitter, similar to a conventional leased line. Such constant-rate services may be superior to best-effort services when the network is under load, but they may also offer inferior service when the network is under negligible load. The essential attribute of these services is one of consistency.

Why is there a need for relative or consistent service profiles within the Internet? The underlying reasons for introducing QoS into the Internet appear to be threefold: First is the desire to provide high-quality support for IP voice and video services, second is the desire to manage the ser-

vice response provided to low-speed access devices, such as Internet mobile wireless devices, and third is the desire to provide a differentiated Internet access service, providing a network client with a range of service-quality levels at a range of prices.

Obviously this is a broad agenda, where there are requirements to extend specific network services to applications, requirements to adapt network services to particular transmission characteristics, and requirements to manage network resources to achieve particular response characteristics for an aggregated collection of traffic.

Approaches to QoS

The relevant efforts within the *Internet Engineering Task Force* (IETF) have been addressing standards for QoS mechanisms within the network.

The initial approach to QoS was that of the *Integrated Services* architecture. This approach focuses on the application as the trigger for QoS. Here, the application first signals its service requirements to the network in the form of a reservation, and the network responds to this request. The application proceeds only if the network has indicated that it is able to carry the additional load at the requested service level by committing to the reservation. The reservation remains in force until the application explicitly requests termination of the reservation, or the network signals to the application that it is unable to continue the reservation. The essential feature of this model is the "all-or-nothing" nature of the service model. Either the network commits to the reservation, in which case the application does not have to monitor the level of network response to the service, or the network indicates that it cannot meet the reservation. This approach imposes per-application state within the network, and for large-scale networks, such as the global Internet itself, this approach alone does not appear to be viable (see Figure 1).

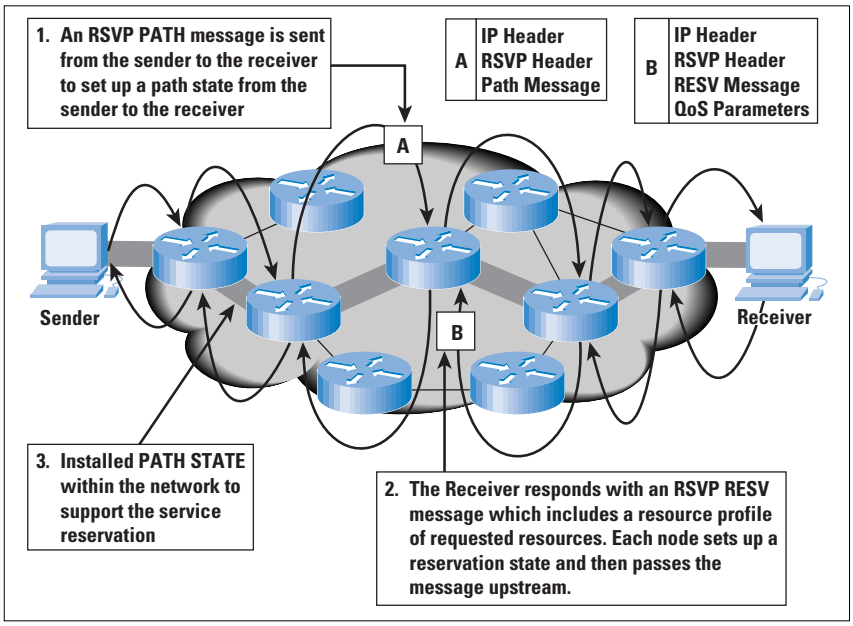


Figure 1: The Integrated Services QoS Architecture

The subsequent approach to QoS mechanisms has been to look at the core of the network, and examine those mechanisms that can provide differentiated service outcomes with appropriate scaling properties. This approach, the *Differentiated Services* architecture, includes dropping the concept of a per-application path state across the network using instead the concept of aggregated service mechanisms. Within the aggregated service model, the network provides a smaller number of different service classes and aggregates similar service demands from a set of applications into a single service class. Aggregated services are typically seen as an entry filter, where on entry to the network each packet is classified into a particular service profile. This classification is carried within the IP packet header, using 6 bits from the deprecated IP *Type of Service* (TOS) header to carry the service coding. The network then uses this service code in the packet header to treat this packet identically to all other packets within the same service code. While this approach does possess the ability to scale across the entire Internet, there are numerous unresolved issues relating to the quality signaling between individual applications and the network. The aggregated service model does not allow an individual application to sense if it is receiving the necessary service response from the network (see Figure 2).

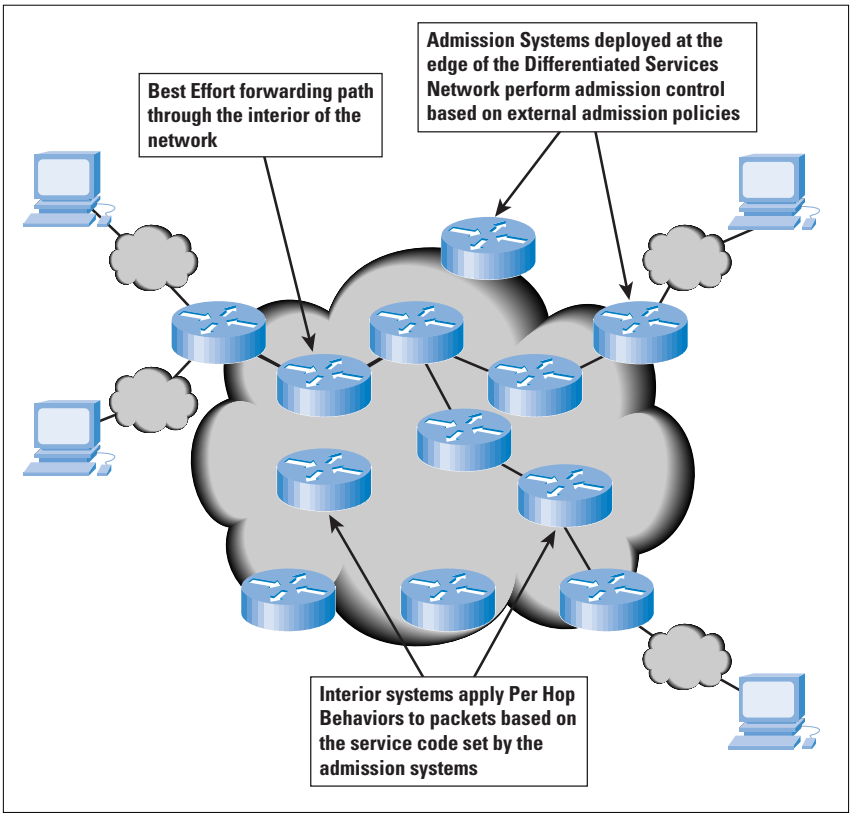


Figure 2: The Differentiated Services QoS Architecture

QoS Deployment

Neither approach alone is adequate to meet the QoS requirements. The Integrated Services approach alone imposes an excessive load in the core of large networks through the imposition of a per-application path state. The Differentiated Services approach does provide superior scaling properties through the use of aggregated service elements, but includes no concept of control signaling to inform the traffic conditioning elements of the current state of the network, or the current per-application requirements.

The underlying question then becomes: Is a combination of these two approaches sufficient to allow QoS to be widely deployed on the Internet?

At this stage the response does appear to be a "No." Perhaps this strong negative response should be further qualified. The existing tools are insufficient to support widespread use of QoS-based services on the multiprovider public Internet. The qualification is that within the enterprise network environment there are much stronger drivers for QoS mechanisms and much greater levels of administrative control over the overall network architecture, while within the multiprovider public Internet, these drivers are not apparent. The enterprise approach may also have some parallels within a single IP carrier's network, or even across some forms of bilateral agreements between carriers. However, such approaches are not anticipated to be a widespread feature of the public Internet service environment.

Let's look more closely at the public Internet and QoS to see why there is a mismatch between the two. The major stumbling blocks in attempting to address how QoS could be deployed in the public Internet are both engineering and economic in nature.

From an engineering perspective, we need to remember that in order to actually deliver any reasonable assurance of a quality-differentiated service, the service-quality mechanism chosen must be deployed across all networks along the end-to-end paths of the quality-service traffic. In a heterogeneous multiprovider environment such as the public Internet, this outcome is very unlikely. Within the tens of thousands of component service providers that make up the global Internet, such uniformity of action is highly improbable. The IPv6 transition structure correctly identifies the first step as isolated "islands" of IPv6 functionality, interconnected by some form of IPv6 "bridges." While the potential scenario of initial QoS deployment may be similar, in terms of isolated islands of deployment of QoS services, there is a much stricter requirement for the "bridges" across the non-QoS-aware parts of the network; namely, that they do not distort the service outcomes. In effect, this scenario requires a QoS response from a non-QoS system (see Figure 3).

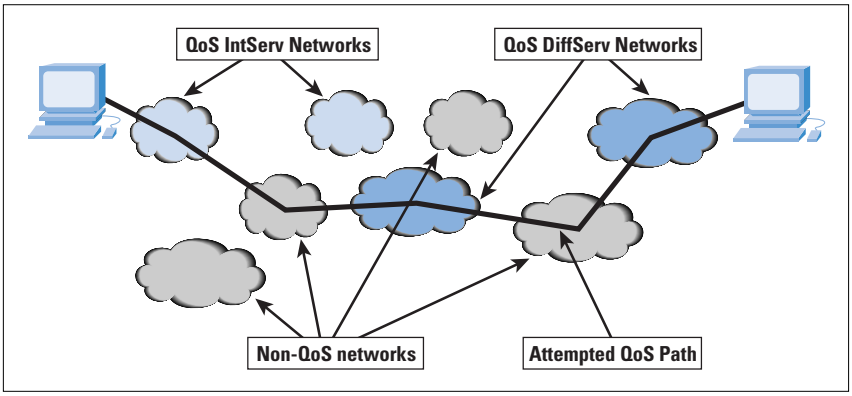


Figure 3: Attempted End-to-End QoS across the Public Internet

The engineering issues are deeper than simply the considerations of transition within a potential deployment scenario. The issues include:

- The need for QoS-enabled applications that can predict their service requirements in advance, and be able to signal these requirements into the network.
- In the case of the differentiated service approach of admission controls, there is a requirement for the interior of the network to be able to signal current load conditions to the network admission systems. This system also requires that the admission control points be able to use admission-decision support systems in order to include consideration of the service load, the current network load, and the policy parameters of the network that may allow some level of preemption of various admission decisions in order to meet high-priority service requirements.
- The signaling and negotiation aspect of QoS extends into the interdomain space, where two or more service providers need to negotiate mutually acceptable service profiles, and associated service access. This extends beyond the addition of bilateral agreements and encompasses the requirement to add QoS attributes to interdomain routing protocols. The tools and operating techniques required to support this functionality remain poorly defined.
- Measurement of service performance remains an area in which existing measurement tools are lacking. While it is possible to instrument every active device within a network into a network management system, such an element-by-element view does not readily translate to the end-to-end view of application service performance.

From an economic perspective, we must remember that no current Internet retail tariff includes a concept of end-to-end tariffed transactions. All tariffs are access based, because application transactions are not readily visible to the Internet network. In addition, no technically stable or financially stable structure of interprovider interconnection financial settlements exists today. The financial model of the Internet from an economic viewpoint is very polarized, with only customer and zero-dollar peer arrangements dominating the interprovider space. However, end-to-end QoS transactions demand a different economic model.

The initiator of the end-to-end QoS transaction has the discretion of choosing whether to request an end-to-end service profile. If such a profile is requested, the initiator should pay the initiating provider a retail tariff to cover the entire end-to-end cost of the transaction, and the initiating provider must then indicate a willingness to financially settle with transit peer networks in order for these transit peers to devote additional resources to service the traffic associated with this transaction, and so forth through the entire path of transit providers. The arbitrary nature of the Internet transits, the dynamic nature of routing, and the lack of transaction setups in any scalable form of QoS mechanisms make this entire scenario highly improbable within our current understanding of interprovider policy-management mechanisms.

The relatively loosely coordinated structure of the public Internet will have to change from the state we have today if we want to use QoS-based services. The changes include:

- A common selection of a set of QoS mechanisms to deploy,
- Ubiquitous deployment of these mechanisms across both service provider and client networks,
- The adoption of a uniform set of retail tariffs for QoS services,
- The definition and common acceptance of multi-party QoS-related financial settlements that support fair and equitable cost distribution among multiple providers, and
- The definition of commonly accepted service performance metrics and related measurement methodologies to allow end-to-end and network-by-network service outcomes to be objectively assessed.

This is a significant agenda for the industry at large to undertake, and more so in an environment that features diversity and vigorous competition between various public Internet service providers.

An additional factor is also working against QoS deployment in the public Internet space. The increasing availability of very-high-speed transmission systems is bringing network carriage capacity down to the level of an abundant commodity across large parts of the Internet world. As the unit costs of network capacity decline in the face of increasing levels of availability of transmission systems, the market niche that QoS could occupy in managing a scarce resource is shrinking. The driver for QoS deployment is not that the best-effort service is not good enough. The problem that QoS is attempting to address is one of allocation of network capacity at those points in time when the network is under heavy load, or, in other words, taking on the task of allocating capacity when there is not enough network capacity to meet every demand. When a network is under load, the QoS response is to place additional control functionality in both applications and in the network to manage this allocation function. Obviously such an activity imposes additional costs on the network operators and the network client. Such additional costs have not created any additional network capacity.

The total sum of demand remains in excess of capacity after the deployment of QoS mechanisms. The alternative approach is to incur additional cost by augmenting the capacity of the network. This approach minimizes the impact of load on the network causing disruption to individual transactions. Again this approach imposes additional costs onto the network, but in an environment of abundant transmission capacity, it may often be the more cost-effective approach.

Where does this leave QoS and the public Internet? There is no doubt that QoS is a very stimulating area of research, with much to offer the enterprise network environment, but in asking for QoS to be deployed within the existing incarnation of the public multiprovider Internet, we may be simply asking for too much at this point in time. More effort is required to turn a QoS Internet into a reliable production platform.

Further Reading

[1] Huston, G., *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, John Wiley & Sons, January 2000.

A detailed examination of Internet Quality of Service technologies and their potential application within the Internet.

[2] Kilkki, K., *Differentiated Services for the Internet,* ISBN 1578701325, Macmillan Technical Publishing, June 1999.

An in-depth look at the Differentiated Services architecture and its use in enabling networks to handle traffic classes in a specific manner.

[3] Durham, D., and Yavatar, R., *Inside the Internet's Resource Reservation Protocol: Foundations for Quality of Service,* ISBN 0471322148, John Wiley & Sons, April 1999.

At the core of the Integrated Services architecture is a signaling protocol to undertake service reservations. The Resource ReSerVation Protocol (RSVP) is a signaling protocol that can undertake this role. This book describes both the Integrated Services architecture and RSVP in detail.

[4] Odlyzko, A., "The Economics of the Internet: Utility, Utilization, Pricing, and Quality of Service," 1998. Available at:
www.research.att.com/~amo

A paper arguing the point of view that overprovisioning data networks is a viable and economically sustainable response to the demands for service quality within data networks, and that such a response is technically and economically superior to implementing QoS responses within the network.

[5] Braden, R., Clark, D., and Shenker, S., "Integrated Services in the Internet Architecture: An Overview," RFC 1633, June 1994.

This RFC describes the components of the Integrated Services architecture, a proposed extension to the Internet architecture, and protocols to support real-time traffic flows through service-quality commitments.

[6] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W., "An Architecture for Differentiated Services," RFC 2475, Proposed Standard, December 1998.

The architecture description for the Differentiated Services enhancements to the Internet Protocol. This architecture achieves scalability by aggregating traffic classification state, which is conveyed by means of IP-layer packet marking using the Differentiated Services (DS) field. Packets are classified and marked to receive a particular per-hop forwarding behavior on nodes along their path. Sophisticated classification, marking, policing, and shaping operations need to be implemented only at network boundaries or hosts. Network resources are allocated to traffic streams by service-provisioning policies that govern how traffic is marked and conditioned upon entry to a differentiated services-capable network, and how that traffic is forwarded within that network.

[7] Gray, T., "Enterprise QoS Survival Guide: 1999 Edition," 1999. Available at:
http://staff.washington.edu/gray/papers/eqos22.html

A detailed view of an approach to supporting QoS in an enterprise environment. The paper is an excellent example of the procedural steps involved in network engineering, detailing the intended environment, the available tools and the desired outcomes, and then examining the viability of a number of QoS solutions.

[8] Huston, G., "Next Steps for the IP QoS Architecture." Available at:
www.ietf.org/internet-drafts/draft-iab-qos-00.txt

While there has been significant progress in the definition of IP QoS architecture, there are a number of aspects of QoS that appear to need further elaboration as they relate to translating a set of tools into a coherent platform for end-to-end service delivery. This document highlights the outstanding issues relating to the deployment and use of QoS mechanisms within the Internet, noting those areas where further standards work may be required. This draft is a work item of the Internet Architecture Board Working Group of the IETF.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Technologist in the Internet area for Telstra. He is also an active member of the IETF, and is the chair of the Internet Society Board of Trustees. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089 and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@telstra.net**

Book Review

Removing the Spam

Removing the Spam: Email Processing and Filtering, Geoff Mulligan, ISBN 0-201-37957-0, Addison-Wesley, 1999.
http://cseng.aw.com/bookdetail.qry?ISBN=0-201-37957-0&ptype=0

Do not be fooled by the title of this book. You might purchase this book, part of the Addison-Wesley Networking Basics Series, thinking you are just getting information dealing with unsolicited commercial e-mail (commonly called, to Hormel's displeasure, "spam"). The title is probably the work of a marketeer who thought "spam" in the title would *sell!* The subtitle really describes the meat of the matter. This short, but thorough, book is about e-mail processing and filtering—dealing with spam, yes, but so much more.

A collection of essential information for the Internet e-mail "gatekeeper," *Removing the Spam* is really geared for the gatekeeper using a UNIX-based system, so NT system administrators be forewarned. Being an e-mail gatekeeper on the Internet involves keeping the e-mail flowing, making sure the automated processes in place do the job, supporting e-mail "mailing lists," and providing the services and features your users want or need for e-mail processing.

Commercial products support some of the many requirements, but the best software for most of these functions is freely available on the Internet. Geoff provides answers to the requirements using the most popular and commonly used solutions: *Sendmail* for mail delivery, *procmail* for e-mail filtering, and *majordomo* and *smartlist* for mailing-list management.

The book, however, tries to do a bit too much. Geoff indicates that the intended audience is not only the system administrator, but also the e-mail end users wanting to filter their own personal e-mail as well as those who want to run their own mailing list. Because of this broad audience, there are times when the book delves too long in the basics, giving the impression of topics added to lengthen the book. The overview of IP protocols, the brief history of the Internet, suggestions for users dealing with spammers, and mailing-list etiquette are examples that come to mind. Nevertheless, the other topics covered are "net essentials," and worth skimming over the already known.

The book clearly defines spam and its evils, and presents the tools and techniques available for removing, or at least minimizing, the spam. It is probably too ambitious when covering e-mail forgery and tracing e-mail spam, but leaves no essential unmentioned.

Sendmail coverage is good, dealing with installation as well as configuration, highlighting antispam features, and how to use them. Though not covering as much detail as other books that focus on Sendmail, the important elements of building and modifying are handled, as well as Sendmail's use of data bases, including the infamous "Realtime Blackhole List" (`http://maps.vix.com/rbl/`).

The e-mail gatekeeper, as well as end users of e-mail, can use procmail to preprocess e-mail before final delivery. Procmail is powerful and flexible, and, so, can be difficult to configure properly. Configuration files examples with explanations allow even the procmail-savvy reader to learn and try something new.

The mailing list section again instructs both system administrator and user. Information about subscribing, unsubscribing, and getting information from the mailing list software is useful for the user. The administrator will appreciate the examples of getting, installing, configuring, and running majordomo and smartlist. Geoff gives suggestions about when a manual versus automated solution is best.

About the Author

I knew Geoff back in our Digital Equipment Corporation days when he worked in the Network Systems Lab. My group ran one of the corporate Internet gateways, modeled after the one at NSL. Further, the group I ran also productized and delivered what is arguably the first commercial Internet firewall, based on a design from the team at NSL. All this to say, Geoff certainly has the background to write about these topics. Since those days, Geoff has been busy with other Internet endeavors, such as starting USA.NET and creating the NetAddress product (permanent, follow-you-anywhere e-mail addresses) and helping develop the Sun Microsystems Sunscreen Firewall. He also founded Geocast Network Systems. In various roles, in differing capacities, Geoff has had to wrestle with the matters covered in his book. What he writes is based on experience learned in the danger zone of the Internet gateway.

Organization

The book is divided into four chapters. The first chapter, the introduction really, is strangely entitled "The Dawn of Electronic Mail." This is also the "roughest" chapter. It is difficult to understand why some topics are covered in the order that they are here (and why some are covered at all—the aforementioned "list etiquette" and "Size and Growth of the Internet," for example). It introduces (needlessly, I think) The Internet Protocols, but then reviews the basics of understanding e-mail systems. It introduces spam, along with antispam resources, and the topics in the rest of the book to be covered in detail: e-mail processing, filtering, and e-mail lists.

Chapter 2 is entitled "Sendmail" and covers obtaining, installing, configuring, and running Sendmail on a UNIX machine. It gives the commands to build and install Sendmail and your Sendmail configuration file. This coverage is not detailed enough for *every* situation, but gives the most common configuration information, which should satisfy most readers' needs. Included are instructions for using Sendmail to help stop (or avert) spam at the mail gateway.

Chapter 3 unravels the mysteries behind procmail configuration for e-mail filtering. This chapter covers getting the software, installing it, and using procmail—the latter for system administrators and users alike. There are example "ready-to-run filters" included. Caveat: Some of the scripts have inherent errors. No doubt these errors are unfortunate publication glitches, but they do detract from the usefulness of this chapter. Geoff has compiled an errata list with corrected scripts. This can be found at: **`http://www.hz.com/spam/eratta`**

Chapter 4 covers mailing lists, specifically discussing administering them "by hand" (just using Sendmail) or "automatically" (majordomo and smartlist). Again, examples are given with step-by-step commands.

Closing Thoughts

Production errors aside (the serious ones in the procmail chapter and others that are just nits to pick—the "P" in ARPA stands for "Projects," not "Project"), this book is useful as an introduction as well as a reminder of things forgotten. I can recommend this book to the novice or seasoned e-mail gatekeeper, and I will recommend it to the students in my Sendmail courses.

—Frederick M. Avolio, Avolio Consulting
`fred@avolio.com`

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **`ipj@cisco.com`** for more information.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

Fragments

ICANN Launches Membership Web Site for Individual Internet Users

The *Internet Corporation for Assigned Names and Numbers* (ICANN) recently announced the launch of its At Large Membership Web site. After considerable public input, the ICANN Board has developed this program as a new way for Internet users from all over the globe to participate directly in the ICANN process. Individuals can register to become ICANN members at **http://members.icann.org**

The At Large Membership of ICANN will give individual members of Internet communities worldwide a voice in the selection of Directors to the ICANN Board. By becoming an ICANN member, individuals will have an opportunity to become part of the ICANN "bottom-up" approach to making policy concerning Internet names and addresses. The basic requirements for applying to become an ICANN At Large member are: The completion of an online membership application, a working Internet e-mail address, and a single physical residence verified by a postal mail address. Thanks to a grant from the Markle Foundation, the initial launch of ICANN's At Large Membership program has been funded without the need for membership dues.

The ICANN Board will consider and adopt further policy about composition and structure of the At Large Membership, and establish rules for the nomination and election of candidates for the At Large Council. It is hoped that the target goal of 5,000 members can be reached in the next few weeks in order to move forward with the At Large Elections later this year.

ICANN is a non-profit, international corporation formed to oversee a select set of Internet technical management functions currently managed by the U.S. Government, or by its contractors and volunteers. Specifically, ICANN is assuming responsibility for coordinating the management of the *Domain Name System* (DNS), the allocation of IP address space, the assignment of protocol parameters, and the management of the root server system.

Online Registration for INET 2000 Now Open

INET 2000, the annual conference of the Internet Society (ISOC) will be held in Yokohama, Japan, July 18–21. You can register for this event by visiting ISOC's Web site at:
http://www.isoc.org/inet2000/register.shtml

Denial of Service Attacks

In early February, several high-profile Internet Web sites were severely disrupted by a number of so-called *Distributed Denial of Service* (DDoS) attacks. We plan to publish an article on this topic in the future. Meanwhile, we recommend you visit the Denial of Service Resource Page at **http://www.denialinfo.com/**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

June 2000 Volume 3, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Two protocols used in the Internet are so important that they deserve special attention: the *Internet Protocol* (IP) from which this journal takes its name, and the *Transmission Control Protocol* (TCP). IP is fundamental to Internet addressing and routing, while TCP provides a reliable transport service that is used by most Internet applications, including interactive Telnet, file transfer, electronic mail, and Web page access via HTTP. Because of the critical importance of TCP to the operation of the Internet, it has received much attention in the research community over the years. As a result, numerous improvements to implementations of TCP have been developed and deployed. In this issue, Geoff Huston takes a detailed look at TCP from a performance perspective and describes several enhancements to the original protocol. In a second article, Geoff will look at the challenges facing TCP in a rapidly growing and changing Internet, and describe work to further augment TCP.

Electronic mail is by far the most used of all Internet applications. The fundamental protocols for delivery and retrieval of e-mail have not changed much since the early days of the ARPANET, but as with TCP, many enhancements have been added to accommodate new uses of e-mail. Today, Internet e-mail supports international character sets, includes the ability to send file attachments, and allows roaming e-mail clients to authenticate themselves to servers. All of this has been made possible by continued development in the *Internet Engineering Task Force* (IETF). In our second article, Paul Hoffman of the Internet Mail Consortium gives an overview of Internet mail standards.

This is the second anniversary issue of *The Internet Protocol Journal* (IPJ). By now more than 10,000 people from virtually every country in the world have subscribed to the paper edition of IPJ. In order to serve our readers better, we are developing an online subscription system, which will be deployed in July 2000. With this new system you will be able to modify your mailing address as well as select your preferred delivery method for the journal. You can choose to receive IPJ on paper, or be notified via e-mail when a new issue becomes available on line. More information about this new system can be found on our Web site at **www.cisco.com/ipj**. We would love to hear your feedback on this system and any other aspect of IPJ. Please send your comments to **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

TCP Performance

by Geoff Huston, Telstra

The *Transmission Control Protocol* (TCP) and the *User Datagram Protocol* (UDP) are both IP transport-layer protocols. UDP is a lightweight protocol that allows applications to make direct use of the unreliable datagram service provided by the underlying IP service. UDP is commonly used to support applications that use simple query/response transactions, or applications that support real-time communications. TCP provides a reliable data-transfer service, and is used for both bulk data transfer and interactive data applications. TCP is the major transport protocol in use in most IP networks, and supports the transfer of over 90 percent of all traffic across the public Internet today. Given this major role for TCP, the performance of this protocol forms a significant part of the total picture of service performance for IP networks. In this article we examine TCP in further detail, looking at what makes a TCP session perform reliably and well. This article draws on material published in the *Internet Performance Survival Guide*[1].

Overview of TCP

TCP is the embodiment of reliable end-to-end transmission functionality in the overall Internet architecture. All the functionality required to take a simple base of IP datagram delivery and build upon this a control model that implements reliability, sequencing, flow control, and data streaming is embedded within TCP[2].

TCP provides a communication channel between processes on each host system. The channel is reliable, full-duplex, and streaming. To achieve this functionality, the TCP drivers break up the session data stream into discrete segments, and attach a TCP header to each segment. An IP header is attached to this TCP packet, and the composite packet is then passed to the network for delivery. This TCP header has numerous fields that are used to support the intended TCP functionality. TCP has the following functional characteristics:

- *Unicast protocol:* TCP is based on a unicast network model, and supports data exchange between precisely two parties. It does not support broadcast or multicast network models.
- *Connection state:* Rather than impose a state within the network to support the connection, TCP uses synchronized state between the two endpoints. This synchronized state is set up as part of an initial connection process, so TCP can be regarded as a connection-oriented protocol. Much of the protocol design is intended to ensure that each local state transition is communicated to, and acknowledged by, the remote party.
- *Reliable:* Reliability implies that the stream of octets passed to the TCP driver at one end of the connection will be transmitted across the network so that the stream is presented to the remote process as the same sequence of octets, in the same order as that generated by the sender.

This implies that the protocol detects when segments of the data stream have been discarded by the network, reordered, duplicated, or corrupted. Where necessary, the sender will retransmit damaged segments so as to allow the receiver to reconstruct the original data stream. This implies that a TCP sender must maintain a local copy of all transmitted data until it receives an indication that the receiver has completed an accurate transfer of the data.

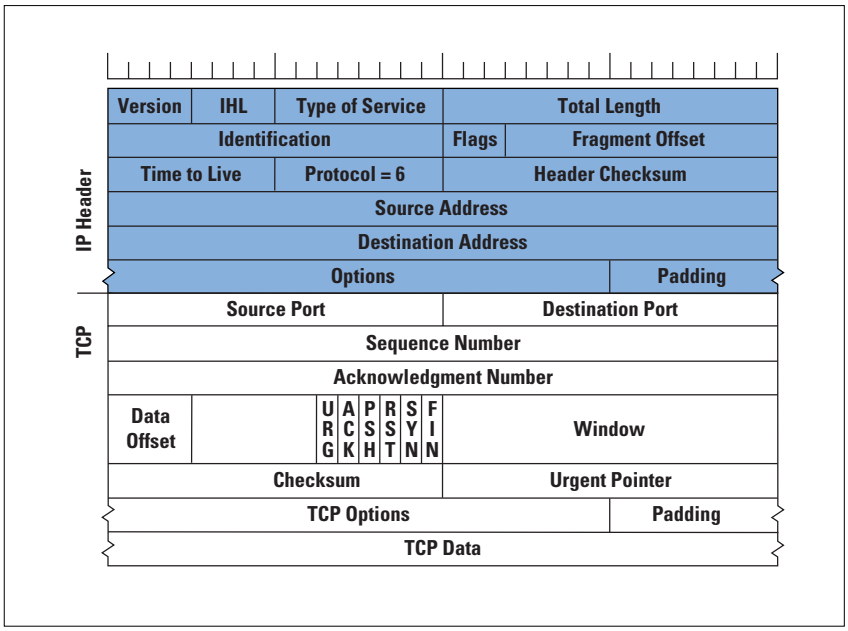
- *Full duplex:* TCP is a full-duplex protocol; it allows both parties to send and receive data within the context of the single TCP con–nection.
- *Streaming:* Although TCP uses a packet structure for network transmission, TCP is a true streaming protocol, and application-level network operations are not transparent. Some protocols explicitly encapsulate each application transaction; for every *write,* there must be a matching *read*. In this manner, the application-derived segmentation of the data stream into a logical record structure is preserved across the network. TCP does not preserve such an implicit structure imposed on the data stream, so that there is no pairing between *write* and *read* operations within the network protocol. For example, a TCP application may *write* three data blocks in sequence into the network connection, which may be collected by the remote reader in a single *read* operation. The size of the data blocks (segments) used in a TCP session is negotiated at the start of the session. The sender attempts to use the largest segment size it can for the data transfer, within the constraints of the maximum segment size of the receiver, the maximum segment size of the configured sender, and the maximum supportable non-fragmented packet size of the network path (path *Maximum Transmission Unit* [MTU]). The path MTU is refreshed periodically to adjust to any changes that may occur within the network while the TCP connection is active.
- *Rate adaptation:* TCP is also a rate-adaptive protocol, in that the rate of data transfer is intended to adapt to the prevailing load conditions within the network and adapt to the processing capacity of the receiver. There is no predetermined TCP data-transfer rate; if the network and the receiver both have additional available capacity, a TCP sender will attempt to inject more data into the network to take up this available space. Conversely, if there is congestion, a TCP sender will reduce its sending rate to allow the network to recover. This adaptation function attempts to achieve the highest possible data-transfer rate without triggering consistent data loss.

The TCP Protocol Header

The TCP header structure, shown in Figure 1, uses a pair of 16-bit source and destination *Port* addresses. The next field is a 32-bit *sequence number,* which identifies the sequence number of the first data octet in this packet. The sequence number does not start at an initial value of 1 for each new TCP connection; the selection of an initial value is critical, because the initial value is intended to prevent delayed data

from an old connection from being incorrectly interpreted as being valid within a current connection. The sequence number is necessary to ensure that arriving packets can be ordered in the sender's original order. This field is also used within the flow-control structure to allow the association of a data packet with its corresponding acknowledgement, allowing a sender to estimate the current round-trip time across the network.

Figure 1: The TCP/IP Datagram



The *acknowledgment sequence number* is used to inform the remote end of the data that has been successfully received. The acknowledgment sequence number is actually one greater than that of the last octet correctly received at the local end of the connection. The *data offset* field indicates the number of four-octet words within the TCP header. Six single *bit flags* are used to indicate various conditions. URG is used to indicate whether the *urgent pointer* is valid. ACK is used to indicate whether the *acknowledgment* field is valid. PSH is set when the sender wants the remote application to *push* this data to the remote application. RST is used to *reset* the connection. SYN (for *synchronize)* is used within the connection startup phase, and FIN (for *finish*) is used to close the connection in an orderly fashion. The *window* field is a 16-bit count of available buffer space. It is added to the acknowledgment sequence number to indicate the highest sequence number the receiver can accept. The TCP *checksum* is applied to a synthesized header that includes the source and destination addresses from the outer IP datagram. The final field in the TCP header is the *urgent pointer*, which, when added to the sequence number, indicates the sequence number of the final octet of urgent data if the urgent flag is set.

Many options can be carried in a TCP header. Those relevant to TCP performance include:

- *Maximum-receive-segment-size option:* This option is used when the connection is being opened. It is intended to inform the remote end of the maximum segment size, measured in octets, that the sender is willing to receive on the TCP connection. This option is used only in the initial SYN packet (the initial packet exchange that opens a TCP connection). It sets both the maximum receive segment size and the maximum size of the advertised TCP window, passed to the remote end of the connection. In a robust implementation of TCP, this option should be used with path MTU discovery to establish a segment size that can be passed across the connection without fragmentation, an essential attribute of a high-performance data flow.
- *Window-scale option:* This option is intended to address the issue of the maximum window size in the face of paths that exhibit a high-delay bandwidth product. This option allows the window size advertisement to be right-shifted by the amount specified (in binary arithmetic, a right-shift corresponds to a multiplication by 2). Without this option, the maximum window size that can be advertised is 65,535 bytes (the maximum value obtainable in a 16-bit field). The limit of TCP transfer speed is effectively one window size in transit between the sender and the receiver. For high-speed, long-delay networks, this performance limitation is a significant factor, because it limits the transfer rate to at most 65,535 bytes per round-trip interval, regardless of available network capacity. Use of the window-scale option allows the TCP sender to effectively adapt to high-bandwidth, high-delay network paths, by allowing more data to be held in flight. The maximum window size with this option is 2^{30} bytes. This option is negotiated at the start of the TCP connection, and can be sent in a packet only with the SYN flag. Note that while an MTU discovery process allows optimal setting of the maximum-receive-segment-size option, no corresponding bandwidth delay product discovery allows the reliable automated setting of the window-scale option[3].
- *SACK-permitted option and SACK option:* This option alters the acknowledgment behavior of TCP. SACK is an acronym for *selective acknowledgment*. The SACK-permitted option is offered to the remote end during TCP setup as an option to an opening SYN packet. The SACK option permits selective acknowledgment of permitted data. The default TCP acknowledgment behavior is to acknowledge the highest sequence number of in-order bytes. This default behavior is prone to cause unnecessary retransmission of data, which can exacerbate a congestion condition that may have been the cause of the original packet loss. The SACK option allows the receiver to modify the acknowledgment field to describe noncontinuous blocks of received data, so that the sender can retransmit only what is missing at the receiver's end[4].

Any robust high-performance implementation of TCP should negotiate these parameters at the start of the TCP session, ensuring the following: that the session is using the largest possible IP packet size that can be carried without fragmentation, that the window sizes used in the transfer are adequate for the bandwidth-delay product of the network path, and that selective acknowledgment can be used for rapid recovery from line-error conditions or from short periods of marginally degraded network performance.

TCP Operation

The first phase of a TCP session is establishment of the connection. This requires a *three-way handshake*, ensuring that both sides of the connection have an unambiguous understanding of the sequence number space of the remote side for this session. The operation of the connection is as follows:

- The local system sends the remote end an initial sequence number to the remote port, using a SYN packet.
- The remote system responds with an ACK of the initial sequence number and the initial sequence number of the remote end in a response SYN packet.
- The local end responds with an ACK of this remote sequence number.

The connection is opened.

The operation of this algorithm is shown in Figure 2. The performance implication of this protocol exchange is that it takes one and a half *round-trip times* (RTTs) for the two systems to synchronize state before any data can be sent.

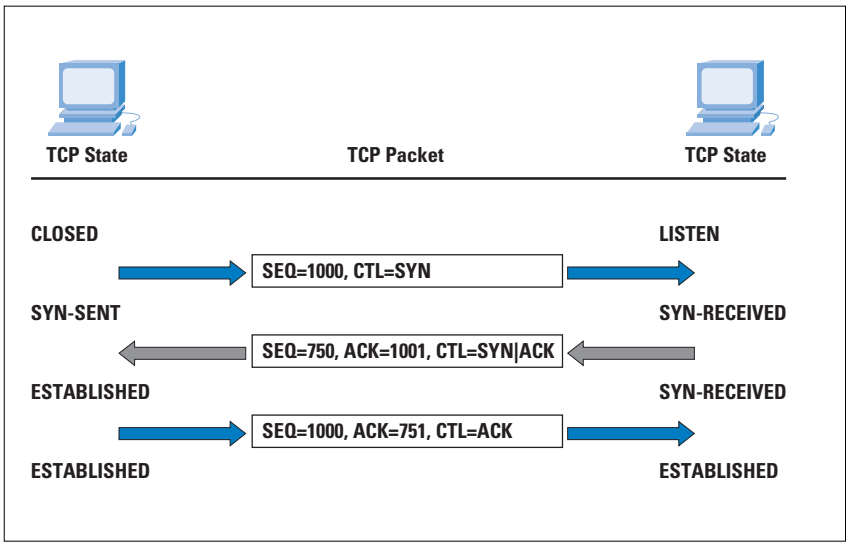


Figure 2: TCP Connection Handshake

After the connection has been established, the TCP protocol manages the reliable exchange of data between the two systems. The algorithms that determine the various retransmission timers have been redefined numerous times. TCP is a *sliding-window* protocol, and the general principle of flow control is based on the management of the advertised window size and the management of retransmission timeouts, attempting to optimize protocol performance within the observed delay and loss parameters of the connection. Tuning a TCP protocol stack for optimal performance over a very low-delay, high-bandwidth LAN requires different settings to obtain optimal performance over a dialup Internet connection, which in turn is different for the requirements of a high-speed wide-area network. Although TCP attempts to discover the delay bandwidth product of the connection, and attempts to automatically optimize its flow rates within the estimated parameters of the network path, some estimates will not be accurate, and the corresponding efforts by TCP to optimize behavior may not be completely successful.

Another critical aspect is that TCP is an adaptive flow-control protocol. TCP uses a basic flow-control algorithm of increasing the data-flow rate until the network signals that some form of saturation level has been reached (normally indicated by data loss). When the sender receives an indication of data loss, the TCP flow rate is reduced; when reliable transmission is reestablished, the flow rate slowly increases again.

If no reliable flow is reestablished, the flow rate backs further off to an initial probe of a single packet, and the entire adaptive flow-control process starts again.

This process has numerous results relevant to service quality. First, TCP behaves *adaptively*, rather than *predictively*. The flow-control algorithms are intended to increase the data-flow rate to fill all available network path capacity, but they are also intended to quickly back off if the available capacity changes because of interaction with other traffic, or if a dynamic change occurs in the end-to-end network path. For example, a single TCP flow across an otherwise idle network attempts to fill the network path with data, optimizing the flow rate within the available network capacity. If a second TCP flow opens up across the same path, the two flow-control algorithms will interact so that both flows will stabilize to use approximately half of the available capacity per flow. The objective of the TCP algorithms is to adapt so that the network is fully used whenever one or more data flows are present. In design, tension always exists between the efficiency of network use and the enforcement of predictable session performance. With TCP, you give up predictable throughput but gain a highly utilized, efficient network.

Protocol Performance

In this section we examine the transfer of data using the TCP protocol, focusing on the relationship between the protocol and performance. TCP is generally used within two distinct application areas: short-delay short data packets sent on demand, to support interactive applications such as *Telnet*, or *rlogin*, and large packet data streams supporting reliable volume data transfers, such as mail transfers, Web-page transfers, and *File Transfer Protocol* (FTP). Different protocol mechanisms come into play to support interactive applications, as distinct from short- and long-held volume transactions.

Interactive TCP

Interactive protocols are typically directed at supporting single-character interactions, where each character is carried in a single packet, as is its echo. The protocol interaction to support this is indicated in Figure 3. These 2 bytes of data generate four TCP/IP packets, or 160 bytes of protocol overhead. TCP makes some small improvement in this exchange through the use of *piggybacking*, where an ACK is carried in the same packet as the data, and *delayed acknowledgment*, where an ACK is delayed up to 200 ms before sending, to give the server application the opportunity to generate data that the ACK can piggyback. The resultant protocol exchange is indicated in Figure 4.

Figure 3: Interactive Exchange

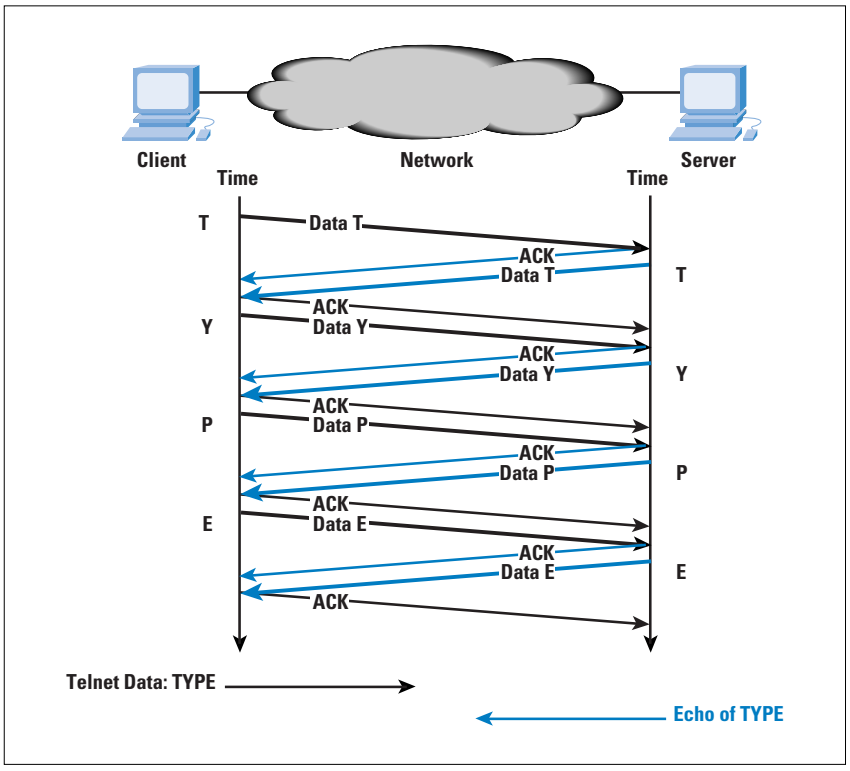
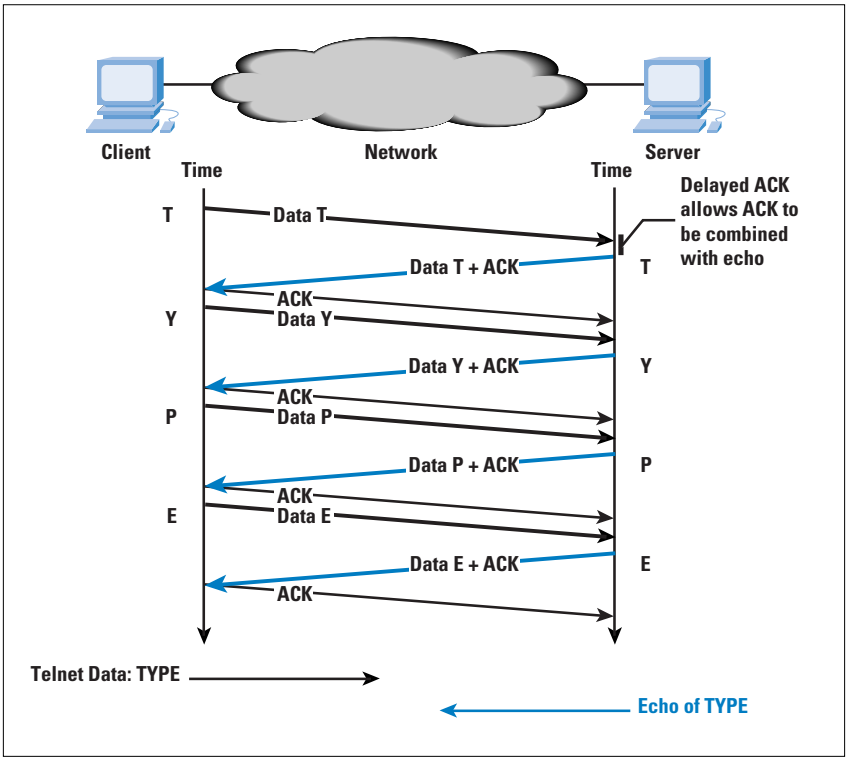


Figure 4:
Interactive Exchange
with Delayed ACK



For short-delay LANs, this protocol exchange offers acceptable performance. This protocol exchange for a single data character and its echo occurs within about 16 ms on an Ethernet LAN, corresponding to an interactive rate of 60 characters per second. When the network delay is increased in a WAN, these small packets can be a source of congestion load. The TCP mechanism to address this small-packet congestion was described by John Nagle in RFC 896[5]. Commonly referred to as the *Nagle Algorithm*, this mechanism inhibits a sender from transmitting any additional small segments while the TCP connection has outstanding unacknowledged small segments. On a LAN, this modification to the algorithm has a negligible effect; in contrast, on a WAN, it has a dramatic effect in reducing the number of small packets in direct correlation to the network path congestion level (as shown in Figures 5 and 6). The cost is an increase in session jitter by up to a round-trip time interval. Applications that are jitter-sensitive typically disable this control algorithm.

TCP is not a highly efficient protocol for the transmission of interactive traffic. The typical carriage efficiency of the protocol across a LAN is 2 bytes of payload and 120 bytes of protocol overhead. Across a WAN, the Nagle algorithm may improve this carriage efficiency slightly by increasing the number of bytes of payload for each payload transaction, although it will do so at the expense of increased session jitter.

Figure 5: WAN Interactive Exchange

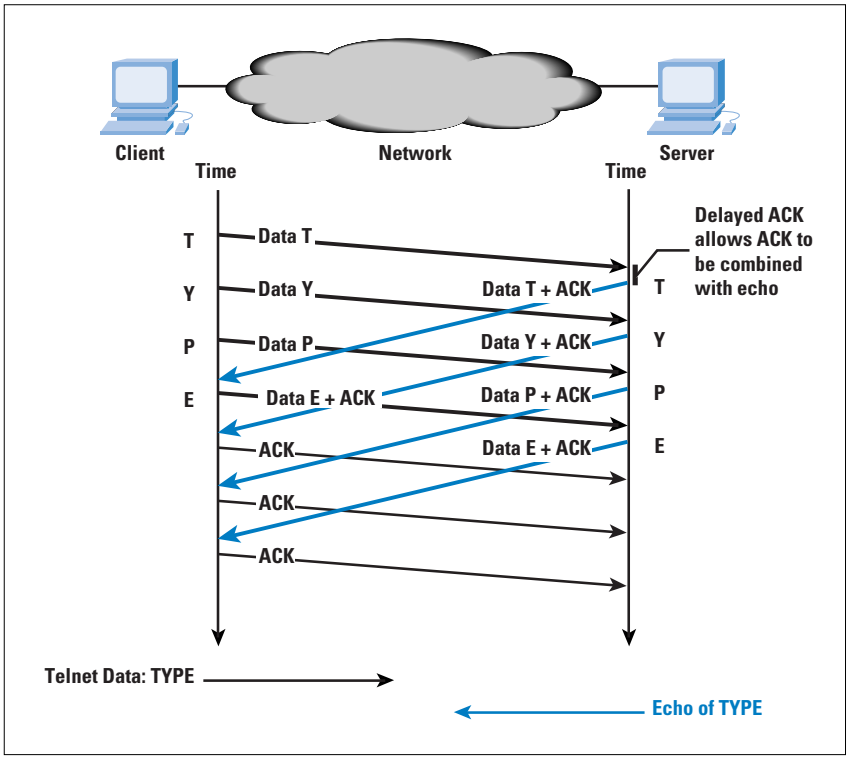
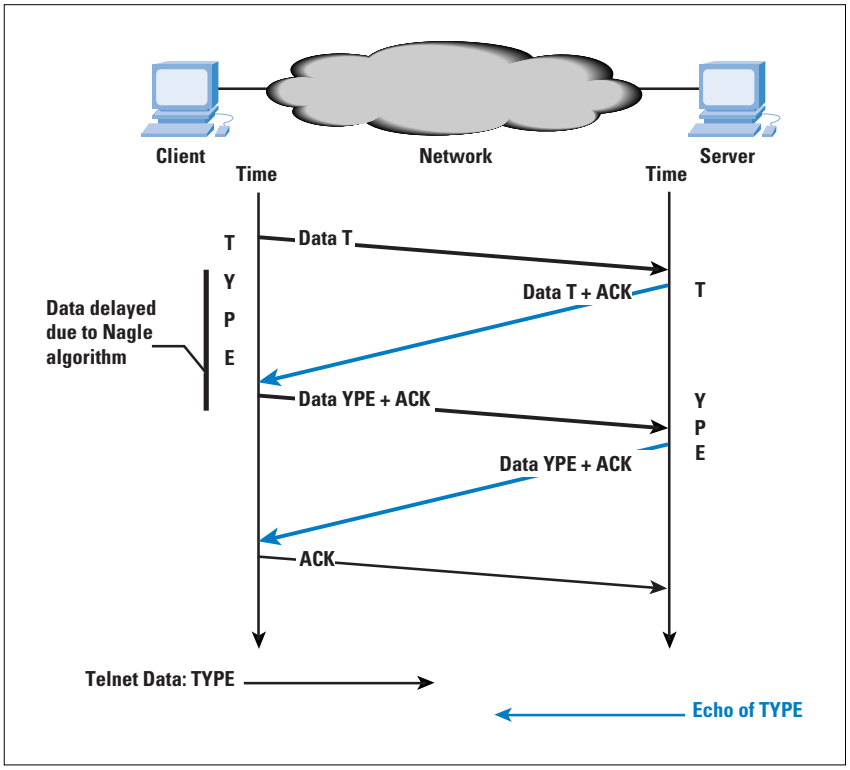


Figure 6: WAN Interactive Exchange with Nagle Algorithm



TCP Volume Transfer

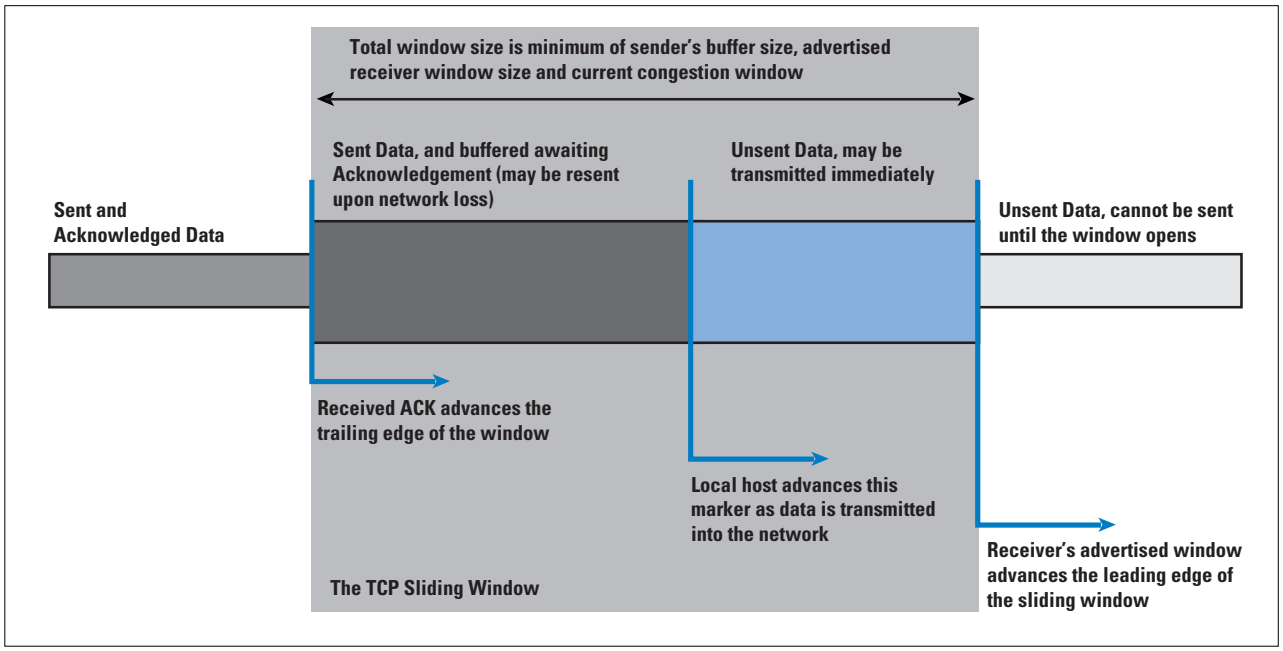
The objective for this application is to maximize the efficiency of the data transfer, implying that TCP should endeavor to locate the point of dynamic equilibrium of maximum network efficiency, where the sending data rate is maximized just prior to the onset of sustained packet loss.

Further increasing the sending rate from such a point will run the risk of generating a congestion condition within the network, with rapidly increasing packet-loss levels. This, in turn, will force the TCP protocol to retransmit the lost data, resulting in reduced data-transfer efficiency. On the other hand, attempting to completely eliminate packet-loss rates implies that the sender must reduce the sending rate of data into the network so as not to create transient congestion conditions along the path to the receiver. Such an action will, in all probability, leave the network with idle capacity, resulting in inefficient use of available network resources.

The notion of a point of equilibrium is an important one. The objective of TCP is to coordinate the actions of the sender, the network, and the receiver so that the network path has sufficient data such that the network is not idle, but it is not so overloaded that a congestion backlog builds up and data loss occurs. Maintaining this point of equilibrium requires the sender and receiver to be synchronized so that the sender passes a packet into the network at precisely the same time as the receiver removes a packet from the network. If the sender attempts to exceed this equilibrium rate, network congestion will occur. If the sender attempts to reduce its rate, the efficiency of the network will drop.

TCP uses a sliding-window protocol to support bulk data transfer (Figure 7). The receiver advertises to the sender the available buffer space at the receiver. The sender can transmit up to this amount of data before having to await a further buffer update from the receiver. The sender should have no more than this amount of data in transit in the network. The sender must also buffer sent data until it has been ACKed by the receiver. The send window is the minimum of the sender's buffer size and the advertised receiver window. Each time an ACK is received, the trailing edge of the send window is advanced. The minimum of the sender's buffer and the advertised receiver's window is used to calculate a new leading edge. If this send window encompasses unsent data, this data can be sent immediately.

Figure 7: TCP Sliding Window



The size of TCP buffers in each host is a critical limitation to performance in WANs. The protocol is capable of transferring one send window of data per round-trip interval. For example, with a send window of 4096 bytes and a transmission path with an RTT of 600 ms, a TCP session is capable of sustaining a maximum transfer rate of 48 Kbps, regardless of the bandwidth of the network path. Maximum efficiency of the transfer is obtained only if the sender is capable of completely filling the network path with data. Because the sender will have an amount of data in forward transit and an equivalent amount of data awaiting reception of an ACK signal, both the sender's buffer and the receiver's advertised window should be no smaller than the *Delay-Bandwidth Product* of the network path. That is:

$$\textit{Window size} \geq \textit{Bandwidth (bytes/sec)} \times \textit{Round-trip time (sec)}$$

The 16-bit field within the TCP header can contain values up to 65,535, imposing an upper limit on the available window size of 65,535 bytes. This imposes an upper limit on TCP performance of some 64 KB per RTT, even when both end systems have arbitrarily large send and receive buffers. This limit can be modified by the use of a window-scale option, described in RFC 1323, effectively increasing the size of the window to a 30-bit field, but transmitting only the most significant 16 bits of the value. This allows the sender and receiver to use buffer sizes that can operate efficiently at speeds that encompass most of the current very-high-speed network transmission technologies across distances of the scale of the terrestrial intercontinental cable systems.

Although the maximum window size and the RTT together determine the maximum achievable data-transfer rate, there is an additional element of flow control required for TCP. If a TCP session commenced by injecting a full window of data into the network, then there is a strong probability that much of the initial burst of data would be lost because of transient congestion, particularly if a large window is being used. Instead, TCP adopts a more conservative approach by starting with a modest amount of data that has a high probability of successful transmission, and then probing the network with increasing amounts of data for as long as the network does not show signs of congestion. When congestion is experienced, the sending rate is dropped and the probing for additional capacity is resumed.

The dynamic operation of the window is a critical component of TCP performance for volume transfer. The mechanics of the protocol involve an additional overriding modifier of the sender's window, the *congestion window,* referred to as *cwnd*. The objective of the window-management algorithm is to start transmitting at a rate that has a very low probability of packet loss, then to increase the rate (by increasing the *cwnd* size) until the sender receives an indication, through the detection of packet loss, that the rate has exceeded the available capacity of the network. The sender then immediately halves its sending rate by reducing the value of *cwnd,* and resumes a gradual increase of the sending rate. The goal is to continually modify the sending rate such that it oscillates around the true value of available network capacity. This oscillation enables a dynamic adjustment that automatically senses any increase or decrease in available capacity through the lifetime of the data flow.

The intended outcome is that of a dynamically adjusting cooperative data flow, where a combination of such flows behaves fairly, in that each flow obtains essentially a fair share of the network, and so that close to maximal use of available network resources is made. This flow-control functionality is achieved through a combination of *cwnd* value management and packet-loss and retransmission algorithms. TCP flow control has three major parts: the flow-control modes of *Slow Start* and *Congestion Avoidance,* and the response to packet loss that determines how TCP switches between these two modes of operation.

TCP Slow Start

The starting value of the *cwnd* window (the *Initial Window,* or IW) is set to that of the *Sender Maximum Segment Size* (SMSS) value. This SMSS value is based on the receiver's maximum segment size, obtained during the SYN handshake, the discovered path MTU (if used), the MTU of the sending interface, or, in the absence of other information, 536 bytes. The sender then enters a flow-control mode termed *Slow Start*.

The sender sends a single data segment, and because the window is now full, it then awaits the corresponding ACK. When the ACK is received, the sender increases its window by increasing the value of *cwnd* by the value of SMSS. This then allows the sender to transmit two segments; at that point, the congestion window is again full, and the sender must await the corresponding ACKs for these segments. This algorithm continues by increasing the value of *cwnd* (and, correspondingly, opening the size of the congestion window) by one SMSS for every ACK received that acknowledges new data.

If the receiver is sending an ACK for every packet, the effect of this algorithm is that the data rate of the sender doubles every round-trip time interval. If the receiver supports delayed ACKs, the rate of increase will be slightly lower, but nevertheless the rate will increase by a minimum of one SMSS each round-trip time. Obviously, this cannot be sustained indefinitely. Either the value of *cwnd* will exceed the advertised receive window or the sender's window, or the capacity of the network will be exceeded, in which case packets will be lost.

There is another limit to the slow-start rate increase, maintained in a variable termed *ssthresh*, or *Slow-Start Threshold*. If the value of *cwnd* increases past the value of *ssthresh,* the TCP flow-control mode is changed from Slow Start to congestion avoidance. Initially the value of *ssthresh* is set to the receiver's maximum window size. However, when congestion is noted, *ssthresh* is set to half the current window size, providing TCP with a memory of the point where the onset of network congestion may be anticipated in future.

One aspect to highlight concerns the interaction of the slow-start algorithm with high-capacity long-delay networks, the so-called *Long Fat Networks* (or LFNs, pronounced "elephants"). The behavior of the slow-start algorithm is to send a single packet, await an ACK, then send two packets, and await the corresponding ACKs, and so on. The TCP activity on LFNs tends to cluster at each epoch of the round-trip time, with a quiet period that follows after the available window of data has been transmitted. The received ACKs arrive back at the sender with an inter-ACK spacing that is equivalent to the data rate of the bottleneck point on the network path. During Slow Start, the sender transmits at a rate equal to twice this bottleneck rate. The rate adaptation function that must occur within the network takes place in the router at the entrance to the bottleneck point. The sender's packets arrive at this router at twice the rate of egress from the router, and the router stores the overflow within its internal buffer. When this buffer overflows, packets will be dropped, and the slow-start phase is over. The important conclusion is that the sender will stop increasing its data rate when there is buffer exhaustion, a condition that may not be the same as reaching the true available data rate. If the router has a buffer capacity considerably less than the delay-bandwidth product of the egress circuit, the two values are certainly not the same.

In this case, the TCP slow-start algorithm will finish with a sending rate that is well below the actual available capacity. The efficient operation of TCP, particularly in LFNs, is critically reliant on adequately large buffers within the network routers.

Another aspect of Slow Start is the choice of a single segment as the initial sending window. Experimentation indicates that an initial value of up to four segments can allow for a more efficient session startup, particularly for those short-duration TCP sessions so prevalent with Web fetches[6]. Observation of Web traffic indicates an average Web data transfer of 17 segments. A slow start from one segment will take five RTT intervals to transfer this data, while using an initial value of four will reduce the transfer time to three RTT intervals. However, four segments may be too many when using low-speed links with limited buffers, so a more robust approach is to use an initial value of no more than two segments to commence Slow Start[7].

Packet Loss

Slow Start attempts to start a TCP session at a rate the network can support and then continually increase the rate. How does TCP know when to stop this increase? This slow-start rate increase stops when the congestion window exceeds the receiver's advertised window, when the rate exceeds the remembered value of the onset of congestion as recorded in *ssthresh,* or when the rate is greater than the network can sustain. Addressing the last condition, how does a TCP sender know that it is sending at a rate greater than the network can sustain? The answer is that this is shown by data packets being dropped by the network. In this case, TCP has to undertake many functions:

- The packet loss has to be detected by the sender.
- The missing data has to be retransmitted.
- The sending data rate should be adjusted to reduce the probability of further packet loss.

TCP can detect packet loss in two ways. First, if a single packet is lost within a sequence of packets, the successful delivery packets following the lost packet will cause the receiver to generate a *duplicate* ACK for each successive packet The reception of these duplicate ACKs is a signal of such packet loss. Second, if a packet is lost at the end of a sequence of sent packets, there are no following packets to generate duplicate ACKs. In this case, there are no corresponding ACKs for this packet, and the sender's retransmit timer will expire and the sender will assume packet loss.

A single duplicate ACK is not a reliable signal of packet loss. When a TCP receiver gets a data packet with an out-of-order TCP sequence value, the receiver must generate an immediate ACK of the highest in-order data byte received. This will be a duplicate of an earlier transmitted ACK. Where a single packet is lost from a sequence of packets, all subsequent packets will generate a duplicate ACK packet.

On the other hand, where a packet is rerouted with an additional incremental delay, the reordering of the packet stream at the receiver's end will generate a small number of duplicate ACKs, followed by an ACK of the entire data sequence, after the errant packet is received. The sender distinguishes between these cases by using three duplicate ACK packets as a signal of packet loss.

The third duplicate ACK triggers the sender to immediately send the segment referenced by the duplicate ACK value (*fast retransmit*) and commence a sequence termed *Fast Recovery*. In fast recovery, the value of *ssthresh* is set to half the current send window size (the send window is the amount of unacknowledged data outstanding). The congestion window, *cwnd,* is set three segments greater than *ssthresh* to allow for three segments already buffered at the receiver. If this allows additional data to be sent, then this is done. Each additional duplicate ACK inflates *cwnd* by a further segment size, allowing more data to be sent. When an ACK arrives that encompasses new data, the value of *cwnd* is set back to *ssthresh,* and TCP enters congestion-avoidance mode. Fast Recovery is intended to rapidly repair single packet loss, allowing the sender to continue to maintain the ACK-clocked data rate for new data while the packet loss repair is being undertaken. This is because there is still a sequence of ACKs arriving at the sender, so that the network is continuing to pass timing signals to the sender indicating the rate at which packets are arriving at the receiver. Only when the repair has been completed does the sender drop its window to the *ssthresh* value as part of the transition to congestion-avoidance mode[8].

The other signal of packet loss is a complete cessation of any ACK packets arriving to the sender. The sender cannot wait indefinitely for a delayed ACK, but must make the assumption at some point in time that the next unacknowledged data segment must be retransmitted. This is managed by the sender maintaining a *Retransmission Timer*. The maintenance of this timer has performance and efficiency implications. If the timer triggers too early, the sender will push duplicate data into the network unnecessarily. If the timer triggers too slowly, the sender will remain idle for too long, unnecessarily slowing down the flow of data. The TCP sender uses a timer to measure the elapsed time between sending a data segment and receiving the corresponding acknowledgment. Individual measurements of this time interval will exhibit significant variance, and implementations of TCP use a smoothing function when updating the retransmission timer of the flow with each measurement. The commonly used algorithm was originally described by Van Jacobson[9], modified so that the retransmission timer is set to the smoothed round-trip-time value, plus four times a smoothed mean deviation factor[10].

When the retransmission timer expires, the actions are similar to that of duplicate ACK packets, in that the sender must reduce its sending rate in response to congestion. The threshold value, *ssthresh,* is set to half of the current value of outstanding unacknowledged data, as in the duplicate ACK case. However, the sender cannot make any valid assumptions about the current state of the network, given that no useful information has been provided to the sender for more than one RTT interval. In this case, the sender closes the congestion window back to one segment, and restarts the flow in slow start-mode by sending a single segment. The difference from the initial slow start is that, in this case, the *ssthresh* value is set so that the sender will probe the congestion area more slowly using a linear sending rate increase when the congestion window reaches the remembered *ssthresh* value.

Congestion Avoidance

Compared to Slow Start, congestion avoidance is a more tentative probing of the network to discover the point of threshold of packet loss. Where Slow Start uses an exponential increase in the sending rate to find a first-level approximation of the loss threshold, congestion avoidance uses a linear growth function.

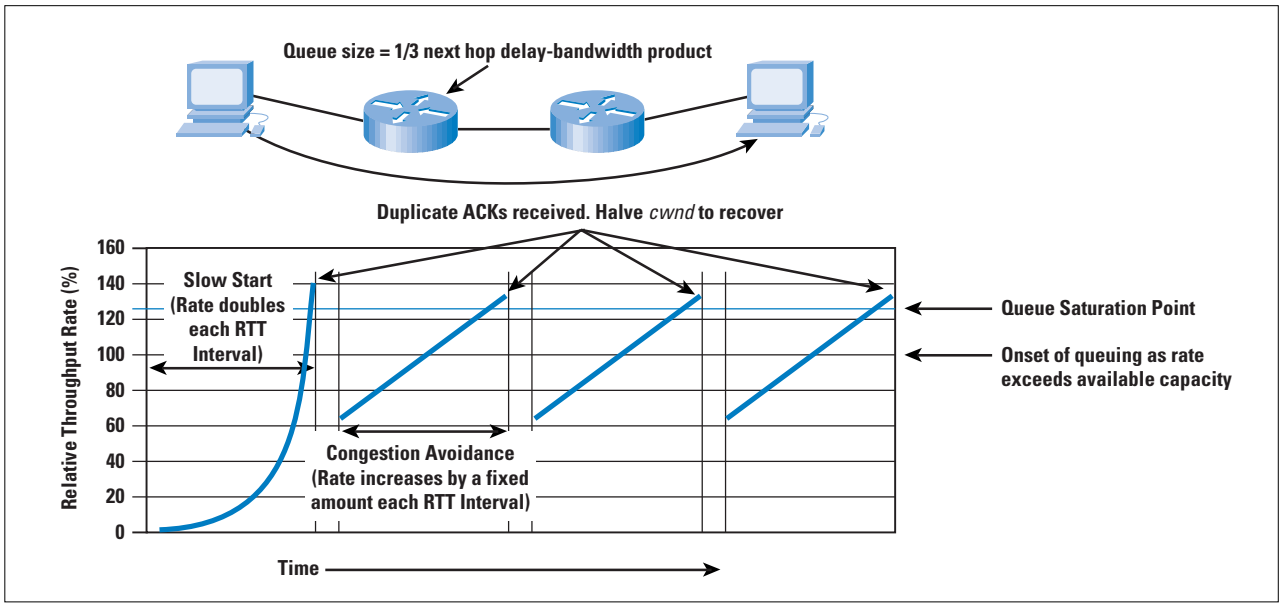
When the value of *cwnd* is greater than *ssthresh*, the sender increments the value of *cwnd* by the value *SMSS* × *SMSS/cwnd,* in response to each received nonduplicate ACK[7], ensuring that the congestion window opens by one segment within each RTT time interval.

The congestion window continues to open in this fashion until packet loss occurs. If the packet loss is isolated to a single packet within a packet sequence, the resultant duplicate ACKs will trigger the sender to halve the sending rate and continue a linear growth of the congestion window from this new point, as described above in fast recovery.

The behavior of *cwnd* in an idealized configuration is shown in Figure 8, along with the corresponding data-flow rates. The overall characteristics of the TCP algorithm are an initial relatively fast scan of the network capacity to establish the approximate bounds of maximal efficiency, followed by a cyclic mode of adaptive behavior that reacts quickly to congestion, and then slowly increases the sending rate across the area of maximal transfer efficiency.

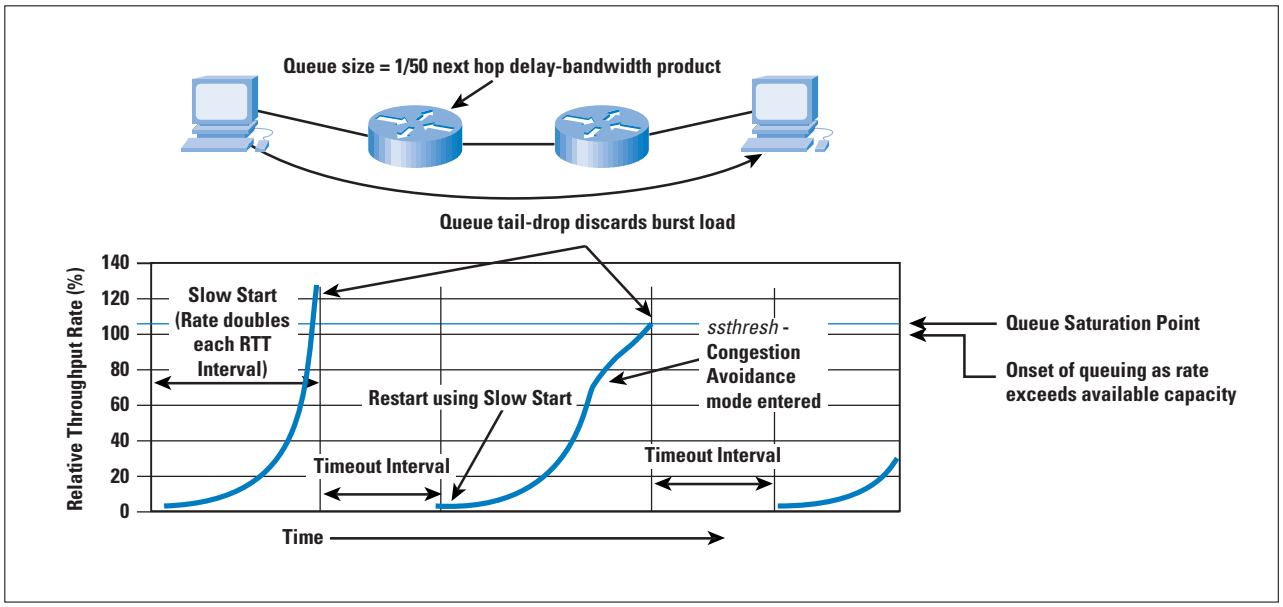
Packet loss, as signaled by the triggering of the retransmission timer, causes the sender to recommence slow-start mode, following a timeout interval. The corresponding data-flow rates are indicated in Figure 9.

Figure 8: Simulation of Single TCP Transfer



The inefficiency of this mode of performance is caused by the complete cessation of any form of flow signaling from the receiver to the sender. In the absence of any information, the sender can only assume that the network is heavily congested, and so must restart its probing of the network capacity with an initial congestion window of a single segment. This leads to the performance observation that any form of packet-drop management that tends to discard the trailing end of a sequence of data packets may cause significant TCP performance degradation, because such drop behavior forces the TCP session to continually time out and restart the flow from a single segment again.

Figure 9: Simulation of TCP Transfer with Tail Drop Queue

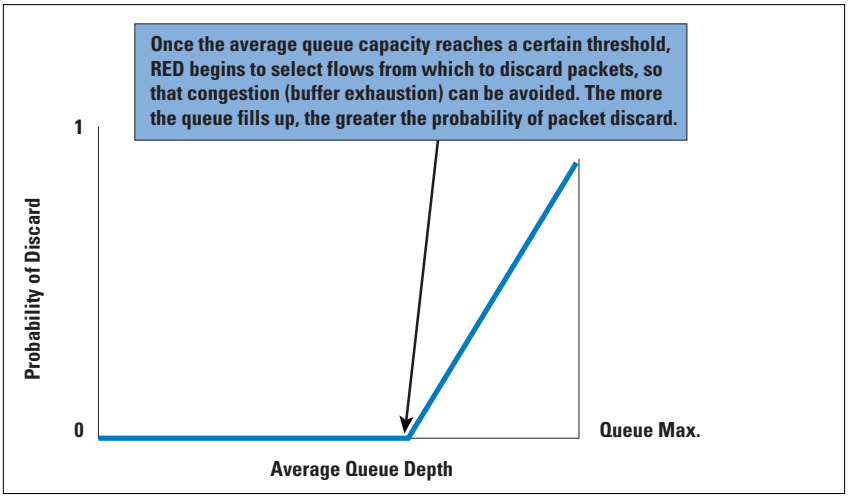


Assisting TCP Performance within the Network—RED and ECN

Although TCP is an end-to-end protocol, it is possible for the network to assist TCP in optimizing performance. One approach is to alter the queue behaviour of the network through the use of *Random Early Detection* (RED). RED permits a network router to discard a packet even when there is additional space in the queue. Although this may sound inefficient, the interaction between this early packet-drop behaviour and TCP is very effective.

RED uses a the weighted average queue length as the probability factor for packet drop. As the average queue length increases, the probability of a packet being dropped, rather than being queued, increases. As the queue length decreases, so does the packet-drop probability. (See Figure 10). Small packet bursts can pass through a RED filter relatively intact, while larger packet bursts will experience increasingly higher packet-discard rates. Sustained load will further increase the packet-discard rates. This implies that the TCP sessions with the largest open windows will have a higher probability of experiencing packet drop, causing a back-off in the window size.

Figure 10: RED Behavior



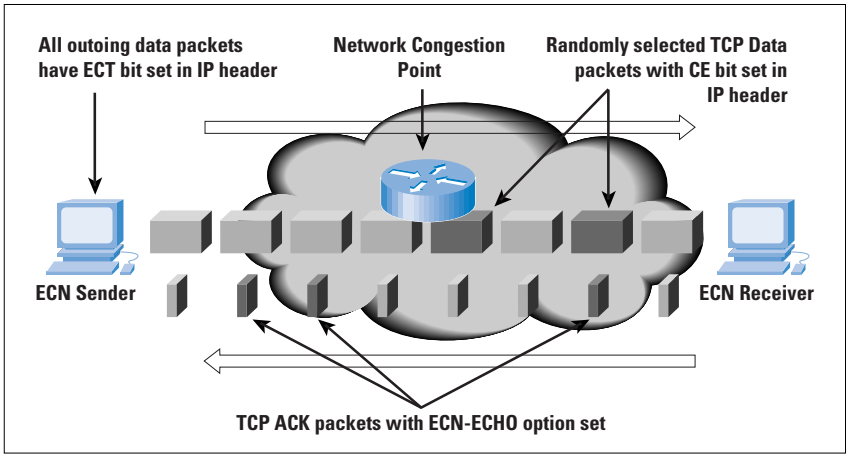
A major goal of RED is to avoid a situation in which all TCP flows experience congestion at the same time, all then back off and resume at the same rate, and tend to synchronize their behaviour[11,12]. With RED, the larger bursting flows experience a higher probability of packet drop, while flows with smaller burst rates can continue without undue impact. RED is also intended to reduce the incidence of complete loss of ACK signals, leading to timeout and session restart in slow-start mode. The intent is to signal the heaviest bursting TCP sessions the likelihood of pending queue saturation and tail drop before the onset of such a tail-drop congestion condition, allowing the TCP session to undertake a fast retransmit recovery under conditions of congestion avoidance. Another objective of RED is to allow the queue to operate efficiently, with the queue depth ranging across the entire queue size within a timescale of queue depth oscillation the same order as the average RTT of the traffic flows.

Behind RED is the observation that TCP sets very few assumptions about the networks over which it must operate, and that it cannot count on any consistent performance feedback signal being generated by the network. As a minimal approach, TCP uses packet loss as its performance signal, interpreting small-scale packet-loss events as peak load congestion events and extended packet loss events as a sign of more critical congestion load. RED attempts to increase the number of small-scale congestion signals, and in so doing avoid long-period sustained congestion conditions.

It is not necessary for RED to discard the randomly selected packet. The intent of RED is to signal the sender that there is the potential for queue exhaustion, and that the sender should adapt to this condition. An alternative mechanism is for the router experiencing the load to mark packets with an explicit *Congestion Experienced* (CE) bit flag, on the assumption that the sender will see and react to this flag setting in a manner comparable to its response to single packet drop[13] [14]. This mechanism, *Explicit Congestion Notification* (ECN), uses a 2-bit scheme, claiming bits 6 and 7 of the IP Version 4 *Type-of-Service* (ToS) field (or the two *Currently Unused* [CU] bits of the IP *Differentiated Services* field). Bit 6 is set by the sender to indicate that it is an ECN-capable transport system (the ECT bit). Bit 7 is the CE bit, and is set by a router when the average queue length exceeds configured threshold levels.

The ECN algorithm is that an active router will perform RED, as described. After a packet has been selected, the router may mark the CE bit of the packet if the ECT bit is set; otherwise, it will discard the selected packet. (See Figure 11).

Figure 11: Operation of Explicit Congestion Notification



The TCP interaction is slightly more involved. The initial TCP SYN handshake includes the addition of ECN-echo capability and *Congestion Window Reduced* (CWR) capability flags to allow each system to negotiate with its peer as to whether it will properly handle packets with the CE bit set during the data transfer. The sender sets the ECT bit in all packets sent. If the sender receives a TCP packet with the ECN-echo flag set in the TCP header, the sender will adjust its congestion window as if it had undergone fast recovery from a single lost packet.

The next sent packet will set the TCP CWR flag, to indicate to the receiver that it has reacted to the congestion. The additional caveat is that the sender will react in this way at most once every RTT interval. Further, TCP packets with the ECN-echo flag set will have no further effect on the sender within the same RTT interval. The receiver will set the ECN-echo flag in all packets when it receives a packet with the CE bit set. This will continue until it receives a packet with the CWR bit set, indicating that the sender has reacted to the congestion. The ECT flag is set only in packets that contain a data payload. TCP ACK packets that contain no data payload should be sent with the ECT bit clear.

The connection does not have to await the reception of three duplicate ACKs to detect the congestion condition. Instead, the receiver is notified of the incipient congestion condition through the explicit setting of a notification bit, which is in turn echoed back to the sender in the corresponding ACK. Simulations of ECN using a RED marking function indicate slightly superior throughput in comparison to configuring RED as a packet-discard function.

However, widespread deployment of ECN is not considered likely in the near future, at least in the context of Version 4 of IP. At this stage, there has been no explicit standardization of the field within the IPv4 header to carry this information, and the deployment base of IP is now so wide that any modifications to the semantics of fields in the IPv4 header would need to be very carefully considered to ensure that the changed field interpretation did not exercise some malformed behavior in older versions of the TCP stack or in older router software implementations.

ECN provides some level of performance improvement over a packet-drop RED scheme. With large bulk data transfers, the improvement is moderate, based on the difference between the packet retransmission and congestion-window adjustment of RED and the congestion-window adjustment of ECN. The most notable improvements indicated in ECN simulation experiments occur with short TCP transactions (commonly seen in Web transactions), where a RED packet drop of the initial data packet may cause a six-second retransmit delay. Comparatively, the ECN approach allows the transfer to proceed without this lengthy delay.

The major issue with ECN is the need to change the operation of both the routers and the TCP software stacks to accommodate the operation of ECN. While the ECN proposal is carefully constructed to allow an essentially uncoordinated introduction into the Internet without negative side effects, the effectiveness of ECN in improving overall network throughput will be apparent only after this approach has been widely adopted. As the Internet grows, its inertial mass generates a natural resistance to further technological change; therefore, it may be some years before ECN is widely adopted in both host software and Internet routing systems. RED, on the other hand, has had a more rapid introduction to the Internet, because it requires only a local modification to router behavior, and relies on existing TCP behavior to react to the packet drop.

Tuning TCP

How can the host optimize its TCP stack for optimum performance? Many recommendations can be considered. The following suggestions are a combination of those measures that have been well studied and are known to improve TCP performance, and those that appear to be highly productive areas of further research and investigation[1].

- *Use a good TCP protocol stack:* Many of the performance pathologies that exist in the network today are not necessarily the by-product of oversubscribed networks and consequent congestion. Many of these performance pathologies exist because of poor implementations of TCP flow-control algorithms; inadequate buffers within the receiver; poor (or no) use of path-MTU discovery; no support for fast-retransmit flow recovery, no use of window scaling and SACK, imprecise use of protocol-required timers, and very coarse-grained timers. It is unclear whether network ingress-imposed Quality-of-Service (QoS) structures will adequately compensate for such implementation deficiencies. The conclusion is that attempting to address the symptoms is not the same as curing the disease. A good protocol stack can produce even better results in the right environment.
- *Implement a TCP Selective Acknowledgment (SACK) mechanism:* SACK, combined with a selective repeat-transmission policy, can help overcome the limitation that traditional TCP experiences when a sender can learn only about a single lost packet per RTT.
- *Implement larger buffers with TCP window-scaling options:* The TCP flow algorithm attempts to work at a data rate that is the minimum of the delay-bandwidth product of the end-to-end network path and the available buffer space of the sender. Larger buffers at the sender and the receiver assist the sender in adapting more efficiently to a wider diversity of network paths by permitting a larger volume of traffic to be placed in flight across the end-to-end path.
- *Support TCP ECN negotiation:* ECN enables the host to be explicitly informed of conditions relating to the onset of congestion without having to infer such a condition from the reserve stream of ACK packets from the receiver. The host can react to such a condition promptly and effectively with a data flow-control response without having to invoke packet retransmission.
- *Use a higher initial TCP slow-start rate than the current 1 MSS (Maximum Segment Size) per RTT.* A size that seems feasible is an initial burst of 2 MSS segments. The assumption is that there will be adequate queuing capability to manage this initial packet burst; the provision to back off the send window to 1 MSS segment should remain intact to allow stable operation if the initial choice was too large for the path. A robust initial choice is two segments, although simulations have indicated that four initial segments is also highly effective in many situations.

- *Use a host platform that has sufficient processor and memory capacity to drive the network.* The highest-quality service network and optimally provisioned access circuits cannot compensate for a host system that does not have sufficient capacity to drive the service load. This is a condition that can be observed in large or very popular public Web servers, where the peak application load on the server drives the platform into a state of memory and processor exhaustion, even though the network itself has adequate resources to manage the traffic load.

All these actions have one thing in common: They can be deployed incrementally at the edge of the network and can be deployed individually. This allows end systems to obtain superior performance even in the absence of the network provider tuning the network's service response with various internal QoS mechanisms.

Conclusion

TCP is not a predictive protocol. It is an adaptive protocol that attempts to operate the network at the point of greatest efficiency. Tuning TCP is not a case of making TCP pass more packets into the network. Tuning TCP involves recognizing how TCP senses current network load conditions, working through the inevitable compromise between making TCP highly sensitive to transient network conditions, and making TCP resilient to what can be regarded as noise signals.

If the performance of end-to-end TCP is the perceived problem, the most effective answer is not necessarily to add QoS service differentiation into the network. Often, the greatest performance improvement can be made by upgrading the way that hosts and the network interact through the appropriate configuration of the host TCP stacks.

In the next article on this topic, we will examine how TCP is facing new challenges with increasing use of wireless, short-lived connections, and bandwidth-limited mobile devices, as well as the continuing effort for improved TCP performance. We'll look at a number of proposals to change the standard actions of TCP to meet these various requirements and how they would interact with the existing TCP protocol.

References

[1] Huston, G., *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, John Wiley & Sons, January 2000.

[2] Postel, J., "Transmission Control Protocol," RFC 793, September 1981.

[3] Jacobson, V., Braden, R., and Borman, D., "TCP Extensions for High Performance," RFC 1323, May 1992.

[4] Mathis, M., Madavi, J., Floyd, S., and Romanow, A., "TCP Selective Acknowledgement Options," RFC 2018, October 1996.

[5] Nagle, J., "Congestion Control in IP/TCP Internetworks," RFC 896, January 1984.

[6] Allman, M., Floyd, S., and Partridge, C., "Increasing TCP's Initial Window," RFC 2414, September 1998.

[7] Allman, M., Paxson, V., and Stevens, W., "TCP Congestion Control," RFC 2581, April 1999.

[8] Stevens, W. R., *TCP/IP Illustrated, Volume 1,* Addison-Wesley, 1994.

[9] Jacobson V., "Congestion Avoidance and Control," ACM *Computer Communication Review,* Vol. 18, No. 4, August 1988.

[10] Jacobson, V., "Berkeley TCP Evolution from 4.3-Tahoe to 4.3, Reno," Proceedings of the 18th Internet Engineering Task Force, University of British Colombia, Vancouver, BC, September 1990.

[11] Floyd, S., and Jacobson, V., "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM *Transactions on Networking,* Vol. 1, No. 4, August 1993.

[12] Braden, R. et al., "Recommendations on Queue Management and Congestion Avoidance in the Internet," RFC 2309, April 1998.

[13] Floyd, S., "TCP and Explicit Congestion Notification," ACM *Computer Communication Review,* Vol. 24, No. 5, October 1994.

[14] Ramakrishnan, K., and Floyd, S., "A Proposal to Add Explicit Congestion Notification (ECN) to IP," RFC 2481, January 1999.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Technologist in the Internet area for Telstra. He is also an active member of the IETF, and is the chair of the Internet Society Board of Trustees. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@telstra.net**

Overview of Internet Mail Standards

by Paul Hoffman, Internet Mail Consortium

People who are new to the Internet often think it is equivalent to "the Web" since that's what they have heard about most in the media. After a few weeks of using their new Internet account, they tend to say the Internet is "e-mail and the Web," in that order.

Business users have an even higher regard for e-mail. According to the American Management Association, most business people say that e-mail has surpassed the telephone in importance for business communication. While many companies believe that their Web site will be very important in a few years, their e-mail system is already extremely critical to them today.

Because mail is one of the oldest services on the Internet, the protocols used to move mail around are more stable and mature than those used for newer services. The flip side of this is that some of the protocols that are used to move the billions of pieces of mail a day are somewhat arcane and even quaint. The *Internet Engineering Task Force* (IETF) motto "if it isn't broken, don't fix it!" has prevented people from redesigning Internet mail. Instead, numerous extensions and enhancements have been added to the original set of mail standards, as we shall see later.

Historically, there have been many other mail systems, such as BITNET, Fidonet, MAPI, cc:Mail, and so on. Of course, users of these systems still exist, and there is quite an active market for systems that act as gateways between Internet mail and other systems. However, this article only covers the tried-and-true Internet mail system.

The Internet Mail Model

Many Internet protocols are simple client/server systems with a single message payload format. Mostly due to history, Internet mail doesn't have this luxury. Figure 1 shows the main protocols and formats used to move Internet mail.

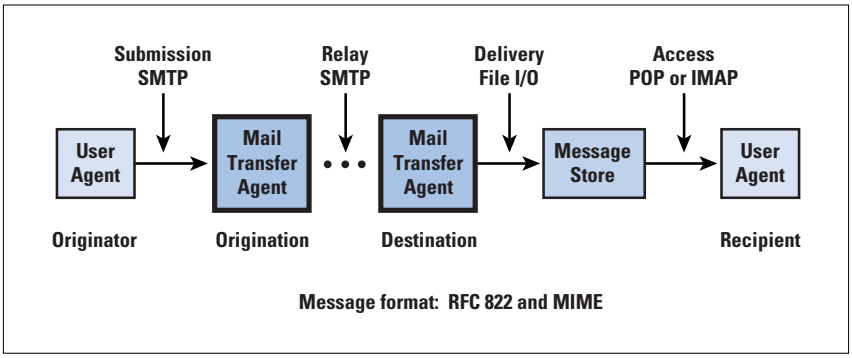


Figure 1: Internet Mail Architecture

A typical mail transaction goes from left to right in the figure. A *Mail User Agent* (MUA), which is most often run by a human but could be controlled by a program, submits a message using the *Simple Mail Transfer Protocol* (SMTP) to the initiating host. That host looks up the IP address associated with the destination host computer and sends the message to the destination host using SMTP. The destination host receives the message and writes it into the local message store (which almost always is a file or database on a hard drive).

The recipient MUA checks the mail store periodically and, if there is mail, retrieves it. Again, the recipient is often a human but might be an automated program such as an order entry system that is controlled by e-mail. The protocols that checks for and retrieves mail are usually the *Post Office Protocol* (POP) or *Internet Message Access Protocol* (IMAP), but it could also be any number of proprietary systems.

E-mail messages have a format that is quite easy to understand, so much so that many other protocols have adopted very similar formats. The message consists of ASCII text *headers* followed by one ASCII (or possibly binary) message *body*. The header format is defined in RFC 822[1], thus the headers are called "RFC 822 headers" or just "822 headers." A simple message body is a single string of text; a complex body uses the *Multipurpose Internet Mail Extensions* (MIME) message format.

Moving Between Hosts: SMTP

Early host-to-host mail delivery was done using file transfer protocols. Since such methods offer little flexibility and requires knowledge of user names and file structures of the remote system, a more general purpose delivery mechanism evolved. The resulting protocol, SMTP was defined in 1982 and has proven to work effectively in the face of orders of magnitude increase in the size of the network.

Even though SMTP moves mail between two host computers, it is a client/server protocol. The host that initiates the contact always acts as a *client,* and the host that was contacted is the *server*. (There are a few rarely-used exceptions to this rule.) The client has a variety of text-based commands that it can give, and the server replies with short responses. The server in the relationship never gives commands on its own, so it is up to the client to ask enough questions, and to carefully watch the server's responses, to know how best to interact with the server.

When a host wants to send mail somewhere on the Internet, it determines where the mail should go and initiates contact with the target server. Thus, the sender is always the SMTP client, and the hosts that are listening for SMTP traffic are always servers. In reality, most SMTP server software can act both as clients and servers; MUAs almost always only participate in SMTP as clients.

The sending host uses the *Domain Name System* (DNS) to determine the IP address of the target host, contacts that host using TCP port 25, and uses SMTP to deliver the message. Sometimes, as we shall see later, this IP lookup involves a level of indirection,—the target host may use a different host to receive mail on its behalf.

SMTP client commands consist of a keyword, possibly followed by command arguments. The server's response always starts with a three-digit number that is a status indicator, which is possibly followed by additional information.

Most SMTP interactions follow a typical set of steps, shown in Figure 2. The initiator who has mail to send (the client) is on the left and the host that is receiving the mail (the server) is on the right. The client first opens a TCP connection on port 25 on the the server. Next, the client and server exchange greetings (the HELO command and response). The client then prepares the server to receive the message by telling the server who the message is from and who it is to; the server gives a positive acknowledgment to each of these commands. The client then asks if the server is ready for the body of the message and, when the server says yes, sends the message as a stream of lines that is followed by a single period on a line by itself. After the server says that it has received the message fully, the client says good-bye and closes the connection.

Figure 2: A Typical SMTP Exchange

Client Action	*Server Response*
<connects to the server using TCP>	
	`220 example.com WhizzyMail server version 2.32`
`HELO somecompany.com`	
	`250-example.com says howdy`
`MAIL FROM:<chrisj@somecompany.com>`	
	`250 OK`
	`354 Start mail input; end with <CRLF>.<CRLF>`

`Text of message...`
`...more text of message...`
`.`	

	`250 OK`
`QUIT`	
	`221 example.com Service finished`
<disconnects from the server>	

Submission and Relay

After a message is created, the creator uses SMTP to submit the message to one of two places: a local mail-forwarding host (such as the mail server at the sender's *Internet Service Provider* (ISP) or corporate IS services) or the mail server that the DNS says is definitive for the recipient. The former is typically used by Internet users who do not have persistent network connections; the latter is more common on systems with network connections that are always available.

Messages may be forwarded hop-by-hop from the sending host, via intermediary hosts, to the recipient. This is called "relaying." In many cases, a message will go through more than two relays, for instance when the recipient's network is configured to accept all incoming messages on one machine that later relays messages to individual departmental hosts. Note that submission and relay uses the same SMTP commands described above (a recent change to this scheme is described near the end of this article).

The last host in the chain makes the message available to the recipient. This is done by moving the message to the message store, which usually means "write the message out on disk." There are, of course, many ways to write something on disk; some hosts write out each message as a separate file, some concatenate the message at the end of a file, while others write the message into a database.

Mail Addresses and MX Records

The initiating host's first job is to determine where a message is supposed to go, that is, how to contact the recipient's host. SMTP is a hop-by-hop protocol, meaning that a sending host does not know the true destination host for a message: it only knows the designated recipient host. Of course, this might be the recipient's final host, or it might be a host that will pass the message along further.

The domain name in mail addresses do not necessarily correspond to hosts on the Internet. For example, there is no host whose domain name is **`imc.org`**. When determining where to send a message, the initiating host first looks in the DNS for a *Mail Exchange* (MX) record that matches the domain name in the recipient's mail address. If there is no MX record, the initiating host looks for a DNS A record that matches the domain name. If there is no MX record or A record, the message cannot be delivered.

Many people find MX records to be somewhat tricky. Part of the confusion comes from the fact that an SMTP host is supposed to look up MX records before they look for A records; there are very few protocols that don't rely on A records. Another confusing aspect is that MX records may have wildcards in them. For instance, if a message is being sent to **`someone@eng.example.com`**, there may be no MX record for **`eng.example.com`**, but there may be one for **`*.example.com`**. Wildcard MX records tell the sending host that any message for a domain name that matches the wildcard specification should be sent to the named host.

Modern Mail Extensions

All protocols must evolve, and SMTP has improved over the years. Early mail implementors realized that the initial set of SMTP commands would have to expand. Since the SMTP client gives all commands in an exchange, the client determines which SMTP commands a server will be able to handle. The *SMTP Service Extension*s (ESMTP), defined in RFC 1869[2], is a small change to SMTP that allows an SMTP server to list the commands it knows at the beginning of an SMTP session.

The bootstrapping process for ESMTP is quite simple. Instead of starting with the "HELO" command, an ESMTP server starts with the "EHLO" command. If the SMTP host indicates that it has no idea what "EHLO" means, the client knows that the server doesn't understand ESMTP, and therefore doesn't understand any SMTP extensions. On the other hand, if the server does understand the "EHLO" greeting, the host responds with the entire list of SMTP extensions that the client is allowed to use during the session.

There have been over a dozen extensions to SMTP that are on standards track in the IETF, and many more have been proposed. However, most modern SMTP servers have only implemented a few of these.

Probably the most publicized SMTP extension in the past few years has been the *SMTP Service Extension for Authentication* (AUTH) for authenticating the SMTP client to the server. The AUTH extension, described in RFC 2554[3], allows roaming users to submit mail from outside their local networks without forcing the servers to accept mail from just anyone. This new method, which is now starting to appear in both mail clients and servers, will reduce the hassle faced by many roaming users as they move from ISP to ISP.

Another significant SMTP extension that has become widely implemented is *Delivery Status Notifications*, or DSNs defined in RFC 1891[4]. These are similar to return receipts in postal mail, but with some significant differences. DSNs are issued by SMTP servers, not end users. Thus, the meaning of a DSN is interpreted as "the message was received by this SMTP host," not "the message was received by the intended recipient."

Retrieving Mail

After the final SMTP server has received a message and written it into the message store, the recipient needs to be able to access the message. In the early days of Internet mail, the message store was nothing more than a text file on disk, and mail was read by reading the text file. In fact, many people still read their mail this way, albeit using somewhat more modern tools.

If the recipient is not directly logged into the host computer that has the message store, reading the disk file can be difficult. To alleviate this problem, the *Post Office Protocol* (POP), described in RFC 1939[5] introduced a client/server model for an MUA to get mail from the message store and store it on the local computer. The vast majority of mail users today use POP to retrieve their mail.

POP looks like many Internet protocols. The client connects to the server, logs in using a user name and password, checks if it has any messages waiting for it, then asks for the messages one by one. The client has the option of leaving messages that it has read on the server or deleting them after they have been retrieved.

Modern Mail Access with IMAP

Although POP works well for many people, it has its drawbacks. The mail client cannot preview a message to see whether or not it wants to download it. The client has only one mailbox which has no hierarchical structure. In most POP systems, leaving all your mail on the server makes retrieving new mail quite slow. To get around these problems, the mail community developed the *Internet Message Access Protocol* (IMAP), described in RFC 2060[6].

IMAP is significantly more powerful than POP. IMAP clients give the user much more control over their mail, such as letting them keep some of their mail locally while leaving other mail on the server. IMAP even allows for mailboxes that are shared among users, such as group announcements lists. It also gives mail administrators many more opportunities to support novice users by keeping their mail in a central location. Most modern mail clients support IMAP, and IMAP servers are available from many vendors.

It should be noted that, although IMAP is considered much more useful than POP and is widely available, it has had very little adoption in the ISP market (it has been accepted much more readily in the enterprise mail market). The reasons for this are not clear. Many ISPs say they do not want to incur the costs and responsibilities of storing users' mail, even if this gives them greater ability to administer the mail. It is not clear what, if anything, will shift ISPs away from POP to IMAP.

Access Through Web Browsers

The ubiquity of the Web has introduced a new method for getting mail that has become surprisingly popular: the use of the *HyperText Transfer Protocol* (HTTP). Web access to e-mail lets users read their mail without a POP or IMAP client. Of course, this offers many fewer features than POP or IMAP; for instance, you can't easily store messages after reading them and getting file attachments in your mail takes many more steps. However, the big advantage of this method is that Web browsers are almost everywhere these days, and there are many situations where you don't care about being able to store your mail on your local computer.

Giving users Web browser access to their mail quickly became a commodity market. Now, almost every portal offers such services. In fact, many corporations and ISP also offer this service because it is a fairly easy add-on to POP and IMAP servers. As more and more users want to access their e-mail from small devices such as cellular phones, it is likely that these devices will include Web-like mail interfaces.

Client Extensions

Both POP and IMAP are extensible, and developers have proposed many extensions for both protocols, although most work is being done on IMAP. Because of the slow adoption of IMAP by ISPs (who could make its advantages much more visible), it's not clear when these will appear in clients and servers, even though many of them add interesting functionality that is wanted by both users and administrators.

There are many client extensions that don't rely on either POP or IMAP, however. One of the most popular is *Message Disposition Notifications* (MDNs), which are quite similar to postal return receipts. Unlike DSNs, which say that a particular message got to one of the servers in an SMTP chain, MDNs are truly end-to-end, and are returned by recipients when they open their mail.

Some people find MDNs intrusive ("why should he know when I read this?"), and they aren't particularly reliable because not all mail clients (most notably Web browser readers) support them. However, they are a good example of what end users are seeing in terms of extensions that add desired functions to the Internet mail system.

The Format of Mail Messages

SMTP, POP, and (to a great extent) IMAP ignore the contents of a message. SMTP uses its own control information to find the recipient of a message; POP and IMAP retrieve messages based on user account names, which may or may not correspond to the address in a message. In users minds, however, the contents of the messages they read are almost always much more important than the way that the message got to them.

Mail messages consist of two parts: the *headers* and the *body*. The headers come first, followed by a blank line, followed by the body, as shown in Figure 3. The basic structure of messages has remained unchanged since it was defined in RFC 822. Originally, the headers were designed to look like inter-office memos, and also to contain control and debugging information; today, some parts of the headers are considered to be as important as the body of the message.

Figure 3: A Typical E-mail Message

```
Received: from mail.somecompany.com ([198.81.17.2])
  by mail.example.com (8.8.8/8.8.5) with ESMTP id VAA17989
  for <althea@example.com>; Wed, 9 Dec 1998 21:07:44 -0800 (PST)
From: jerry@somecompany.com
Message-ID: <823227a3.366f53ea@somecompany.com>
Date: Wed, 9 Dec 1998 23:54:02 EST
To: Althea Cassidy <althea@example.com>
Mime-Version: 1.0
Subject: I'm outta here
Content-type: text/plain; charset=US-ASCII
Content-transfer-encoding: 7bit

Sorry to make this so brief, but I've got a train to catch.
I'll meet you at the jubilee.
      --J
```

Message Headers

Because they were designed to be functional, message headers have a very straight-forward design. Each header has a single token, followed by a colon, followed by the parameters and options of the header. Headers usually consist of a single line, but you can create multi-line headers by starting the continuation lines with blanks.

There are dozens of common headers, and dozens more that are rarely used. Almost all mail users are familiar with "To:", "From:", "Subject:", and "Date:", and they may have seen additional common headers such as "Cc:" and "Received:". Depending on the interface of the MUA, users typically see some of these headers after they have retrieved a message with POP or IMAP but before they have "opened" the message to see the message body.

Basic and Advanced Message Bodies

Originally, the body of mail messages consisted of plain ASCII text. This was sufficient for the inventors of e-mail, who spoke mostly English and had access to other information transfer mechanisms such as FTP to move binary data around. Of course, such restrictions would not last.

Probably the biggest advance in Internet mail in the past ten years is in the format of mail messages, not in their transport. In the early 1990s, Internet mail went from being text-only to allowing the transfer of non-text messages and parts of messages. MIME, described in RFCs 2045–2047[7, 8, 9], revolutionized the usefulness of Internet mail by allowing senders to include files with messages, to use styled text, to give their messages useful structure, and to provide the first interoperable support for international e-mail.

Unfortunately, the term "attachments" became associated with MIME even though it is much more powerful than just allowing files to be attached to a message. The majority of MIME-enabled messages today don't contain any attachments: instead, they use MIME's capability of labeling the type of a single message body part. MIME labeling can tell the receiving client the format of the message (for instance, an HTML message) and, if it is a text message, the type of characters in the message.

Another great feature of MIME is that it allows messages to have structure. For instance, Figure 4 shows a message with two representations of the same information: text and HTML. A mail client that cannot display HTML can skip that part of the message and just display the plain text. This allows message content to gradually migrate towards new technology. In the near future, it is likely that similar logic will be used for messages that contain XML, HTML, and plain text.

Figure 4: A Multipart MIME message

```
From: jerry@somecompany.com
Message-ID: <828d83ffzwd.47r7c2dxsa@somecompany.com>
Date: Wed, 10 Dec 1998 03:24:00 EST
To: Althea Cassidy <althea@example.com>
Subject: What you should know
MIME-Version: 1.0
Content-Type: multipart/alternative;
   boundary=ad8ekd2ddr9332dc3df332

--ad8ekd2ddr9332dc3df332
Content-Type: text/plain; charset=us-ascii

Important stuff.
Blah blah blah.

--ad8ekd2ddr9332dc3df332
Content-Type: text/html

<html><head><title>Important stuff</title></head><body>
<h1>Important stuff.</h1>
<p><b>Blah blah blah.</b>
</body></html>

--ad8ekd2ddr9332dc3df332--
```

Because of the capability to structure messages, MIME can be used for multimedia and unified messaging. A single mail message can contain one or more movies, sound files, text files in a variety of formats, binary files such as word processing documents, calendar events, fax images, and so on. The MIME structure tells the recipient software which parts of the message contain particular types of data, as well the relationship between the parts (such as "this part contains three different alternative sound formats").

With the explosion of the popularity of the Web, users have come to expect that the content they read will look like Web pages. Most users don't understand that a "Web page" that "contains" graphics in fact isn't a single entity but is really a page of HTML that has links to other pages that contain individual images. They expect to be able to receive mail messages that look just like the things they see on the Web. The MHTML protocol (described in RFC 2557[10]) describes how to structure MIME messages that contain both HTML parts and images so that they appear together in mail clients exactly like they appear in Web browsers.

MIME enables a plethora of other uses for e-mail. For example, secure e-mail using S/MIME and PGP uses MIME to structure the messages so that the cryptographic control information is separate from the message itself. For instance, in a digitally-signed message, the signature information (which is unreadable to the human recipient) is in a different part of the structure than the human-readable content. You can even have layers of encryption and signatures, all structured through MIME.

Internationalization of E-mail

You can use character sets other than ASCII in both the headers and body of Internet e-mail messages. Using different character sets in text bodies requires the use of the "charset" parameter in the "Content-type:" header, as described in RFC 2046[8]. You can also use character sets in message headers with the methods described in RFC 2047[9].

The Future of E-mail

E-mail is incredibly popular with Internet users, but it is far from finished. The next billion new e-mail users will most likely be much less technically savvy than today's Internet users, and they will come to the Internet with very different expectations. In order to give these users a more pleasant experience, the Internet mail industry will have to add many new features and make mail clients easier.

The number of ISPs is also increasing, although not as fast as the number of Internet users. Since e-mail is such an integral part of the service that an ISP offers, mail server software will also have to become easier to administer. Internet mail server vendors are working on such enhancements as a way of gaining a competitive advantage.

The most major change that users will see in the next few years are more highly enabled MUAs. These clients will be all-in-one messaging centers that will handle faxes, voice messages, paging, calendar and event management, and probably some sort of instant messaging. In this way, traditional mail will be only one part of what the user sees when they go to their messaging client.

The importance of Internet fax should not be underestimated. The recent standards for Internet fax, defined in RFCs 2301–2306[11, 12, 13, 14, 15, 16] specify how faxes go through Internet mail. Although there have been a raft of proprietary real-time fax proposals, fax vendors have rallied around faxes in e-mail as an easy way to transition from fax over phone lines. Comparing the high cost of sending international faxes to the near-zero cost of sending e-mail, many companies are quickly moving towards the new standards.

Other mail-enabled services are becoming standardized as well. For example, calendaring over Internet mail is nearing completion. This will allow users to coordinate schedules for meetings, even with people who are not online. E-mail fall-back for phone conversations that were not completed is also being researched.

The e-mail world five and ten years from now will not necessarily look completely different from the way it looks today. Certainly, there will be many more enriched text and multimedia messages being composed by end users. Mailing lists will grow and the mail on them will be more like Web pages than today's text messages. Many people predict that the face of e-mail will change radically if e-mail becomes the "universal inbox" for voicemail, faxes, and other types of communication. Many companies are discovering that regular newsletters sent through e-mail are more effective than expecting users to come to a web site regularly, and it is likely that there will be an increase in the number of publications that are delivered as e-mail.[18]

There is still plenty of room for additions to Internet mail that resemble today's non-Internet services. For instance, users are already clamoring for features such as true message tracking, which is currently available from many package delivery services. Better security is clearly desired, although there seems to be major impediments caused by the need for trusted certificates before we can see wide deployment of secure mail. More problematic features such as message rescinding also have been proposed.

Forces outside the Internet mail world will also change how Internet mail works. For instance, the rapid increase in wireless users will change the way that large messages are handled by message stores. As more users start reading their mail from more than one system, IMAP may become more popular. At the same time, users will expect to be able to move their configuration information with them from machine to machine, probably using protocols such as the *Application Configuration Access Protocol* (ACAP) defined in RFC 2244[17].

There are plenty of opportunities in the Internet mail market. The only significant dark cloud is the possibility that increasing unsolicited e-mail—so called "spam"—might scare away users. To date, the technical solutions for battling spam have been limited, and they probably won't scale well if the amount of spam increases by an order of magnitude. On the bright side, it appears that most legitimate marketers have been scared away from spam and are focusing on opt-in e-mail marketing. This could be a boon for ISPs who specialize in bringing interested e-mail users and potential advertisers together.[19]

In such an environment, mail with rich media and lots of convenience could become the place where many users want to spend much of their time. To get there, we need to build on today's well-established mail protocols and to be creative in the kinds of features we add to both the transport and display of e-mail. Fortunately, we don't need to do much with SMTP, IMAP, and MIME in order to bring these new capabilities to the burgeoning numbers of new users waiting to get on the Internet.

References

[1] Crocker, D., "Standard for the format of ARPA Internet text messages," RFC 822, August 1982.

[2] Klensin, J., Freed, N., Rose, M., Stefferud, E., Crocker, D., "SMTP Service Extensions," RFC 1865, November 1995.

[3] Myers, J., "SMTP Service Extension for Authentication," RFC 2554, March 1999.

[4] Moore, K., "SMTP Service Extension for Delivery Status Notifications," RFC 1891, January 1996.

[5] Myers, J. and Rose, M., "Post Office Protocol—Version 3," RFC 1939, May 1996.

[6] Crispin, M., "Internet Message Access Protocol—Version 4rev1," RFC 2060, December 1996.

[7] Freed, N. and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies," RFC 2045, November 1996.

[8] Freed, N. and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part Two: Media," RFC 2046, November 1996.

[9] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text," RFC 2047, November 1996.

[10] Palme, J., Hopmann, A., Shelness, N., "MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)," RFC 2557, March 1999.

[11] McIntyre, L., Zilles, S., Buckley, R., Venable, D., Parsons, G., Rafferty, J., "File Format for Internet Fax," RFC 2301, March 1998.

[12] Parsons, G., Rafferty, J., Zilles, S., "Tag Image File Format (TIFF)—image/tiff MIME Sub-type Registration," RFC 2302, March 1998.

[13] Allocchio, C., "Minimal PSTN address format in Internet Mail," RFC 2303, March 1998.

[14] Allocchio, C., "Minimal FAX address format in Internet Mail," RFC 2304, March 1998.

[15] Toyoda, K., Ohno, H., Murai, J., Wing, D., "A Simple Mode of Facsimile Using Internet Mail," RFC 2305, March 1998.

[16] Parsons, G., Rafferty, J., "Tag Image File Format (TIFF)–F Profile for Facsimile," RFC 2306, March 1998.

[17] Newman, C., Myers J. G., "ACAP—Application Configuration Access Protocol," RFC 2244, November 1997.

[18] *Poor Richard's E-mail Publishing,* by Chris Pirillo, ISBN 0966103254, Top Floor Publishing, 1999.

[19] *Internet Messaging: From the Desktop to the Enterprise,* by Marshall T. Rose and David Strom, ISBN 0-13-978610-4, Prentice Hall PTR, 1998.

[20] *Essential Email Standards: RFCs and Protocols Made Practical,* by Pete Loshin, ISBN 0-471-34597-0, Wiley, 1999.

[21] *Internet Email Protocols: A Developer's Guide,* by Kevin Johnson, ISBN 0-201-43288-9, Addison-Wesley, 1999.

PAUL HOFFMAN is the director of the Internet Mail Consortium (**http://www.imc.org/**), which is the trade association for Internet mail software vendors and service providers. He is the editor of many recent mail standards, as well as Internet-related books such as *Netscape For Dummies*. He has been active on the Internet for twenty years. E-mail: **phoffman@imc.org**

Book Review

Introduction to Data Communications and Networking

Introduction to Data Communications and Networking, Behrouz Farouzan, ISBN 0-256-23044-7, WCB/McGraw-Hill, 1998.

As personal computers have proliferated the landscape over the years, they have become the domain of an increasing number of nontechnical end users. Two things assisted in this transformation. The realization of their value as a productivity tool became apparent, as well as their ability to become more user friendly to the masses. Networks, and networking, have followed a similar path. The investment in creating a networked environment in the past may have been a burden—in both time and added complexity—to all but the largest corporations. However, as the world becomes more "wired," the presence of networks has become commonplace in nearly every work environment, not to mention the movement into private residences. The need to become familiar with concepts and terms as they relate to data communications and networks has become an important part of the technological landscape. *Introduction to Data Communications and Networking* assists the novice in grasping these concepts, as well as serving as a refresher to the more experienced audience.

Organization

The preface explains the ways this book can be useful. The textbook portion is helpful. Multiple choice as well as discussion questions are provided within each chapter, although all the answers are not. In addition, some of the questions asked do not always seem to be posed in the context of the chapter just covered. However, it does turn out to be a rather small inconvenience. The requisite appendices are included as well—such as ASCII and EBCDIC codes, and various representations of numbers. However, two areas that usually receive only fleeting recognition—Fourier analysis and Huffman coding—are covered. Not being an engineer, I'm not sure that I now understand these concepts, but at least now I know why.

Although the areas covered in this book are covered in many introductory network books, this one takes nothing for granted. A good portion of the more experienced readers will know that Layers 2–6 of the OSI model have headers, only Layer 2 will include a trailer. Details such as these are easily forgotten. Introducing concepts in meaningful, practical ways is another positive attribute of this book. One great example is how the author describes the difference between analog and digital. Hands of a traditional, or analog, clock do not jump from minute to minute or hour to hour. The notion of time advancing seems to be a smooth transition, much like an analog signal is a continuous wave form that changes smoothly over time. Digital (as in the case of a digital clock), on the other hand, indicates discrete units of time—usually whole hours and minutes—and can have only limited numbers of defined values. In Chapter 4, analog and digital signals are detailed and explained with clarity and excellent examples are given as well.

In fact, the only subject matter I had difficulty deciphering concerned material presented in Chapter 5. The concepts of polar, unipolar, and bipolar encoding seemed straightforward enough, but digital-to-analog and analog-to-analog encoding will definitely have to be revisited. Amplitude and phase shifting keys may or may not be revisited. In fact, it was at this point that I realized that the material was moving to a different, more difficult, level.

Although the preface states that the first eight chapters are essential for readers being introduced to networking concepts, I found that chapters 5–8 went into a level of depth that would be particularly daunting for an introductory discussion.

Summary

I don't remember exactly how I was introduced to this book—whether I read about it in a journal or it was recommended by a friend—but the book got favorable reviews wherever I inquired about it. It is a practical addition to your bookshelf, regardless of your level of comfort with networks and voice/data communications.

The book is relevant and practical for the professional who has been working in the field for a few years. It is also useful as a textbook for use in the classroom. However, I do not believe that all the information can be adequately covered in a semester, as the author suggests. I believe one of the reasons I enjoyed this book was because of the way it explained ideas and concepts that were never used in any class I had ever taken. I recall promises of receiving a good, comprehensive background in these areas, yet years later I continue to struggle with some of the same concepts I've encountered in classes before. I found myself continually searching for a source that would provide me the information in a comprehensive, understandable fashion. I believe I have finally found it.

—Steve Barsamian, Cisco Systems
sbarsam@cisco.com

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Fragments

New Top-Level Domains Are Coming

For several years, there have been proposals to introduce new *generic top-level domains* (gTLDs) into the Internet *Domain Name System* (DNS). Although the introduction of gTLDs raises several issues that are of concern to various members of the Internet community, significant progress has been made recently toward achieving a consensus solution. The *Internet Corporation for Assigned Names and Numbers* (ICANN) Board of Directors is expected to consider adopting a policy to introduce new gTLDs at its meeting in July 2000. The *Names Council* has recommended to the ICANN Board that: "...a limited number of new top-level domains be introduced initially and that the future introduction of additional top-level domains be done only after careful evaluation of the initial introduction."

ICANN Announces CPR Institute as New Dispute Resolution Provider

ICANN recently announced that the *CPR Institute for Dispute Resolution* has been designated an approved provider under their *Uniform Dispute Resolution Policy* (UDRP) for domain name disputes. CPR, an alliance of 500 general counsel of global corporations and partners of major law firms, is the fourth dispute resolution provider to be designated by ICANN to handle domain disputes, joining the *National Arbitration Forum,* the *Disputes.org/eResolution Consortium,* and the *World Intellectual Property Organization.* The UDRP establishes a streamlined, economical process administered by neutral arbitration companies to provide a quick and cheap alternative to litigation. The procedure applies to cases that meet all three of the following criteria: The domain name must be identical or confusingly similar to a name in which the complaining party has trademark rights (either through a registered trademark or a common-law trademark); The domain name holder must have no legitimate right or interest in the name; The domain name must have been registered and used in bad faith.

In its first few months of operation, the UDRP has proven to be a very popular means of quickly resolving trademark/domain name disputes. To date, 691 proceedings have been commenced under the policy involving 1022 domain names. Of those proceedings, 348 have already been resolved. For additional information on UDRP, see **`http://www.icann.org/udrp/udrp.htm`**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Strategy Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

September 2000 Volume 3, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

In our last issue, Geoff Huston described the basic design and operation of the *Transmission Control Protocol* (TCP). He outlined how numerous enhancements to TCP implementations have been developed over time to improve its performance, particularly in the face of congested networks. The Internet is a rapidly changing environment in which both the applications and the underlying transmission systems are undergoing an evolution, if not a revolution. Some of these changes, such as the introduction of wireless devices, affect the way TCP works, because the protocol makes many implicit assumptions about the network over which it operates. In this issue, Geoff looks at the future for TCP and describes techniques for adopting TCP to today's Internet.

Security continues to be a major concern for everyone involved in the design and operation of networks. Widely publicized "hacker attacks," "denial-of-service attacks," and outright online fraud has brought the topic into sharp focus in the last few years. Because security was not part of the original design of the Internet, numerous solutions at every level of the protocol stack have been proposed and implemented over the last three decades. Today's network manager is, therefore, faced with a *system* of security components that must be carefully configured and monitored in order to provide sufficient security without preventing users from getting their work done. In our second article, Chris Lonvick explores a model for evaluating and securing a network.

The online subscription system for this journal is now up and running at **www.cisco.com/ipj** In addition to offering a subscription form, the system allows you to select delivery options, update your mailing and e-mail address, and much more. Please visit our Web site and give it a try. If you encounter any difficulties, please send your comments to **ipj@cisco.com**.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

The Future for TCP

by Geoff Huston, Telstra

The previous article, "TCP Performance," examined the operation of the *Transmission Control Protocol* (TCP) protocol[1]. The article examined the role of TCP in providing a reliable end-to-end data transfer function, and described how TCP incorporates numerous control functions that are intended to make efficient use of the underlying IP network through a host-based congestion control function. Congestion control is an important component of TCP implementations, and today TCP congestion control plays an important role in the overall stability of the Internet.

Today's Internet spans a very broad base of uses, and ensuring that TCP provides a highly robust, efficient, and reliable service platform for such a diversity of use is a continuing task. The Web has introduced a component of short duration reliable transfers into the public Internet traffic profile. These short sessions are often referred to as "TCP mice" because of the short duration and large number of such TCP sessions. Complementing these short sessions is the increasing size of large transfers as *File Transfer Protocol* (FTP) data sets become larger in response to increasing capacity within the public Internet network[4]. In addition, there is an increasing diversity of media used within the Internet, both in terms of higher-speed systems and in the use of wireless systems for Internet access. In this article we will extend our examination of TCP by looking at how TCP is being used and adapted to match this changing environment.

A Review of TCP Performance

Within any packet-switched network, when demand exceeds available capacity, the packet switch will use a queue to hold the excess packets. When this queue fills, the packet switch must drop packets. Any reliable data protocol that operates across such a network must recognize this possibility and take corrective action. TCP is no exception to this constraint. TCP uses data sequence numbering to identify packets, and explicit acknowledgements (ACKs) to allow the sender and receiver to be aware of reliable packet transfer. This form of reliable protocol design is termed "end-to-end" control, because interior switches do not attempt to correct packet drops. Instead, this function is performed through the TCP protocol exchange between sender and receiver. TCP uses cumulative ACKs rather than per-packet ACKs, where an ACK referencing a particular point within the data stream implicitly acknowledges all data with a sequence value less than the ACKed sequence.

TCP also uses ACKs to clock the data flow. ACKs arriving back at the sender arrive at intervals approximately equal to the intervals at which the data packets arrived at the sender. If TCP uses these ACKs to trigger sending further data packets into the network, then the packets will be entered into the network at the same rate as they are arriving at their destination. This mode of operation is termed "ACK clocking."

TCP recovers from packet loss using two mechanisms. The most basic operation is the use of packet timeouts by the sender. If an ACK for a packet fails to arrive within the timeout value, the sender will retransmit the oldest unacknowledged packet. In such a case, TCP assumes that the loss was caused by a network congestion condition, and the sender will enter "Slow Start" mode. This condition causes significant delays within the data transfer, because the sender will be idle during the timeout interval and upon restarting will recommence with a single packet exchange, gradually recovering the data rate that was active prior to the packet loss. Many networks exhibit transient congestion conditions, where a data stream may experience loss of a single packet within a packet train. To address this, TCP introduced the mechanism of "fast recovery." This mechanism is triggered by a sequence of three duplicate ACKS received by the data sender. These duplicate ACKs are generated by the packets that trail the lost packet, where the sender ACKs each of these packets with the ACK sequence value of the lost packet. In this mode the sender immediately retransmits the lost packet and then halves its sending rate, continuing to send additional data as permitted by the current TCP sending window. In this mode of operation, "congestion-avoidance" TCP increases its sending window at a linear rate of one segment per *Round-Trip Time* (RTT). This mode of operation is referred to as *Additive Increase, Multiplicative Decrease* (AIMD), where the protocol reacts sharply to signs of network congestion, and gradually increases its sending rate in order to equilibrate with concurrent TCP sessions.

TCP Design Assumptions

It is difficult to design any transport protocol without making some number of assumptions about the environment in which the protocol is to be used, and TCP certainly has some inherent assumptions hidden within its design. The most important set of assumptions that lie behind the design of TCP are as follows:

- *A network of wires, not wireless:* As we continually learn, wireless is different. Wireless systems typically have higher *bit error rates* (BERs) than wire-based carriage systems. Mobile wireless systems also include factors of signal fade, base-station handover, and variable levels of load. TCP was designed with wire-based carriage in mind, and the design of the protocol makes numerous assumptions that are typical of such of an environment. TCP makes the assumption that packet loss is the result of network congestion, rather than bit-level corruption. TCP also assumes some level of stability in the RTT, because TCP uses a method of damping down the changes in the RTT estimate.
- *A best-path route-selection protocol:* TCP assumes that there is a single best metric path to any destination because TCP assumes that packet reordering occurs on a relatively minor scale, if at all. This implies that all packets in a connection must follow the same path within the network or, if there is any form of load balancing, the order of packets within each flow is preserved by some network-level mechanism.

- *A network with fixed bandwidth circuits, not varying bandwidth:* TCP assumes that available bandwidth is constant, and will not vary over short time intervals. TCP uses an end-to-end control loop to control the sending rate, and it takes many RTT intervals to adjust to varying network conditions. Rapidly changing bandwidth forces TCP to make very conservative assumptions about available network capacity.
- *A switched network with first-in, first-out (FIFO) buffers:* TCP also makes some assumptions about the architecture of the switching elements within the network. In particular, TCP assumes that the switching elements use simple FIFO queues to resolve contention within the switches. TCP makes some assumption about the size of the buffer as well as its queuing behavior, and TCP works most efficiently when the buffer associated with a network interface is of the same order of size as the delay bandwidth product of the associated link.
- *The duration of TCP sessions:* TCP also makes some assumptions about the nature of the application. In particular, it assumes that the TCP session will last for some number of round-trip times, so that the overhead of the initial protocol handshake is not detrimental to the efficiency of the application. TCP also takes numerous RTT intervals to establish the characteristics of the connection in terms of the true RTT interval of the connection as well as the available capacity. The introduction of short-duration sessions, such as found in transaction applications and short Web transfers, is a new factor that impacts the efficiency of TCP.
- *Large payloads and adequate bandwidth:* TCP assumes that the overhead of a minimum of 40 bytes of protocol per TCP packet (20 bytes of IP header and 20 bytes of TCP header) is an acceptable overhead when compared to the available bandwidth and the average payload size. When applied to low-bandwidth links, this is no longer the case, and the protocol overheads may make the resultant communications system too inefficient to be useful.
- *Interaction with other TCP sessions:* TCP assumes that other TCP sessions will also be active within the network, and that each TCP session should operate cooperatively to share available bandwidth in order to maximize network efficiency. TCP may not interact well with other forms of flow-control protocols, and this could result in unpredictable outcomes in terms of sharing of the network resource between the active flows as well as poor overall network efficiency.

If these assumptions are challenged, the associated cost is that of TCP efficiency. If the objective is to extend TCP to environments where these assumptions are no longer valid, while preserving the integrity of the TCP transfer and maintaining a high level of efficiency, then the TCP operation itself may have to be altered.

There are two basic ways of altering TCP operation: by altering the actions of the end host by making changes to the TCP protocol, or by altering the characteristics of the network, making them more "friendly" to TCP. We will look at the potential for both responses in examining various scenarios for adapting TCP to suit these changing environments.

Some caution should be noted about making changes to the TCP protocol. The major constraint is that any changes that are contemplated to TCP should be backward compatible with existing TCP behavior. This constraint requires a modified TCP protocol to attempt to negotiate the use of a specific protocol extension, and the knowledge that a basic common mode of protocol operation may be required if the negotiation fails. The second constraint is that TCP does assume that it is interacting with other TCP sessions within the network, and the outcome of fair sharing of the network between concurrent sessions depends on some commonality of the protocol used by these sessions. Major changes to the protocol behavior can lead to unpredictable outcomes in terms of sharing of the network resource between "unmodified" and "modified" TCP sessions, and unpredictable outcomes in terms of efficiency of the use of the network. For this reason there is some understandable reluctance to undertake modifications of TCP that radically alter TCP startup behavior or behavior in the face of network congestion.

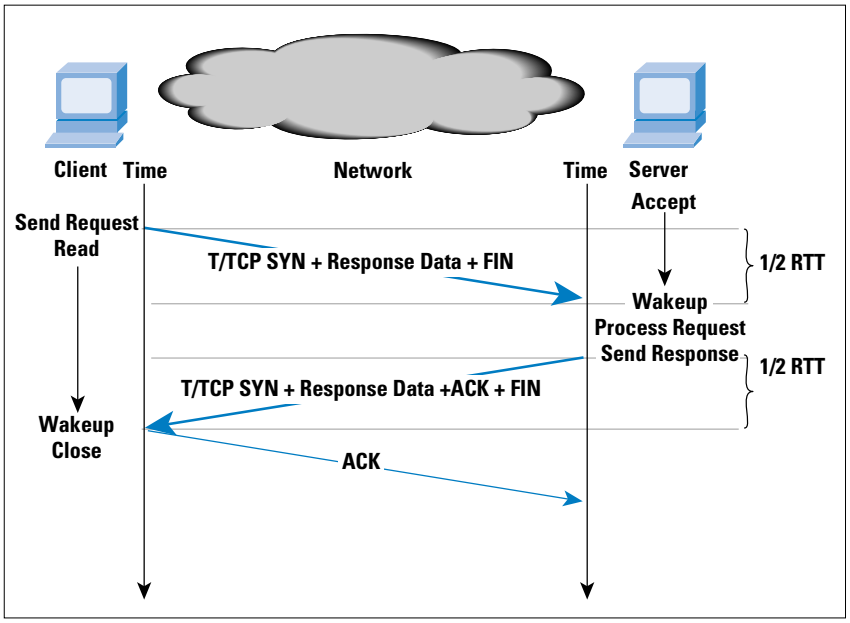
Short-Duration Sessions—TCP for Transactions

For network applications that generate small transactions, the application designer is faced with a dilemma. The application may be able to use the *User Datagram Protocol* (UDP), in which case the sender must send the query and await the response. This operation is highly efficient, because the total elapsed time for the client is a single RTT. However, this speed is gained at the cost of reliability. A missing response is ambiguous, in that it is impossible for the initiator to tell whether the query was lost or the response was lost. If multiple queries are generated, it is not necessarily true that they will arrive at the remote server in the same order as they were generated. Alternatively, the application can use TCP, which will ensure reliability of the transaction. However, TCP uses a three-way handshake to complete the opening of the connection, and uses acknowledged FIN signals for each side to close its end of the connection after it has completed sending data. Under the control of TCP, the sender will retransmit the query until it receives an acknowledgment that the query has arrived at the remote server. Similarly, the remote host will retransmit the response until the server receives an indication that the response has been successfully delivered. The cost of this reliability is application efficiency, because the minimum time to conduct the TCP transaction for the client is two RTT intervals.

TCP for Transactions (commonly referred to as T/TCP[5]) attempts to improve the performance of small transactions while preserving the reliability of TCP. T/TCP places the query data and the closing FIN in the initial SYN packet. This can interpreted as attempting to open a session, pass data, and close the sender's side of the session within a single packet. If the server accepts this format, the server responds with a single packet, which contains its SYN response, an ACK of the query data, the server's data in response, and the closing FIN. All that is required to complete the transaction is for the query system to ACK the server's data and FIN (Figure 1). If the server does not accept this format, the client can back off to a conventional TCP handshake followed by a data exchange.

For the client, the time to undertake this T/TCP transaction is one RTT interval, a period equal to the UDP-supported transaction, while still allowing for the two systems to use TCP to negotiate a reliable exchange of data as a backup.

Figure 1: T/TCP Operation



T/TCP requires changes to the protocol stack of both the sender and the receiver in order to operate correctly. The design of the protocol explicitly allows the session initiator to back off to use TCP if the receiver cannot correctly respond to the initial T/TCP packet.

T/TCP is not in common use in the Internet today, because while it improves the efficiency of simple transactions, the limited handshake makes it more vulnerable from a security perspective, and concerns over this vulnerability have been a prohibitive factor in its adoption. This is illustrative of the nature of the trade-offs that occur within protocol design, where optimizing one characteristic of a protocol may be at the expense of other aspects of the protocol.

Long Delay—TCP for Satellite Paths

Satellite-based services pose a set of unique issues to the network designer. Most notably, these issues include delay, bit errors, and bandwidth.

When using a satellite path, there is an inherent delay in the delivery of a packet due to signal propagation times related to the altitude of communications satellites. Geo-stationary orbit spacecraft are located at an altitude of some 36,000 km, and the propagation time for a signal to pass from an earth station directly below the satellite to the satellite and back is 239.6 ms. If the earth station is located at the edge of the satellite view area, this propagation time extends to 279.0 ms. In terms of a round trip that uses the satellite path in both directions, the RTT of a satellite hop is between 480 and 560 ms.

The strength of a radio signal falls in proportion to the square of the distance traveled. For a satellite link, the signal propagation distance is large, so the signal becomes weak before reaching its destination, resulting in a poor signal-to-noise ratio. Typical BERs for a satellite link today are on the order of 1 error per 10 million bits (1×10^{-7}). *Forward error correction* (FEC) coding can be added to satellite services to reduce this error rate, at the cost of some reduction in available bandwidth and an increase in latency due to the coding delay.

There is also a limited amount of bandwidth available to satellite systems. Typical carrier frequencies for commercial satellite services are 6/4 GHz (C-band) and 14/12 GHz (Ku band). Satellite transponder bandwidth is typically 36 MHz[6].

When used in a data carriage role for IP traffic, satellite channels pose several challenges for TCP.

The delay-bandwidth product of a transmission path defines the amount of data TCP should have within the transmission path at any one time, in order to fully utilize the available channel capacity. The delay used in this equation is the RTT and the bandwidth is the capacity of the bottleneck link in the network path. Because the delay in satellite environments is large, a TCP flow may need to keep a large amount of data within the transmission path. For example, a typical path that includes a satellite hop may have a RTT of some 700 ms. If the bottleneck bandwidth is 2 Mbps, then a sender will need to buffer 180 kB of data to fully utilize the available bandwidth with a single traffic flow. For this to be effective, the sender and receiver will need to agree on the use of TCP Window Scaling to extend the available window size beyond the protocol default limit of 64 kB. A sender using an 8 kB buffer would be able to achieve a maximum transfer rate of 91 kbps, irrespective of the available bandwidth on the satellite path.

Even with advanced FEC techniques, satellite channels exhibit a higher BER than typical terrestrial networks. TCP interprets packet drop as a signal of network congestion, and reduces its window size in an attempt to alleviate the situation. In the absence of certain knowledge about whether a packet was dropped because of congestion or corruption, TCP must assume the drop was caused by congestion in order to avoid congestion collapse[7, 8]. Therefore, packets dropped because of corruption cause TCP to reduce the size of its sending window, even though these packet drops do not signal congestion in the network. To mitigate this, some care must be taken with the satellite hop *Maximum Transmission Unit* (MTU) size, to reduce the probability of packet corruption. This is an area of compromise, in that the consequence is the potential for a high level of IP packet fragmentation on the satellite feeder router. In addition, the sender needs to use the TCP fast retransmit and fast recovery algorithms[9] in order to recover from the packet loss in a rapid, but stable fashion. In addition, the sender needs to use larger sending windows to operate the path more efficiently, with a consequent risk of multiple packet drops per RTT window. For this reason the use of *Selective Acknowledgements* (SACKs) is necessary in order to recover from multiple packet drops in a single RTT interval.

The long delay causes TCP to react slowly to the prevailing conditions within the network. The slow start of TCP commences with a single packet exchange, and it takes some number of RTT intervals for the sender's rate to reach the same order of size as the delay bandwidth product of the long delay path. For short-duration TCP transactions, such as much of the current Web traffic, this is a potential source of inefficiency. For example, if a transaction requires the transfer of ten packets, the slow-start algorithm will send a single packet in the first RTT interval, two in the second interval, four in the third, and the remaining three packets in the fourth RTT interval. Irrespective of the available bandwidth of the path, the transaction will take a minimum of four RTT intervals. This theoretical model is further exacerbated by delayed ACKs [RFC 1122], where a receiver will not immediately ACK a packet, but will await the expiration of the 500ms ACK timer, or a second full-sized packet. During slow start, where a sender sends an initial packet, and then awaits an ACK, the receiver will delay the ACK until the expiration of the delayed ACK timer, adding up to 500ms additional delay in the first data exchange. The second part of the delayed ACK algorithm is that it will only ACK every second full-sized data packet, slowing down the window inflation rate of slow start. Also, if congestion occurs on the forward data path, the TCP sender will not be aware of the condition until it receives duplicate ACKs from the receiver. A congestion condition may take many RTT intervals to clear, and in the case of a satellite path, transient congestions may take tens of seconds to be resolved.

The TCP mechanisms that assist in mitigating some of the more serious effects of satellite systems include *Path MTU Discovery*[10], *Fast Retransmit* and *Fast Recovery*, window scaling options, in order to extend the sender's buffer beyond 65,535 bytes[11], and the companion mechanisms of *Protection Against Wrapped Sequence Space* (PAWS) and *Round-Trip Time Measurements* (RTTM) and SACKs[12]. A summary of TCP options is shown in Figure 2.

Figure 2: TCP Options for Satellite Paths (after RFC 2488)

Mechanism	Use	Location
Path-MTU Discovery	Recommended	Sender
FEC	Recommended	Link
TCP		
Slow Start	Required	Sender
Congestion Avoidance	Required	Sender
Fast Retransmit	Recommended	Sender
Fast Recovery	Recommended	Sender
Window Scaling	Recommended	Sender and Receiver
PAWS	Recommended	Sender and Receiver
RTTM	Recommended	Sender and Receiver
SACK	Recommended	Sender and Receiver

Further refinements to the TCP stack have been considered in relation to satellite performance[13].

The options considered include the use of T/TCP as a means of reducing the overhead of the initial TCP three-way handshake. This is effective for short transactions where the data to be transferred can be held in a single packet, or in a small number of packets.

The use of delayed acknowledgements also is an issue for long-delay network paths, particularly if the sender is using slow start with an initial window of a single segment. In this case, the receiver will not immediately acknowledge the initial packet, but will wait up to one-half second for the delayed ACK timer to trigger. Altering the initial window size to two segments allows the receiver to trigger an ACK on reception of the second packet, bypassing the delayed ACK timer. However, even this change to TCP does not completely address the performance issue relating to delayed ACKs on long delay paths for TCP slow start. The delayed ACK algorithm triggers an ACK on every second full-sized packet. Because the sender's congestion window is opened on receipt of ACKs, this causes the slow-start window to open more slowly than if the receiver generated an ACK every packet. One variant of TCP congestion control allows the TCP sender to count the number of bytes acknowledged in an ACK message to control the expansion of the congestion window, making the algorithm less sensitive to delayed ACKs[9]. Although this approach has some merit for long delay paths, this is a case where the correction is potentially as bad as the original problem. The byte counting mode of congestion control allows a sender to sharply increase its sending rate, causing potential instabilities within the network and impacting concurrent TCP sessions.

One approach to address this is to place a limit on the size of the window expansion, where each increment of the congestion window is limited to the minimum of one or two segment sizes and the size of the data spanned by the ACK. If the limit is set to a single segment size, the window expansion will be in general slightly more conservative to the current TCP ACK-based expansion mechanism. If this upper limit is set to two segments, the congestion window expansion will account for the delayed ACKs, expand at a rate equal to one segment for every successfully transmitted segment during slow start, and expand the window by one segment size each RTT during congestion avoidance. Because a TCP receiver will ACK a large span of data following recovery, this byte counting is bounded to a single segment per ACK in the slow-start phase following a transmission timeout. Another approach that has been explored is for the receiver to disable delayed ACKs until the sender has completed the slow-start phase. Although such an approach shows promising results under simulated conditions, the practical difficulty is that it is difficult for the receiver to remotely determine the current TCP sending state, and the receiver cannot reliably tell if the sender is in slow start, congestion avoidance, or in some form of recovery mode. Explicit signaling of the sender's state as a TCP flag is an option, but the one-half RTT delay in the signaling from the sender to the receiver may prove to be an issue here. This area of congestion control for TCP remains a topic of study.

All of these approaches can mitigate only the worst of the effects of the long delay paths. TCP, as an adaptive reliable protocol that uses end-to-end flow control, can undertake only incremental adjustments in its flow rates in intervals of round-trip times. When the round-trip times extend, then TCP is slower to speed up from an initial start, slower to recover from packet loss, and slower to react to network congestion.

Tuning TCP—ACK Manipulation

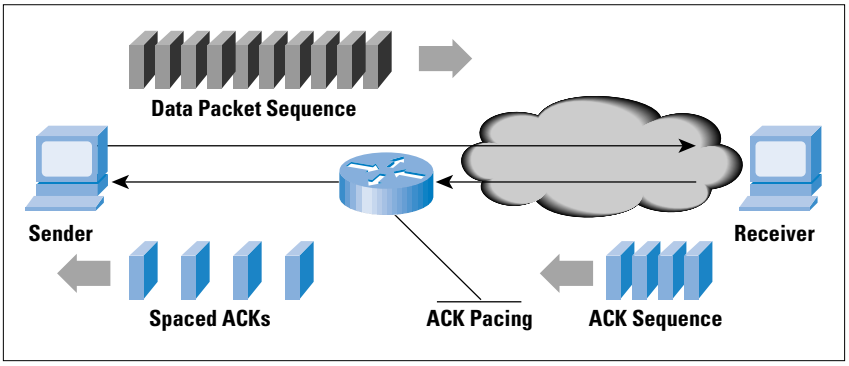
The previous article of TCP Performance discussed numerous network responses to congestion using *Random Early Detection* (RED) for active queue control and *Explicit Congestion Notification* (ECN) as an alternative to RED packet drop. It is feasible for a network control point to impose a finer level of control on a TCP flow by using an approach of direct manipulation of the TCP packets.

The approaches described above to mitigate some of the side effects of satellite paths all share in the side effect of having some latency associated with the congestion response. The sender must await the reception of trailing packets by the receiver, and then await the reception of the matching ACK packets from the data receiver back to the sender to learn of the fate of the original data packet. This may take up to one RTT interval to complete. An alternative approach to congestion management responses is to manipulate the ACK packets to modify the sender's behavior.

The prerequisite to perform this manipulation is that the traffic path be symmetric, so that the congestion point can identify ACK packets traveling in the opposite direction. If this is the case, a couple of control alternatives can mitigate the onset of congestion:

- *ACK Pacing:* Each burst of data packets will generate a corresponding burst of ACK packets. The spacing of these ACK packets determines the burst rate of the next sending packet sequence. For long-delay systems, the size of such bursts becomes a limiting factor. TCP slow start generates packet bursts at twice the bottleneck data rate, so that the bottleneck feeder router may have to absorb one-half of every packet burst within its internal queues. If these queues are not dimensioned to the delay bandwidth product of the next hop, these queues become the limiting factor, rather than the path bandwidth itself. If you can slow down the TCP burst rate, the pressure on the feeder queue is alleviated. One approach to slow down the burst rate is to impose a delay on successive ACKs at a network control point (Figure 3). This measure will reduce the burst rate, but not impact the overall TCP throughput. ACK pacing is most effective on long delay paths, and it is intended to spread out the burst load, reducing the pressure on the bottleneck queue and increasing the actual data throughput.

Figure 3: ACK Pacing



- *Window Manipulation:* Each ACK packet carries a receiver window size. This advertised window determines the maximum burst size available to the sender. Manipulating this window size downward allows a control point to control the maximal TCP sending rate. This manipulation can be done as part of a traffic-shaping control point, enforcing bandwidth limitations on a flow or set of flows.

Both of these mechanisms make some sweeping assumptions about the network control point that must be carefully understood. The major assumption is that these mechanisms assume symmetry of data flows at the network control point, where the data and the associated ACKs flow through this control point (but in opposite directions, of course). Both mechanisms also assume that the control point can cache per-flow state information, so that the current flow RTT and the current transfer rate and receiver window size are available to the service controller.

ACK pacing also implicitly assumes that a single ACK timing response is active at any time along a network path. A sequence of ACK delay actions may cause the sender's timers to trigger, and the sender to close down the transfer and reenter slow-start mode. These environmental conditions are more common at the edge of the network, and such mechanisms are often part of a traffic control system for Web-hosting platforms or similar network service delivery platforms. As a network control tool, ACK manipulation makes too many assumptions, and the per-flow congestion state information represents a significant overhead for large network systems. In general, such manipulations are more appropriate as an edge traffic filter, rather than as an effective congestion management response. For this reason, the more indirect approach of selective data packet discard is more effective as a congestion management measure.

Assisting Short-Duration TCP Sessions—Limited Transmit

One of the challenges to the original set of TCP assumptions is that of short-duration TCP sessions. The Web has introduced a large number of short-duration sessions, and the issue with these sessions is that they use small initial windows. If congestion loss occurs within this early period of TCP slow start, there are not enough packets in the network to generate the three duplicate ACKs required to initiate fast retransmit and fast recovery. Instead the TCP sender must await the expiry of the *retransmission timeout* (RTO), a timer that uses a minimum value of one second. For short-duration TCP sessions that may last six or seven RTT intervals of a small number of milliseconds, the incremental penalty of single packet loss is then extremely severe. A study of this problem indicates that approximately 56 percent of retransmissions are sent following an RTO timeout[25].

One potential mitigation to this is a mechanism termed "Limited Transmit." With this mechanism, a duplicate ACK may trigger an immediate transmission of a segment of new data. Two conditions are applied to this; the receiver's advertised window allows the transmission of this segment, and the amount of outstanding data would remain less than the congestion window plus the duplicate ACK threshold used to trigger Fast Retransmit. This second condition implies that the sender can send only two segments beyond the congestion window, and will do so only in response to the receiver lifting a segment off the network. The basic principle of this strategy is to continue the signaling between the sender and receiver in the face of packet loss, increasing the probability that the sender will recover from packet loss using duplicate ACKs and fast recovery, and reducing the probability of the one-second (or longer) RTO timeout as being the recovery trigger. The limited transmit also reduces the potential for the recovery actions to burst into the network at a level that may cause further packet loss.

Low Bandwidth and High Error Rates—TCP for Wireless Systems

One of the more challenging environments for a the Internet Protocol, and TCP in particular, is that of mobile wireless.

One approach to supporting the wireless environment is that of the so-called "walled garden." Here the protocols in use within the wireless environment are specifically adapted to the wireless world. The transport protocols can account for the low bandwidth, the longer latency, the BERs, and the variability within all three of these metrics. In this model, Internet applications interact with an application gateway to reach the wireless world, and the application gateway uses a wireless transport protocol and potentially a modified version of the application data to interact with the mobile wireless device. The most common approach is extension of the World Wide Web client into the mobile wireless device, using some form of proxy server at the boundary of the wireless network and the Internet. This is the approach adopted by the *Wireless Access Protocol Forum* (WAP)[14].

An alternative approach lies in extending not only the World Wide Web to a mobile handset, but also allowing mobile devices to access a complete range of Internet-based services as the functional objective. In this approach, the intent is to allow the mobile wireless device to function as any other Internet-connected device, and there is a consequent requirement for some form of end-to-end direct IP continuity, and an associated requirement for end-to-end TCP functionality, where the TCP path straddles both wired and wireless segments. Ensuring the efficient operation of TCP in this environment is an integral part of the development of such an environment. Given that TCP must now work within a broader environment, it is no longer a case of adjusting TCP to match the requirements of the wireless environment, but one of attempting to provide seamless interworking between the wired and wireless worlds (Figure 4).

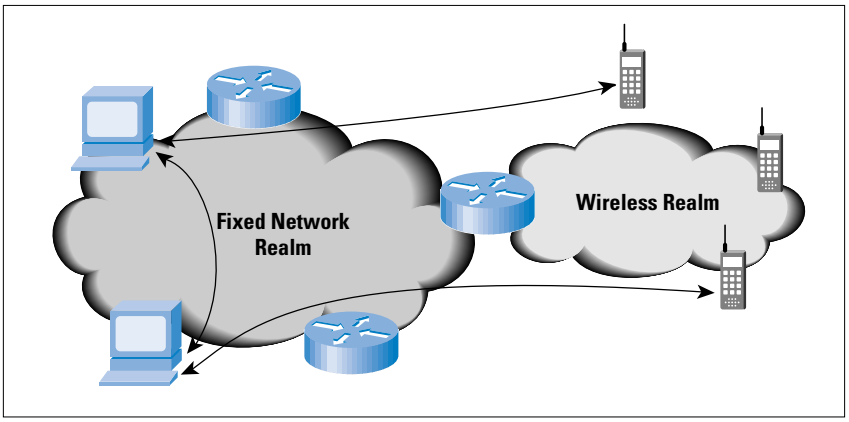


Figure 4: Linking the Wired and Wireless Worlds

The wireless environment challenges many of the basic assumptions of TCP noted above. Wireless has significant levels of bit error rates, often with bursting of very high error rates. Wireless links that use forward error correcting codes have higher latency. If the link level protocol includes automatic retransmission of corrupted data, this latency will have high variability. Wireless links may also use adaptive coding techniques that adjust to the prevailing signal to noise ratio of the link, in which case the link will have varying bandwidth. If the wireless device is a hand-held mobile device, it may also be memory constrained. And finally, such an environment is typically used to support short duration TCP sessions.

The major factor for mobile wireless is the BER, where frame loss of up to 1 percent is not uncommon, and errors occur in bursts, rather than as evenly spaced bit errors in the packet stream. In the case of TCP, such error conditions force the TCP sender to initially attempt fast retransmit of the missing segments, and when this does not correct the condition, the sender will have an ACK timeout occur, causing the sender to collapse its sending window and recommence from the point of packet loss in slow-start mode. The heart of this problem is that assumption on the part of TCP that packet loss is a symptom of network congestion rather than packet corruption. It is possible to use a model of TCP AIMD performance to determine the effects of this loss rate on TCP performance. If, for example the link has a 1-percent average packet loss rate, a *Maximum Segment Size* (MSS) size of 1000 bytes, and a 120ms RTT, then the AIMD models predict a best-case performance of 666Kbps throughput, and a more realistic target of 402Kbps throughput[15]. (See the appendix on page 24 for details of these models.) TCP is very sensitive to packet loss levels, and sustainable performance rapidly drops when packet drop levels exceed 1 percent.

Link-level solutions to the high BER are available to designers, and FEC codes and *automatic retransmission systems* (ARQ) can be used on the wireless link. FEC introduces a relatively constant coding delay and a bandwidth overhead into the path, but cannot correct all forms of bit error corruption. ARQ uses a "stop and resend" control mechanism similar to TCP itself. The consequent behavior is one of individual packets experiencing extended latency as the ARQ mechanisms retransmit link-level fragments to correct the data corruption, because the packet flow may halt for an entire link RTT interval for the link-level error to be signaled and the corrupted level 2 data to be retransmitted. The issue here is that TCP may integrate these extended latencies into its RTT estimate, making TCP assume a far higher latency on the path than is the case, or, more likely, it may trigger a retransmission at the same time as the level 2 ARQ is already retransmitting the same data. An alternative Layer 2 approach to bit-level corruption is to deliver those level 2 frames that were successfully transmitted, while resending any frames that were corrupted in transmission.

The problem for TCP here is that the level 2 drivers are adding packet reordering to the extended latency, and from TCP perspective the delivery of the out-of-order packets will generate duplicate ACKs that may trigger a simultaneous TCP fast retransmit.

Perversely, some approaches have advocated TCP delaying its duplicate ACK response in such situations[13]. To quote from RFC 2488, "The interaction between link-level retransmission and transport-level retransmission is not well understood."[6]

If ARQ is not the best possible answer to addressing packet loss in mobile wireless systems, then what can be done at the TCP level to address this? TCP can take numerous basic steps to alleviate the worst aspects of packet corruption on TCP performance. These include the use of Fast Retransmit and Fast Recovery to allow a single packet loss to be repaired moderately quickly. This mechanism triggers only after three duplicate ACKs, so the associated action is to ensure that the TCP sender and receiver can advertise buffers of greater than four times the MSS. SACKs allow a sender to repair multiple segment losses per window within a single RTT, and where large windows are operated over long delay paths, SACK is undoubtedly useful.

However, useful as these mechanisms may be, they are probably inadequate to allow TCP to function efficiently over all forms of wireless systems. Particularly in the case of mobile wireless systems, packet corruption is sufficiently common that, for TCP to work efficiently, some form of explicit addressing of network packet corruption appears to be necessary.

One approach is to decouple TCP congestion control mechanisms from data recovery actions. The intent is to allow new data to be sent during recovery to sustain TCP ACK clocking. This approach is termed *Forward Acknowledgements with Rate Halving* (FACK)[13], where one packet is sent for every two ACKs received while TCP is recovering from lost packets. This algorithm effectively reduces the sending rate by one-half within one RTT interval, but does not freeze the sender to wait the draining on one-half of the congestion window's amount of data from the network before proceeding to sending further data, nor does it permit the sender to burst retransmissions into the network. This is particularly effective for long-delay networks, where the fast recovery algorithm causes the sender to cease sending for up to one RTT interval, thereby losing the accuracy of the implicit ACK clock for the session. FACK allows the sender to continue to send packets into the network during this period, in an effort to allow the sender to maintain an accurate view of the ACK clock. FACK also provides an ability to set the number of SACK blocks that specify a missing segment before resending the segment, allowing the sender greater levels of control over sensitivity to packet reordering. The changes to TCP to support FACK are a change in the sender's TCP to use the FACK algorithm for recovery, and, for optimal performance, use of SACK options by the receiver.

In looking for alternative responses to packet corruption, it is noted that TCP segments that are corrupted are often detected at the link level, and are discarded by the link-level drivers. This discard cannot be used to generate an error message to the packet sender, given that the IP header of the packet may itself be corrupted, nor can the discard signal be reliably passed to the receiver, for the same reason. However, despite this unreliability of information, this signaling from the link level to the transport level is precisely the objective here, because, at the TCP protocol level, the sender needs to be aware that the packet loss was not due to network congestion, and that there is no need to take corrective action in terms of TCP congestion behavior.

One approach to provide this signaling from the data link level to the transport level calls for the link-level device to forward a "corruption experienced" *Internet Control Message Protocol* (ICMP) packet when discarding a corrupted packet[13]. This approach has the ICMP packet being sent in the forward direction to the receiver, who then has the task of converting this message and the associated lost packet information into a signal to the sender that the duplicate ACKs are the result of corruption, not network congestion. This signal from the receiver to the sender can be embedded in a TCP header option. The sending TCP session will maintain a corruption experienced state for two RTT intervals, retransmitting the lost packets without halving the congestion window size.

As we have noticed, corruption may have occurred in the packet header, and the sender's address may not be reliable. This approach addresses this by having the router keep a cache of recent packet destinations, and when the IP header information is unreliable because of a failed IP header checksum, the router will forward the ICMP message to all destinations in the cache. The potential weakness in this approach is that if network congestion occurs at the same time as packet corruption, the sender will not react to the congestion, and will continue to send into the congestion for a further two RTT intervals. This approach is not without some deployment concerns. It calls for modification to the wireless routers and to the receiver's link-level drivers to generate the ICMP corruption experienced messages, modification to the receiver's IP stack in order to take signals from the IP ICMP processor and from the link-level driver and convert them to TCP corruption loss signals within the TCP header of the duplicate ACKs, and modifications to the TCP processor at the sender to undertake corruption-experienced packet loss recovery. Even with these caveats in mind, this approach of explicit corruption signaling is a very promising approach to addressing performance issues with TCP over wireless.

Of course high levels of bit errors is not the only problem facing TCP over wireless systems. Mobile wireless systems are typically small handsets or personal digital assistants, and the application transactions are often modified to reduce the amount of data transferred, given that a limited amount of data can be displayed on the device.

In this case, the ratio between payload and IP and TCP headers starts to become an issue, and some consideration of header compression is necessary. Header compression techniques typically take the form of stripping out those fields of the header that do not vary on a packet-by-packet basis, or that vary by amounts that can be derived from other parts of the header, and then transmitting the delta values of those fields that are varying[16, 17].

Although such header compression schemes can be highly efficient in operation, the limitation of such schemes is that the receiver needs to have successfully received and decompressed the previous packet before the receiver can decompress the next packet in the TCP stream. In the face of high levels of bit error corruption, such systems do introduce additional latencies into the data transfer, and multiple packet drops are difficult to detect and signal via SACK in this case.

A more subtle aspect of mobile wireless is that of temporary link outages. For example, a mobile user may enter an area of no signal coverage for a period of time, and attempt to resume the data stream when signal is obtained again. In the same way that there is no accepted way of a link-level driver informing TCP of packet loss due to corruption, there is no way a link-level driver can inform TCP of a link-level outage. In the face of such link-level outages, TCP will assume network-level congestion, and in the absence of duplicate ACKs, TCP retransmission timers will trigger. TCP will then attempt to restart the session in slow-start mode, commencing with the first dropped packet. Each attempt to send the packet will result in TCP extending its retransmission timer using an exponential backoff on each attempt, so that successive probes are less and less frequent. Because the link level cannot inform the sender on the resumption of the link, TCP may wait some considerable time before responding to link restoration. The intention is for the link level to be able to inform the TCP for resumption of the connection following a link outage. One approach is for the link level to retain a packet from each TCP stream that attempted to use the link. When the link becomes operational again, the link-level driver immediately transmits these packets on the link. The result is that the receiver will then generate a response that will then trigger the sender into transmission within a RTT interval. Only a single packet per active TCP stream is necessary to trigger this response, so that the link level does not need to hold an extensive buffer of undeliverable packets during a link outage. Of course if the routing level repaired the link outage in the meantime, the delivery of an out-of-order TCP packet would normally be discarded by the sender.

The bottom line here is the question: Is TCP suitable for the mobile wireless environment? The answer appears to be that TCP can be made to work as efficiently as any other transport protocol for the mobile wireless environment.

However, this does imply that some changes in the operation of TCP need to be undertaken, specifically relating to the signaling of link-level states into the TCP session and use of advanced congestion control and corruption signaling within the TCP session. Although it is difficult to conceive of a change to every deployed TCP stack within the deployed Internet to achieve this added functionality, there does exist a middle ground between the "walled garden" approach and open IP. In this middle ground, the wireless systems would have access to "middleware," such as Web proxies and mail agents. These proxies would use a set of TCP options when communicating with mobile wireless clients that would make the application operate as efficiently as possible, while still permitting the mobile device transparent access to the Internet for other transactions.

Unbundling TCP—Stream Control Transmission Protocol

There are occasions where the application finds the control functions of TCP too limiting. In the case of handling *Public Switched Telephone Network* (PSTN) signaling across an Internet network, the application requirements are somewhat different from those of TCP delivered service. PSTN signaling reliable delivery is important, but the individual transactions within the application are included within each packet, so the concept of preservation of strict order of delivery is unnecessary. Relaxation of this requirement of strict order of packet delivery allows the transport protocol to function more efficiently, because there is no head-of-line blocking at the receiver when awaiting retransmission of lost packets. TCP also assumes the transfer of a stream of data, so that applications that wish to add some form of record delineation to the data stream have to add their own structure to the data stream. In addition, the limited scope of TCP sockets complicates the support of a high-availability application that may use multihomed hosts, and TCP itself is vulnerable to many attacks, such as SYN attacks. The intention of the *Stream Control Transmission Protocol* (SCTP) is to address these application requirements[16].

The first major difference between SCTP and TCP occurs during initialization, where the SCTP endpoints exchange a list of SCTP endpoint addresses (IP addresses and port numbers) that are to be associated with the SCTP session. Any pair of these source and destination addresses can be used within the SCTP session.

The startup of SCTP is also altered into a four-way handshake, where the initiator sends a tag value to the other end, which then responds with a copy of this tag and a tag of its own. At this stage the recipient does not allocate any resources for the connection, making the initialization sequence more robust in the face of TCP SYN-styled attacks. The initiator can then respond to this with an echo of the recipient's tag (COOKIE-ECHO), and can also attach data to the response, allowing data to be transferred as early as possible in the handshake process.

After the recipient ACKs this message, the SCTP session is now established. The closing of an SCTP session is also different from TCP. In TCP, one side can close its sending function via a FIN TCP packet, and continue to receive packets, operating in a "half-open" state. In SCTP, a close from one side will cause the other end to drain its send queues and also shut down.

SCTP also functions in a form of transport-level multiplexing, where numerous logical streams can be supported across a single transport-level association. Although message order within an individual stream is preserved by SCTP, retransmission within one stream does not impact the operation of any other stream that is supported across the same SCTP transport association. Each stream has an explicit identification and a per-stream sequence identification to support this function. SCTP also provides for nonsequenced message delivery, where a message within a stream is marked for immediate delivery, irrespective of the relative order of the message within a stream (Figure 5).

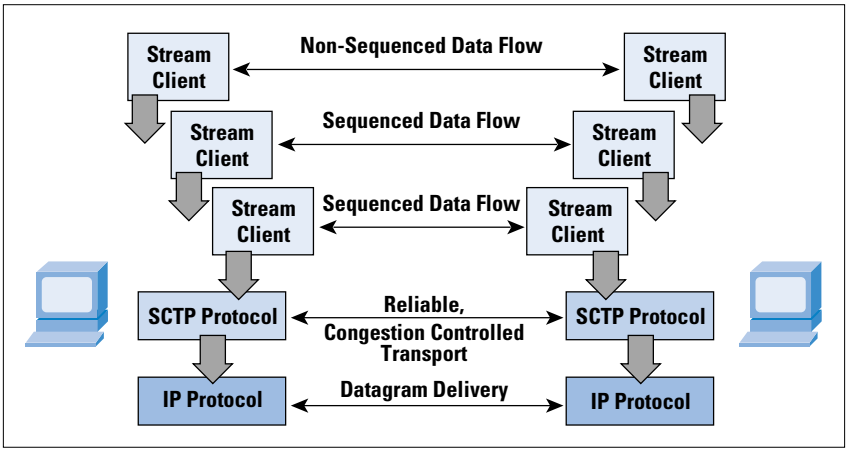


Figure 5: The SCTP Transport Service Model

SCTP explicitly uncouples transport-level reliability and congestion control from per-stream sequenced delivery through the use of a separate transport-level interaction. The transport-level data and ACKs and the corresponding transport-level congestion window controls operate using a transport-level sequence space. This sequence space counts transport-level messages, not byte offsets within the message, so that no explicit window scaling option is necessary for SCTP. The congestion control functions reference those of TCP with fast retransmit and fast recovery, with an explicit specification of the SACK protocol and specification of the maintenance of the transmission timers and congestion control. SCTP also requires the use of MTU path discovery, so that larger transactions will use SCTP-level segmentation, avoiding the IP retransmission problem with lost fragments of a fragmented IP packet. SCTP does use a modified retransmission mechanism to that of TCP. Like TCP, SCTP associates a retransmission timer with each message, and if the timer expires the message is retransmitted and SCTP collapses the congestion window to a single message size. The SCTP receiver will generate SACK reports for a minimum of every second received packet.

If a message is within a SACK gap, then after three further such SACK messages, the sender will immediately send the missing messages, and half its congestion window, analogous to the fast retransmit and fast recovery of TCP.

The use of multiple endpoint addresses assumes that each of the endpoint addresses is associated with the same end host, but with a potentially different network path between the two endpoints. SCTP refreshes path availability to each of the endpoint addresses with a periodic keepalive, so that in the event of primary path failure, SCTP can continue by using one of the secondary endpoint addresses.

One could describe SCTP as being overly inclusive in terms of its architecture, and there is certainly a lot of capability in the protocol that is not contained within TCP. The essential feature of the protocol is to use a single transport congestion state between two systems to allow a variety of applications to attach as stream clients. In itself, this is analogous to TCP multiplexing. It also implicitly assumes that every stream is provided the same service level by the network, an assumption shared by almost all transport multiplexing systems. The essential alteration with SCTP is the use of many transport modes: reliable sequenced message streams, reliable sequenced streams with interrupt message capability, and reliable nonsequenced streams. It remains to be seen whether the utility provided by this protocol will become widely deployed within the Internet environment, or whether it will act as a catalyst for further evolution of transport service protocols.

Sharing TCP information—Endpoint Congestion Management

The notion of sharing a single TCP congestion state across multiple reliable streams is one that may also be applied to a mix of reliable and nonreliable data streams that operate concurrently between a pair of endpoints. It is this form of the multiplexing service model that is explored by the congestion manager model. The Congestion Manager is an end-system module that allows a collection of concurrent streams from the host to a single destination to share a common congestion control function, and permits various forms of reliable and nonreliable streams to use the network in a way that cooperates with concurrent congestion controlled flows[19].

One of the major motivations for the congestion manager is the observation that the most critical part of network performance management is that of managing the interaction between congestion-controlled TCP streams and nonresponsive UDP data streams. In the extreme cases of this interaction, either traffic class can effectively deny service to the other by placing sufficient pressure on the network queuing resources that starve the other traffic class of any usable throughput. The observation made in the motivation for the congestion manager is that applications such as the Web typically open up a set of parallel connections to provide service, sending a mix of reliable flow-controlled data

along one connection and unreliable real-time streaming content along another. If the set of flows used a common congestion-control function at the sending host, the collection of flows would utilize the network resources in a manner analogous to a single TCP connection.

The manner of providing this common congestion control function is an advisory function to applications, as shown in Figure 6. One mechanism is that of a *callback,* where an application inserts a request to send a single message segment with the congestion manager. The Congestion Manager responds with invoking a callback to the requestor when the application may pass the data segment to the protocol driver. The other supported mechanism is that of *synchronous transmission,* where the Congestion Manager has a callback function that updates the application with a maximal available bit rate, the smoothed round-trip time estimate, and the smoothed linear deviation in the round-trip time estimate. In this mode the application can request further notification only when the network state changes by some threshold amount.

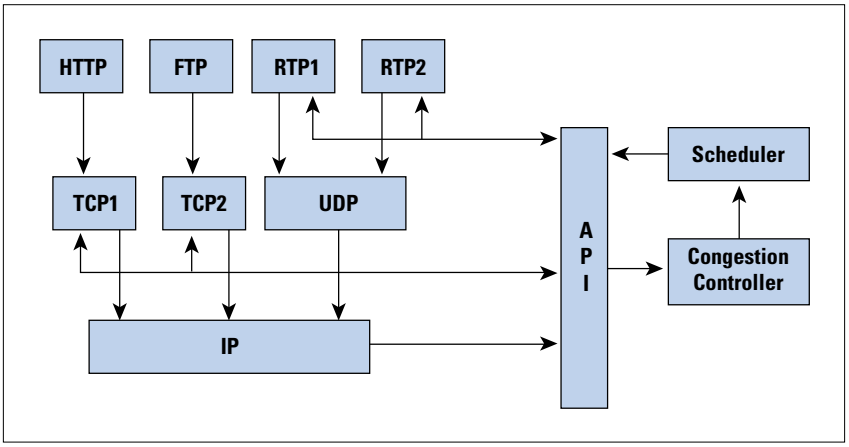


Figure 6: The CM Model, (after "The Congestion Manager"[19])

For the Congestion Manager to maintain a current picture of the congestion state of the path to the destination, each active stream needs to update the congestion manager as to the response from the remote host. It does this by informing the congestion manager of the number of bytes received, the number of bytes lost, and the RTT measurement, as measured at the application level. The application is also expected to provide an indication of the nature of the loss, as a timeout expiry, a transient network condition, or based on the reception of an ECN signal.

There has been little practical experience as yet with this model of shared congestion control within the Internet environment. There also remains a number of issues about how network performance information is passed back from the receiver to the sender in the absence of an active concurrent TCP session. The concurrent operation of a TCP session with a UDP streaming session to the same destination allows Congestion Manager to use the TCP congestion state to determine the sending capability of the streaming flow.

If the TCP session is idle, or if there is no TCP session, then the UDP streaming application will require some form of receiver feedback. The feedback will need to report on the span of data covered by the report, and the data loss rates and jitter levels, allowing the sender to assess the current quality and capacity of the network path.

This approach, and that of SCTP, are both illustrative of the approach of unbundling the elements of TCP and allowing applications to use combinations of these elements in ways that differ from the conventional monolithic transport-level protocol stack, with the intention of allowing the TCP congestion control behavior to be applied to a wider family of applications.

Better than TCP?

Recently, numerous "better-than-TCP" protocol stacks have appeared on the market, most commonly in conjunction with Web server systems, where the performance claim is that these protocol stacks can interoperate with standard TCP clients, but offer superior download performance to a standard TCP protocol implementation.

This level of performance is achieved by modifying the standard TCP flow control systems in a number of ways. The modified implementation may use a lower initial RTT estimate to provide a more aggressive startup rate, and a more finely grained RTT timer system to allow the sender to react more quickly to network state changes. Other modifications may include using a larger initial congestion window size or may use an even faster version of slow start, where the sending rate is tripled, or more, every round-trip time interval. The same technique of incremental modification can be applied to the congestion avoidance state, where the linear rate increase of one segment size per round-trip time interval can be increased to some multiple of the segment size, or use a time base other than the round-trip time for linear expansion of the congestion window. The backoff algorithm can also be altered such that the congestion window is reduced by less than half during congestion backoff. Resetting the TCP session to slow-start mode following the ACK timeout can also be avoided in such modified protocol implementations.

These techniques are all intended to force the sender to behave more aggressively in its transmission of packets into the network, thereby increasing the pressure on the network buffers. The network is not the only subject of this increased sending pressure; such modified protocol systems tend to impose a significant performance penalty on other concurrent TCP sessions that share the path with these modified protocol hosts. The aggressive behavior of the modified TCP systems in filling the network queues tends to cause the other concurrent standard TCP sessions to reduce their sending rate. This in turn opens additional space in the network for the modified TCP session to increase its transmission rate.

In an environment where the overall network resource-sharing algorithm is the outcome of dynamic equilibration between cooperative sending systems, such aggressive flow control modification can be considered to be extremely antisocial behavior at the network level. Paradoxically, such systems can also be less efficient than a standard TCP implementation. TCP server systems modified in this way tend to operate with higher levels of packet loss because their efforts to saturate the network with their own data packets make them less sensitive to the signals of network congestion.

Consequently, when delivering large volumes of traffic, or where there are moderately low levels of competitive pressure for network resources, the modified TCP stack may often perform less efficiently than a standard TCP implementation. Accordingly, these modified better-than-TCP implementations remain in the experimental domain. Within the production environment, their potential to impose undue performance penalties on concurrent TCP sessions and their potential to reduce overall network efficiency are reasonable indicators that such modified stacks should be used in private network environments, and with considerable care and discretion, if at all. Their utility in the public Internet is highly dubious.

TCP Evolution

The evolution of TCP is a careful balance between innovation and considered constraint. The evolution of TCP must avoid making radical changes that may stress the deployed network into congestion collapse, and also must avoid a congestion control "arms race" among competing protocols[20]. The Internet architecture to date has been able to achieve new benchmarks of network efficiency, and translate this carriage efficiency into ground-breaking benchmark prices for IP-based carriage services. Much of the credit for this must go to the operation of TCP, which manages to work at that point of delicate balance between self-optimization and cooperative behavior.

Widespread deployment of transport protocols that take a more aggressive position on self-optimization will ultimately lead to situations of congestion collapse, while widespread deployment of more conservative transport protocols may well lead to lower jitter and lower packet retransmission rates, but at a cost of considerably lower network efficiency.

The challenges faced with the evolution of TCP is to maintain a coherent control architecture that has consistent behavior within the network, consistent interaction with instances of data flows that use the same control architecture, and yet be adequately flexible to adapt to differing network characteristics and differing application profiles. It is highly likely that we will see continued innovation within Internet transport protocols, but the bounds of such effort are already well recognized.

We can now state relatively clearly what levels of innovation are tolerable within an Internet network model that achieves its efficiency not through enforcement of rigidly enforced rules of sharing of the network resource, but through a process of trust between competing user demands, where each demand is attempting to equilibrate its requirements against a finite network capacity. This is the essence of the TCP protocol.

Appendix: TCP Performance Models

This appendix is an extract from "Advice for Internet Subnet Designers," work in progress[15].

The performance of the TCP AIMD Congestion Avoidance algorithm has been extensively analyzed. The current best formula for the performance of the specific algorithms used by Reno TCP is given by Padhye et. al.[21], this formula is:

$$BW = \frac{MSS}{(RTT \times \sqrt{(1.33 \times \rho)}) + (RTO \times \rho \times [1 + 32 \times \rho^2] \times min(1, 3 \times \sqrt{0.75 \times \rho}))}$$

MSS is the segment size being used by the connection.
RTT is the end-to-end round-trip time of the TCP connection.
RTO is the packet timeout (based on RTT).
ρ is the packet loss rate for the path (that is, 0.01 if there is 1-percent packet loss)

This is currently considered to be the best approximate formula for Reno TCP performance. A further simplification to this formula is generally made by assuming that RTO is approximately 5 × RTT.

TCP is constantly being improved. A simpler formula, which gives an upper bound on the performance of any AIMD algorithm that is likely to be implemented in TCP in the future, was derived by Ott, et.al.[22, 23].

$$BW = 0.93 \times \frac{MSS\ 1}{RTT\sqrt{\rho}}$$

Assumptions of these formulae:

- Both of these formulae assume that the TCP Receiver Window is not limiting the performance of the connection in any way. Because the receiver window is entirely determined by end hosts, we assume that hosts will maximize the announced receiver window in order to maximize their network performance.
- Both of these formulae allow for bandwidth to become infinite if there is no loss. This is because an Internet path will drop packets at bottleneck queues if the load is too high. Thus, a completely lossless TCP/IP network can never occur (unless the network is being underutilized).
- The RTT used is the average RTT including queuing delays.

- The formulae are calculations for a single TCP connection. If a path carries many TCP connections, each will follow the formulae above independently.
- The formulae assume long-running TCP connections. For connections that are extremely short (<10 packets) and don't lose any packets, performance is driven by the TCP slow-start algorithm. For connections of medium length, where on average only a few segments are lost, single-connection performance will actually be slightly better than given by the formulae above.
- The difference between the simple and complex formulae above is that the complex formula includes the effects of TCP retransmission timeouts. For very low levels of packet loss (significantly less than 1 percent), timeouts are unlikely to occur, and the formulae lead to very similar results. At higher packet losses (1 percent and above), the complex formula gives a more accurate estimate of performance (which will always be significantly lower than the result from the simple formula).

Note that these formulae break down as ρ approaches 100 percent.

Addendum: An Update on Explicit Congestion Notification

The previous article on TCP performance noted that there was no explicit standardization of the IPv4 header field to carry the *Explicit Congestion Notification* (ECN) signals. As an update to the status of ECN, RFC 2481, the document that describes ECN, categorizes this proposal as an "Experimental" RFC document[27]. The Internet Standards process[28] describes this category as follows: "The 'Experimental' designation typically denotes a specification that is part of some research or development effort. Such a specification is published for the general information of the Internet technical community ..." ECN is the only experimental proposal to use these two bits of the IP header, and the use of the category "Experimental" reflects the current status of the proposal, in that the Internet Engineering Steering Group has, at the time of publication, yet to make a final decision to allocate these two bits of the IP header to ECN.

Some encouragement to use ECN is certainly timely. As RFC 2481 notes: "Given the current effort to implement RED, we believe this is the right time for router vendors to examine how to implement congestion avoidance mechanisms that do not depend on packet drops alone. With the increased deployment of applications and transports sensitive to the delay and loss of a single packet (e.g., realtime traffic, short web transfers), depending on packet loss as a normal congestion notification mechanism appears to be insufficient (or at the very least, non-optimal)."

References and Further Reading

[1] Huston, G., TCP Performance, *The Internet Protocol Journal,* Vol. 3, No. 2, Cisco Systems, June 2000.

[2] Huston, G., *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, John Wiley & Sons, January 2000.

[3] Postel, J., "Transmission Control Protocol," RFC 793, September 1981.

[4] Claffy, K., Miller, G., Thompson, K., "The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone," INET'98 Proceedings, Internet Society, July 1998. Available at: **http://www.isoc.org/inet98/proceedings/6g/6g_3.htm**

[5] Braden, R., "T/TCP—TCP Extensions for Transactions Functional Specification," RFC 1644, July 1994.

[6] Allman, M., Glover, D., Sanchez, L., "Enhancing TCP over Satellite Channels Using Standard Mechanisms," RFC 2488, January 1999.

[7] Jacobson, V., "Congestion Avoidance and Control," ACM SIGCOMM, 1988.

[8] Floyd, S., Fall, K., "Promoting the Use of End-to-End Congestion Control in the Internet," Submitted to IEEE *Transactions on Networking*.

[9] Allman, M., Paxson, V., Stevens, W., "TCP Congestion Control," RFC 2581, April 1999.

[10] Mogul, J., Deering. S., "Path MTU Discovery," RFC 1191, November 1990.

[11] Jacobson, V., Braden, R., Borman, C., "TCP Extensions for High Performance," RFC 1323, May 1992.

[12] Mathis, M., Mahdavi, J., Floyd, S., Romanow, A., "TCP Selective Acknowledgement Options," RFC 2018, October 1996.

[13] Allman, M., editor, "Ongoing TCP Research Related to Satellites," RFC 2760, February 2000.

[14] Wireless Access Protocol Forum, **http://www.wapforum.org**

[15] Karn, P., Falk, A., Touch, J., Montpetit, M., Mahdavi, J., Montenegro, G., Grossman, D., Fairhurst, G., "Advice for Internet Subnet Designers," work in progress, July 2000.

[16] Jacobson, V., "Compressing TCP/IP Headers for Low-Speed Serial Links," RFC 1144, February 1990.

[17] Casner, S., Jacobson, V., "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links," RFC 2508, February 1999.

[18] Stewart, R., et al., "Stream Control Transmission Protocol," work in progress, July 2000.

[19] Balakrishnan, H., Seshan, S., "The Congestion Manager," July 2000.

[20] Floyd, S., editor, "Congestion Control Principles," work in progress, June 2000.

[21] Padhye, J., Firoiu, V., Towsley, D., Kurose, J., Modeling TCP Throughput: A Simple Model and Its Empirical Validation, UMASS CMPSCI Tech Report TR98-008, Feb. 1998.

[22] M. Mathis, M., Semke, J., Mahdavi, J., Ott, T., "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," *Computer Communication Review,* Vol. 27, No. 3, July 1997.

[23] Ott, T., Kemperman, J., Mathis, M., "The Stationary Behavior of Ideal TCP Congestion Avoidance," available at:
ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps

[24] Floyd, S., Mahdavi, J., Mathis, M., Podolsky M., "An Extension to the Selective Acknowledgement (SACK) Option for TCP," RFC 2883, July 2000.

[25] Allman, M., Balakrishnan, H., Floyd, S., "Enhancing TCP's Loss Recovery Using Early Duplicate Acknowledgment Response," work in progress, June 2000.

[26] Allman, M., "TCP Congestion Control with Appropriate Byte Counting," work in progress, July 2000.

[27] Ramakrishnan, K., Floyd, S., "A Proposal to Add Explicit Congestion Notification (ECN) to IP," RFC 2481, January 1999.

[28] Bradner, S., "The Internet Standards Process—Revision 3," RFC 2026, October 1996.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Scientist in the Internet area for Telstra. He is also a member of the Internet Architecture Board, and is the Secretary of the Internet Society Board of Trustees. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@telstra.net**

Securing the Infrastructure

by Chris Lonvick, Cisco Systems

People are becoming much more reliant upon the proper operation of their networks. Consequently, the administrators of these networks are being tasked with providing an ever-increasing level of service. At this time of high reliance upon the network, methods and procedures need to be instilled into the network so the operators can maintain control of their network and they can know with some certainty the effect of each potential change. This may become increasingly difficult as network resiliency techniques are being proposed and deployed with the intent of automatically keeping these networks in top operation. Having a predictable network that is secured in a proper manner results in a network that is more suitable for the users and better meets the intended purpose of the network.

Most of the current network security models start with the physical perimeter of the network as its defining boundary. All things within this boundary are supposed to be protected from the perceived inimical forces that are outside of the perimeter. We are, however, finding that the perimeter of the network is no longer solidly defined. There are many exceptions to the "hard-shell perimeter" model—companies merge, remote sites are linked through *Virtual Private Networks* (Site-to-Site VPNs) across untrusted paths, access is granted in-bound for the network users through *Access Virtual Private Networks* (Access VPNs), and there are several other exceptions. For this article, let's consider a different model. This model has a boundary of the acceptable network users rather than any geographical or logical perimeter. It is important that these users are allowed access to the services provided by the network. It is equally important that the people who are not authorized to use the network must be prevented from consuming its resources and otherwise disrupting its services.

Other models tend to focus on the restrictions of the users to access devices to provide security to the network. This model, however, looks at the effect that the users and each of the devices have upon the state of the network. To conceptualize this model, visualize that the only time this network would be running at a "steady state" is when there is no user traffic, no administrative or management traffic, and no routing update changes. The insertion of any traffic, or the addition or removal of any device or link, would change the state of this network. Changes to the state of this network may come from any number of sources, but they can be seen as coming from four different, quantifiable areas.

- Operators may enable or disable lines and devices.
- A network device publishing a new route or a different metric to a destination may cause the remainder of the network devices to dynamically recompute paths to all other destinations.
- Servers may insert traffic.
- Users may insert traffic.

Of these, the last two should be the least disruptive to the network as long as the traffic amounts are within the predicted and acceptable ranges. Changes that are within the goals of the network—for example to provide a service to the users—are considered good, while changes that cause outages or other disruptions are to be avoided. As such, it is vital that the network administrators understand the potential impact and consequences of each possible change in their network.

In this model, then, the administrators must know and understand the influences that will change the state of the network. The desire to achieve this goal sometimes leads to improper restrictions placed upon the users. Consider one extreme case of this model where each change in the network must be stringently authorized and authenticated. As a narrow example, this would mean that even traffic that is fundamentally taken for granted as a proper process of the network would have to be authenticated and authorized. *Domain Name System* (DNS) transactions would show that this extreme case is impractical. Each DNS query would have to be associated with a user or authenticated process, and that user or process would have to be authorized to make each specific query. A vastly more practical case for real networks would be for the administrators to allow any DNS query from any device without authentication—as it is done in existing dynamic networks today. In the model, the normal DNS queries and responses would be an influence upon the state of the network. For this influence to change the network in a way that meets the goals of the network, the administrators would have to feel comfortable that the servers and the available bandwidth will adequately handle the amount of DNS traffic as well as all other traffic. On the other hand, the administrators do need to establish a strict set of rules for the influences that they consider sensitive or possibly disruptive to their network. Continuing this example, the administrators may want to place restrictions upon the devices and processes that can insert and update the DNS records. It would be rather inappropriate, and potentially devastating, if any unauthorized person or network device were allowed to overwrite any existing records. If anyone were allowed to perform any DNS update that he or she wished, chaos would soon result. There must be a center position for this example that allows the operators to maintain control but still permits the dynamic freedoms expected by the users. Specifically to address this, the DNS Extensions Working Group has proposed several Internet Drafts[1].

In the broader sense, this places a very heavy responsibility upon the people who are running the network. They must find some acceptable median between the desire to rigidly control all aspects of the network and the freedoms that are expected by the users, while at the same time satisfying the business requirements of their network. However, defining the freedoms and restrictions of the users is only one part of maintaining the network. The administrators and operators must have an understanding of the influences on the network as described in the model. In this, each aspect of the parts of the network must be under-

stood well enough to predict their behavior as they are normally used, and to limit the potential for disruption if they are used beyond their means. The one area that is vital to the proper working of the network is the infrastructure. This article explores some of the thoughts that may go into the process of securing the network infrastructure.

Table 1: Sources of Change to the Network

Sources of Change to the Network	Some Examples of How the Source Influences the Network	Examples of Device Types within the Network (The 4 Groups)	
Operators and their Devices	• Add/remove new lines and circuits • Install/remove network devices		
	• Login to the network devices to change their configuration • Poll network devices for their status	• Operations Consoles • Network Management Stations	Operators
Network Devices	• Dynamically route or switch traffic • Dynamically mark lines and circuits in or out of service and then use them accordingly • Authenticate users and permit their accesses accordingly • Dynamically assign addresses and register that information for retrieval by others	• Routers and Switches • Firewalls	Infrastructure Devices
		• Authentication Servers • DNS/DHCP Servers	
Servers	• Servers send content to User's workstations to fulfill their requests • Servers broadcast and multicast content to recipients	• Servers offering Content and Servers	Servers
Users and their Devices	• Client workstations request content from servers and upload content to servers • Client workstations utilize services that are offered within the network	• Client Workstations	Users
	• A user encourages many others to visit a particular web site which causes a stampede • A user tells others that a particular service is down or unavailable causing others to not attempt access		

Description of Problem

In this abstracted network model, four sources of change were noted. As shown in Table 1, these changes, or influences to the network, may come from the operators, the network devices, the servers, and the users of the network. Let's first look at the influences that each of these groups can effect upon the network by first categorizing the network devices. All the devices on the network may be somewhat separated into four groups that correspond to the four sources. These groups of network devices can be seen in the third column of the table.

- *Operators:* For the purpose of this article, let's describe the Operators as all the people who operate the network, including the network engineers, the installers, the people who monitor the net-

work, and all the other people who make it work. The first group then is made of the operators and the devices that these operators use to run the network, such as the network management stations and all other operations consoles. Operators periodically make changes for moves and additions for better network performance, or to overcome disruptions. They will also monitor the network through polling, receiving alerts, and sometimes directly interacting with the network devices. Generally the amount of traffic inserted into the network from their activities is minimal. Because they generally have physical access to all locations, they can insert or remove network devices. Operators can have influence over all aspects of the network at all layers—from the physical layer, all the way up the stack. Operators can influence the network either in band or out of band, and they should be the only people who directly access the network infrastructure devices such as the routers and DNS servers. Usually this access will be from the management platforms, but in many situations, operators require access from devices that would otherwise be classified as a user's workstation.

- *Infrastructure Devices:* The network infrastructure devices themselves have the ability to change the network as well. This is mostly done through the dynamic nature of the network. At some times the physical portions of the network might fail and cause outages. In some cases, such as self-healing ring topologies, physical-layer devices may heal the network. In other cases, such as when a router is taken out of the network for maintenance, the routing updates will heal the network to the best of their abilities. The network infrastructure devices can be somewhat separated into two categories. The first of these would be the infrastructure devices that have no direct interaction with the users of the network. This category would consist of the devices such as the routers, switches, access control devices, and perhaps even the physical-layer devices such as multiplexers and modems. The user machines and content servers normally would not form sessions or require any information from these devices. The second category would be the devices with which customers indirectly interact. These would be devices such as the DNS servers, *Dynamic Host Configuration Protocol* (DHCP) servers, *Network Time Protocol* (NTP) servers, authentication servers, and the like. The users and servers would form sessions with these supporting devices and would require information from them for the basic operation of the network. In some cases, such as with a DNS/DHCP server, the results of the indirect user interaction would even update the servers with information. This latter group may be called "supporting devices." These two categories can be taken together with all the wires, circuits, and lines to form the infrastructure of the network. Although the users do not actively see their presence, this infrastructure must be available and functioning before any user can actually do anything productive on the network.

- *Servers:* The servers in this group are those that contain content or services with which the users directly interact. These would be databases, Web servers, application servers, and the like. Like the operators group, this group is not considered to be part of the network infrastructure.
- *Users:* The users and their machines constitute the bulk of the network. The changes that the users make upon the network will probably come through transferring content or requesting and utilizing services. They can change the nature of the network by withdrawing from the network, or by causing others to withdraw from the network. In a nonmalicious way, the user base can degrade the state of the network by using it beyond its expected capacity. In certain situations, users with malicious intent may find exploitable network vulnerabilities. In most normal cases, however, the influence from the users upon the network will be through their interactions with the servers.

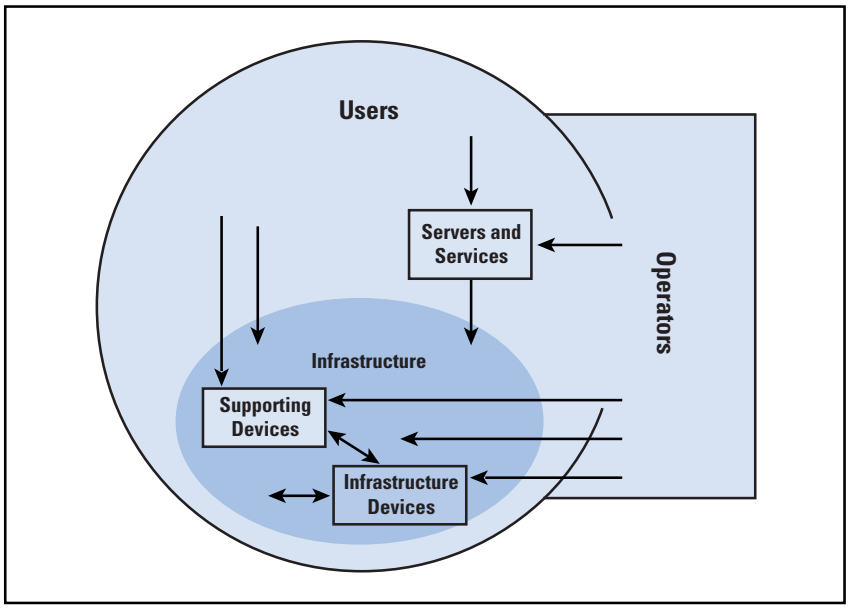
Each type of influence may also be considered to have a different weight. For example, the insertion of a new router into an existing network would be expected to have a larger effect upon the operations of the network than the change to the network caused by a user retrieving some information through a Web browser. To quantify some of the expected network changes, consider that there may be spheres and levels of influence. Any influence that may cause a change over the entire network may be considered to have a global sphere of influence. A router recently inserted into the network would start exchanging routing information with its neighbors. With no restrictions placed upon routing updates, this router could announce a new network, or it could announce the best path to an otherwise difficult-to-reach network. The remainder of the network would be affected, and all other routers would have to recalculate their paths. If the announcements were true, then the network would continue servicing the needs of the users. If the announcements were false, possibly because of an incorrect configuration, then the whole network could suffer. In this case, it is possible to limit the sphere of influence by restricting the acceptance of routing updates. In one method, all the routers could be restricted to disallow the acceptance of an announcement to the "default" network. Additionally, all the routers may be restricted to accept only announcements that are known to be within an acceptable address range. In another method, the routers could be grouped to accept announcements only from a select set of other routers. Additionally, some routing protocols have an option to include an authentication and integrity check through signing the updates. Any of these methods would help to reduce the sphere of influence and thus the potential for changes that could be made by the insertion of a router. There is, however, a cost associated with this; the operators would have to diligently enforce this control.

The level of influence can also be considered a factor in this model. The sphere of influence of a single transmission line can be defined to include any portion of the network that uses that line. If that line develops a fault, it may corrupt or discard packets and the associated network devices may automatically disable that line. If there is a backup line or an alternate path, then this change will be a small problem to the operations staff and its loss may go unnoticed by the users. That would be a low level of influence upon the network. On the other hand, if the line has an intermittent fault that can cause a route flap, or if the line has no backup, then major disruptions may occur. That would be considered a high level of influence.

If the goal of the network is to provide a service to its users, then its operators must try to quantify each of the influences. In a theoretically ideal network, the administrators would appropriately limit the sphere and would try to minimize the level for every influence. As was noted above, however, attempting to do this would require numerous operations tasks. Many of those may be unnecessary for their specific environment. For example, in a small business where there is a high degree of trust that no one has any malicious intent, controls would still be placed upon the influences that would most probably cause network problems through accidents. If the security policy allowed anyone to connect any device to the network, it may still be prudent to disallow the routers from receiving routing updates from any source other than the other routers.

A well-running network is the result of a well-controlled network. These networks must have a separation of authorized administration from other influences, and these other influences must be understood well enough to know how they will change the network. The following diagram shows the network and the groupings of influences upon it, and the table below that describes the elements of this model. This model does not show access paths, but rather the influences that each grouping of devices has upon the infrastructure and upon other devices. As can be seen, the users are pervasive throughout the network (because they are a principal reason for its existence), and they must have the access paths to contact the servers and necessary infrastructure devices. The users will influence the infrastructure as they insert traffic upon the lines, but they should have no direct influence upon the infrastructure devices such as the routers and digital access cross-connects. The operators do have influence upon the infrastructure devices and must have an access path to those devices. It would be most appropriate if the users were not allowed to usurp the access paths of the operators. However, because the two are sometimes nearly indistinguishable, the task of separating the administrative channels from the user channels becomes difficult.

Figure 1: The Network Security Model



The following table describes the elements in this model.

Table 2: Network Model Elements

Element	Description	Example
Operators	The devices and people who operate, manage and support the network	Monitoring and Management Workstations, Syslog servers
Infrastructure	This composite area denotes the entire infrastructure. This is broken out to show the actual infrastructure devices as well as the supporting devices.	• Infrastructure Devices: Routers and Switches • Supporting Devices: DNS and DHCP servers • All other infrastructure components: wires, circuits, DSU/CSUs, SONET equipment, repeaters, etc.
Servers and Services	The devices that host content and services for the users	Web servers, file servers
Users	All of the users of the network and their workstations	Alice, Bob, Carol, Dan and their workstations
Arrows	Define which element influences or changes which other element	Users insert traffic into the network and thus influence the Servers and Services. Operators may also influence each of the components of the Infrastructure

Know Your Business

All well-running networks must have a *security policy* defined. This must reflect the goals of the network and must also be acceptable to the users and administrators. There are good examples of policies as well as methods that can be used to generate them. RFC 2196[2] contains several thoughts about constructing a policy and SANS[3] offers courses on this. While defining a network security policy, it will be advantageous to list the most likely disruptive influences to the network. This is commonly called *The Threat Model*. All potentially disruptive factors should be considered when forming the threat model, and they must be addressed when writing the security policy. It may, however, be beyond the capabilities of the operations staff to negate all of them. It may also be prohibitively expensive to try. In those cases, the writers of the policy should acknowledge the factors that won't be negated, but they should still find ways to minimize them. For example, in an Enterprise network the operators are somewhat likely to require access to routers and switches from any physical location in the network. In Service Provider networks, there may be less of a chance of that because the operators traditionally reside with the network management devices. In both cases, it would not be considered good for the network if a user could gain control of a router. The security policy for an Enterprise network may explain that network access to routers will be opened and available for any other device within the network. This will allow any operator to access the routers from any location. On the other hand, the security policy for a Service Provider network may state that access to the routers will be opened only for specific address ranges. Implementing this will prevent users, who reside within the address spaces assigned to the users, from accessing the infrastructure devices while allowing the operators, who reside within their own address space, to access the routers. In both cases, however, strong authentication will probably be required to additionally limit access to only authorized people.

Within the business of the network, operators must have the ability to control the infrastructure devices. Traditionally, the ways to interact with a device have been called "interfaces." A terminal with keyboard attached to the console port of a router is an interface just as is a Web browser accessing the router via the *Hypertext Transfer Protocol* (HTTP) through the network. Also, the path between the controlling device and the infrastructure device has traditionally been called a "channel." The wire connecting the terminal to the router is a channel just as is the TCP session that transports the HTTP in the prior example. The channels between the operators and the infrastructure devices must be secured, as well as the channels between the infrastructure devices. The first step in obtaining this goal is to identify all of the interfaces needed by the operations staff to access each of the remote devices. Along with this, they also need to identify each of the interfaces needed for the proper functioning of the network. The following lists are some of the possible network-available interfaces to some of the infrastructure devices in a dynamic network. This is somewhat broken down

into the interfaces needed by the operations staff, the interfaces usually needed by the other infrastructure devices, and some ancillary interfaces.

Table 3: Some Interfaces of Infrastructure Devices

Operations and Interfaces	Infrasturcture Interfaces	Ancillary Interfaces
telnet, Kerberized telnet, SSH, rsh, rcmd, rexec, HTTP, FTP, tftp, rcp, scp, SNMP, LDAP, COPS, Finger	Syslog, ICMP, DNS, DHCP, RIP, OSPF, BGP, IS-IS, IGRP, EIGRP, HSRP, NTP, SNMP, Multicast controls	RADIUS, TACACS+, Kerberos Authentication, PAP, CHAP, EAP, chargen, echo, time, discard, Auth (Ident)

Each of these interfaces may be exposed to the nefarious forces that are known to inhabit large networks, and each of these exposures has vulnerabilities that may be exploited. Telnet sessions may be hijacked, DNS queries may be answered by nonauthoritative and possibly maliciously incorrect responses, and sinister people can insert forged routing updates to confound and disrupt the network. The network security policy should expect that these vulnerabilities may be exploited and it should address the mechanisms that may be used to either negate the vulnerabilities or to minimize the exposures. In this model, the process may be used to limit the sphere and level of the influences. The policy may also make some attempt to identify the potential consequences of the disruption caused by the exploitation of these interfaces. It should also describe an escalation procedure for dealing with encountered problems.

Possibly, during the exercise of identifying the open interfaces in an existing network, some of them may be closed or removed if it is determined that they are not needed or if their function can be fulfilled by the use of another interface. As an example, consider a UNIX host that has both the *Secure Shell Protocol* (SSH) and *finger* services running on it. If the policy of the network is to tightly control the information that anyone can obtain from any device, then the operators may want to remove the *finger* service. The operators will be able to obtain similar information by running the *who* command on the UNIX system through an SSH remote execution request. On the other hand, if the operations processes have been built upon the format of the information returned by *finger*, then the operators may want to prevent direct access to *finger* from the network and require that it be run on the device or through the SSH request.

At some point, it would be a good idea to run a scanner against the infrastructure devices. The *Network Mapper* (NMAP)[4] is a freely available tool that can pick out some of the active interfaces of a device. This, or a similar tool, should be periodically used by the operations staff to ensure that the open ports of an infrastructure device are those that are known to be open. This investigation should not be limited to operations channels, but should also include application channels. For example, the question should be asked if the operations workstations should have open application interfaces—such as *Simple Mail Transfer Protocol* (SMTP) or *Network File System* (NFS). There are exploitable

vulnerabilities associated with some application interfaces that should be addressed in the security policy. In most cases, it would be prudent to remove applications that are not needed from infrastructure devices and supporting servers, as well as from operations devices when they are not needed. In all cases, it is usually considered to be a good practice to review the entries in the *inetd* configuration in UNIX systems.

It should be remembered that there will almost certainly be an access path between the users and the network interfaces of the infrastructure devices. The network security model diagram shows that neither the users nor the servers should have any direct influence to change or control the infrastructure devices. This is somewhat analogous to the policy of giving privileges on a multiuser system. In most well-run multiuser computing systems, the operators give only the most meager of privileges to the users of the system. This prevents most accidental and malicious disruptions. If the users need to run a privileged process or to access the files of other users, processes that utilize setuid are used or consensual groups are established. Generally, efforts are made to prevent users from having significant privileges on these machines. The alternative of giving each user high-level privileges usually results in disaster after a short time because the users then have the ability to overwrite or delete files, and may run processes that are generally disruptive to the operating system and to others.

Similarly, giving users high-level access to the routers of a network would have a deleterious effect. In the case of *Quality of Service* (QoS), a user given the privileges to reconfigure routers along a path would be able to provide his/her own designated flows with bandwidth and priority assurances. Subsequent users would also have that capability, and their modifications may leave the first user without his/her expected QoS—and possibly without a session at all. A far better mechanism to fairly deploy QoS is through the use of a brokering service. In a "policy network," users or authenticated processes may request a level of service for their flows through a *Policy Manager*. This Policy Manager should have the capability to arbitrate requests to provide a semblance of fairness. The Policy Manager would then directly control the appropriate routers within the rules established by the administrators.

Along these lines, conveying security-related policy to infrastructure devices should take a similar path. For example, if the network security policy states that user access to a particularly sensitive network resource must be authenticated and controlled, the operators may elect to place a firewall between the users and that resource. That firewall would be classified as an infrastructure device and users should not directly access or control it. Rather, the users may authenticate themselves to an authentication service, which would notify the firewall that their access to the resource is permitted or denied. The authentication service may also send a set of restrictions for the access method; it may permit HTTP access but deny Telnet and *File Transfer Protocol* (FTP) for one person, but for another it may permit only Telnet.

The reasons for authentication, authorization, and access control must be described in the network security policy. It would be simple to mandate strict controls at many places in the network. However, that may not meet the needs of the business or the tolerance of the users. More to the point in this article is the requirement in the model that the disruptive influences be negated or minimized. Having a firewall or other access control device silently discard disruptive packets may be preferable to having a user or unconstrained process continue to spew garbage around the network.

Decide on the Methods of Securing the Channels and Interfaces

Some of the very first computing devices were designed to be managed locally and not remotely. Consoles consisting of a teletype device and a roll of paper were among the first interfaces to modern computing devices. Various methods were devised to extend these administrative interfaces beyond the confines of the frigid "Computer Room." The first efforts were to keep these interfaces out of band, a scenario that meant separate wires from the physical port on the machine to a console in the operations room. In many cases, the wires from the remote terminals to the system were still visible because they were laid along the floor and could, therefore, be considered a secure channel. While this maintained a secure administrative channel—or path—that could not be tapped or exploited by others, it didn't scale as more and more computing and ancillary devices were placed into the computer room, each requiring its own console. When remote terminals became commonplace, administrative functions were allowed over that channel. In almost all cases, the operating systems were mature enough to require some form of authentication before critical management operations were allowed.

The out-of-band channels for secure remote administration of devices may no longer be applicable to large networks. There are costs associated with running separate secure networks for the sole purpose of out-of-band operations, and there is the impracticality of one-at-a-time access through the console port of each device. This applies equally to the practice of placing a modem on the console ports of devices—a deployment that is not considered secure because there are still many automated dialers looking for answering modems. For these reasons, in-band access of operations has become the preferred method for modern networks. Telnet has been the oldest remote channel—and interface—for remote operations. Since then, other remote interfaces have been opened for controlling, commanding, and operating devices.

Many attempts have been made to "secure" Telnet and its use as a command and control channel. These efforts address the vulnerabilities of the protocol, and some address the interface itself. The *Berkeley Software Distribution* (BSD) "r" command set, such as *rlogin, rsh, rexec,* and others, were meant to be a substitute for the most common uses of Telnet within a trusted environment. It was assumed that the person initiating the command had previously been successfully authenticated.

SSH was meant to be a secure replacement of the Berkeley "r" tools. The SSH console session has been widely deployed to remotely operate devices. This replicated the Telnet interface while replacing the channel. The protocol addressed machine authentication, user authentication, and session confidentiality and integrity. When used as it was intended, it can effectively replace Telnet as a secure interface and channel. The *scp* feature of SSH can also securely replace *rcp,* and it has been used as a replacement for FTP. Likewise, a Kerberized Telnet and Kerberized FTP have been released to do the same.

Several other efforts have also been undertaken to secure some of the other administrative interfaces and channels. For example, the security issues of *Simple Network Management Protocol* (SNMP) are being addressed with the options of SNMPv3[5]. Also, applications that utilize HTTP can be secured with HTTP over SSL (HTTPS) (*Secure Sockets Layer/Transport Layer Security* [SSL/TLS])[6]. At this time, it appears that SSL/TLS is emerging as a mechanism that can be utilized to provide some security to many different applications. Beyond the operational interfaces and channels, work has been done to secure some of the infrastructure and ancillary interfaces. Some routing protocols have built-in authentication and integrity through the use of signing the routing updates with a shared key. Each mechanism that has been secured has been the subject of a focused effort to address that specific interface and channel. However, unlike those named above, some channels, such as Syslog and *Trivial File Transfer Protocol* (TFTP), have not been explicitly secured at this time.

IP Security (IPSec)[7] was developed as a general-purpose mechanism that may be used to provide a secure wrapper around any unicast flow. Its cryptographic mechanisms can provide strong authentication, confidentiality, and integrity. While IPSec can be used to secure any flow, it may require additional infrastructure. A *Public Key Infrastructure* (PKI) must be established within the network. The alternative is to use preshared keys, a solution that is operationally intensive and doesn't scale well. IPSec also requires consistent time synchronization between the devices, as well as a consistent DNS. If these pieces are in place, the operations staff can utilize IPSec to secure each of the needed operations channels. If the operators and administrators choose this method, then they should ensure that the unsecured channels are unavailable to anyone but themselves. For example, if the Telnet channel is secured with IPSec, then the Telnet port on remote devices should be closed for inbound access.

One method of closing the exposures is through *Access Control Lists.* Routers and switches usually have mechanisms that can be used to allow inbound and outbound sessions from only certain devices. UNIX devices usually have the ability to run TCP wrappers that can provide access-control mechanisms for inbound and outbound sessions. If infrastructure devices can be grouped together, the operators may decide to

place them behind an internal firewall. The decision to do that should be thought through. Generally the internal firewall will limit access of the protected devices to the specified interfaces[8]. If this is done for a group of network management stations, the net effect may be that any attempts to access those workstations from outside of the firewall would be denied. The only inbound flows may be SNMP responses and traps. This implementation would limit the operations staff to being physically present before they could operate those devices. On the other hand, the firewall would prevent users from mistakenly or intentionally forming sessions with those devices. Because any received packet would have to be assessed by the device, a firewall that would discard packets before they are received by the device would help to prevent denial-of-service attacks. The use of internal firewalls should not be used as an excuse for poor security measures on the protected devices. Regardless of how effective the operators feel their firewall is, the protected devices must be treated as if they were otherwise exposed.

In determining the channels that will be used for the administration of the infrastructure devices, the packages will also be selected. At this time, many devices are sporting Telnet, FTP, and HTTP channels and the operators may utilize workstations that have these packages already loaded onto them. Also, networks comprising Microsoft NT servers may be managed remotely by the NT administrative tools, which commonly run on NetBIOS over TCP/IP (NBT). When given the choice, most often the operations staff will select easy-to-use and commonly available packages to access the interfaces of the infrastructure devices for remote operations and control. In all cases, these will be packages that will be available to the user community of the network as well. The users of the network may also easily download packages of these types if they don't already have them on their machines. For example, the operators may choose to utilize SSH for secured access to some devices. It is a trivial task for the users to also download an SSH client package and to start poking around the network to see what they can find. Even SNMP packages can be easily downloaded to the workstations of the users.

The operators and administrators must avoid the temptation to select a less-well-known package for infrastructure management based upon the thought that the users probably won't know about it. Users may not be initially aware that some packages are being used, but they can also download sniffer packages. Given enough time, even passive sniffing will give them enough clues to determine the channels used for administration. When they know that, they can then probably download the package themselves, and may then attempt to use it to explore the network. It should also be noted that the more heavily used packages have been scrutinized much more than the newer or less used packages. As a very general rule, the older a package gets, the more it becomes trusted because more people have been using it and *probably* attempting to break it.

As described above, and as it is seen in the diagram of the model, some of the channels that are available to the operators are also available to the users. This means that if the operators utilize Telnet to control their routers, it may be possible for a user to also initiate a Telnet session to a router. There must be an extremely strong discriminator to differentiate between the authorized operators and the unauthorized users before access to control the device is granted. Almost exclusively, the discriminator used is some form of authentication. An operator should be able to satisfy an authorization challenge, whereas an unauthorized user should not. A username and password is the most common form of in-band authentication. Specifically within Telnet and FTP, an in-stream challenge is presented to the user attempting a session; the user is asked for a username and then for a password. If these credentials match the values stored on the host, then the session is permitted. In these sessions, the credentials are exposed to casual observation. Anyone with a packet-sniffing device will be able to plainly see the username and password. These credentials must be regarded as secrets that must be protected. If they are compromised or stolen, then the operators have lost their control of their network. Some packages, such as SSH and Kerberos, have addressed these problems and have found ways to prevent secrets from being passed during authentication.

It must also be noted that some infrastructure devices do not offer any in-band channels for control. Many *Channel Service Units/Data Service Units* (CSU/DSUs) are not IP aware and do not offer any in-band channels for control. In cases like those, physical access may be the discriminator that prevents unauthorized users from controlling the device. Typically, a lock on a door or a cabinet would be the "challenge," and the key would be the authentication credential, which must be treated like a secret. It cannot be emphasized enough that these secrets must be protected. The *CERT Coordination Center* has written a very broad Tech Tip, which explores the topic of password security[9]. Many companies have found it very beneficial to periodically hold training courses to highlight the importance of this subject both to their operators and to their users.

Ancillary Channels Also Require Security

One of the parallel problems with using authentication credentials is its distribution. Many devices are capable of maintaining a local database of usernames and passwords. However, maintaining identical databases on each device throughout large networks is infeasible. More often, the authentication credentials are stored in a centralized database and an *Authentication, Authorization, and Accounting* (AAA) protocol is used to transfer them as needed. The AAA protocols most often used are *Remote Access Dial-In User Service* (RADIUS), TACACS+, and *Kerberos* authentication. Each of these has different characteristics and security mechanisms. Kerberos authentication was designed to securely transport authentication material. A password is never transferred across the network in this architecture. This protocol has withstood the test of

time, but it has been difficult to establish in networks that aren't committed to maintaining it. This situation seems to be changing because more "productized" versions are becoming available on the market. TACACS+ has a mechanism to hide the exchanges between the TACACS+ client and the server. It is also capable of transferring authorization rules for each user. RADIUS uses a mechanism to hide portions of the exchange between the RADIUS client and server as well.

Beyond this, the channels for telemetry, audit, and accounting may need to be secured. There are no inherent mechanisms to secure syslog at this time, and SNMPv1 may be protected with a Community String, but that solution is considered weak. It is possible to allow read-only access to the SNMP interface, but SNMPv3 has many of the security features that have been requested to secure this protocol. Other channels that are required by the operations staff should also be critically reviewed because many forms of attacks are on open channels.

It would be appropriate for the operations staff to keep up with new exploits and to assume that the users of the network have access to the latest "hacker" tools. It is quite common for people to hear about an exploit or published vulnerability and then "try it out" in the nearest available network. For this reason, it should be in the security policy of the network that "security patches" be given the highest priority and should be loaded on the affected platforms as soon as they are available and have been approved for the environment.

Conclusions

When any security mechanism is applied, the appropriateness and applicability of the solution should be questioned. On the surface, some security solutions may appear to be good; however, their applicability to the situation must be verified. As an example, SSL may be used to secure HTTP traffic, and it is commonly found in many Web browsers. Unfortunately, not many people explore the browser options that are enabled by default. In most browsers, SSLv2 is still available, even though it has published and exploitable vulnerabilities. Additionally, even in SSLv3—which negates the vulnerabilities of SSLv2—low key-length cipher suits are still available and enabled by default. In many cases, a null-cipher crytpo algorithm is available. In the internal networks of many companies, SSL may be selected and implemented using a self-signed certificate. Care must be taken to ensure that this certificate is the one distributed to each administrative workstation. SSL sessions may be formed without certificates supplied by either endpoint. An attacker could exploit this through a man-in-the-middle attack. Another example would be the use of SSH. SSHv1 has known vulnerabilities. If the administrators decide to deploy SSH for the control of the remote infrastructure devices, they should first decide if they should be worried about attacks against those known vulnerabilities in their infrastructure. If they are, then they should either deploy SSHv2, which addresses the vulnerabilities of SSHv1, or they should explore the use of Telnet with IPSec.

In many cases, rather than using the "most secure" solution, perhaps a simpler solution would still provide adequate protection. The "most secure" solution—the one that mitigates all perceived threats—is usually too costly to implement. In many cases, network operators and administrators with many years of experience have decided that SSHv1 is adequate for their needs and they can mitigate or minimize the exposure. In other cases, some operators are turning to SSHv2 or IPSec to cover the vulnerabilities that have been found in SSHv1. In some cases, the use of SNMPv1 may also be acceptable as long as its exposures are understood and the operators determine that its use will not pose a problem.

Excessive "security" may also intolerably reduce the usability of the network. It is important to remember that the network is there for the users. Placing security restrictions upon them to keep them out of the infrastructure is like keeping the doors locked to the building boiler room. Untrained people entering that area may hurt themselves or they may cause serious problems to others. If they have malicious intent, they could damage the machinery. Excessive security for that analogy would be similar to locking the boiler room, locking the ingress and egress points to the building, and mandating that armed guards accompany anyone that is permitted to enter the building. In some cases, that may be appropriate for the perceived threat. However, in the case that this applies to an elementary school building, it is inappropriate and would make some parents think of moving their children to other schools.

The model described in this article may be used as a thought process to review an entire network at a high layer to see the relationships between the various devices. It may also be used to design the security policy and the acceptable use policy of the network. Another use for it may be to define the operational procedures for the operators to securely administer the network and to define how the infrastructure devices will communicate. However it is used, some settlements must be made between the desire to provide security and the usefulness of the network. The cost of the security mechanisms cannot be unreasonably high, and the mechanisms cannot change the business model of the company. The enforcement of the policy must be effective, yet above all it must not change the expectations of the users. In all cases, the administrators and operators must find some balance between their need to secure the infrastructure and the need for the users to have the ability to actually use their network.

References

[1] Internet Engineering Task Force DNS Extensions Working Group, last updated July 2000,
http://www.ietf.org/html.charters/dnsext-charter.html

[2] Fraser, B., "Site Security Handbook," RFC 2196, September 1997.

[3] System Administration, Networking, and Security Institute,
http://www.sans.org/

[4] Fyodor **<fyodor@dhp.com>**, "NMAP—The Network Mapper,"
http://www.insecure.org/nmap/index.html

[5] Stallings, William, "Security Comes to SNMP: The New SNMPv3 Proposed Internet Standards," *The Internet Protocol Journal,* Vol. 1, No. 3, December 1998.

[6] Stallings, William, "SSL: Foundation for Web Security," *The Internet Protocol Journal,* Vol. 1, No. 1, June 1998.

[7] Stallings, William, "IP Security," *The Internet Protocol Journal,* Vol. 3, No. 1, March 2000.

[8] Avolio, Fred, "Firewalls and Internet Security," *The Internet Protocol Journal,* Vol. 2, No. 2, June 1999.

[9] CERT® Coordination Center, Tech Tips, "Protecting Yourself from Password File Attacks," Last revised February 12, 1999.

CHRIS LONVICK holds a Bachelor of Science degree from Christian Brothers College and is in the Consulting Engineering Department of Cisco Systems in Austin, Texas. He is currently the chair of the IETF Syslog Working Group. Chris can be reached at **clonvick@cisco.com**

Book Reviews

Multiwave Optical Networks

Multiwavelength Optical Networks: A Layered Approach, by Thomas E. Stern and Krishna Bala, ISBN 020130967X, Addison-Wesley, 1999.

Initial Impressions

This book attempts to fit into two camps; one, an overview of the potential choices that could be offered in wavelength-division multiplexing, or WDM, and the other, an academic text. Because of its scope, the treatment is uneven.

Organization

The first four chapters lay the groundwork. Chapter 1 starts by defining terms and positing why WDM is an enabling technology. The authors believe that the driving application will be LAN interconnection, ostensibly in metro areas. It is worthwhile noting that the authors make no claims about this text relating to an all-optical network. They simply expose the choices available to manipulate the various wavelengths, or lambda. The current methods for performing lambda manipulation are still bound in the electrical domain.

Chapter 2 covers the hierarchy or layering present in a WDM environment and some of the choices for configuration at each point in the hierarchy. The authors spend some time on the concepts of spectrum partitioning and what routing and switching in this domain means. A key point raised relates to the concept of wavelength conversion at network access points. The chapter closes with a brief review of some types of logical overlays that may sit on top of a WDM network. Three types are examined, ATM, *Synchronous Optical Network* (SONET), and IP networks.

The third chapter covers how network interconnection may occur and how the management and control features may be implemented. Four basic topologies are described, each with its salient features highlighted. These topologies include shared channel networks; wavelength routed networks, linear lightwave networks, and hybrid, logically routed networks. It is interesting to note that many commercial implementations, especially from traditional telecom providers, tend to follow the simpler topologies, while we are beginning to see newer telecom providers utilizing the more robust topologies.

Chapter 4 discusses what the authors consider enabling technology. To a large degree, these enabling technologies are the basic components of an optical system, for example, fibers, amplifiers, transmitters, and receivers. Crosstalk is mentioned in particular. The authors then delve into photonic device technologies and wavelength converters, and then they close with some simulation work on end-to-end transmission paths.

Chapters 5, 6, and 7 discuss in depth the ramifications of each of the four techniques. What is fairly intriguing here is that the authors have extensive bibliographies at the end of each chapter, and they include a series of problems that are left as an exercise to the reader.

The eighth chapter touches on the concepts involved with survivability and restoration of service. This chapter should help the practical network engineer in understanding most of the possible failure modes. In the last chapter, the authors look at current trends, and they try to predict business drivers for WDM deployment. Once again, they show their true colors as academics when they close with a statement on the importance of testbeds.

On to the Appendices! I am grateful to the authors for including some basic material on graph theory, scheduling algorithms, Markov chains and queuing, some work on minimal interference routing in the optical domain and, finally, close with a synopsis of the SONET standard.

Good Reference

Overall, there is a fair amount of practical material here, but it is tucked into large amounts of academic detail. I'm not sure this volume would work as a standalone textbook, but it clearly is a good reference for the state of optical networks in the last years of the 20th century.

—Bill Manning,
University of Southern California
Information Sciences Institute
manning@isi.edu

Net Slaves

Net Slaves: True Tales of Working the Web, Bill Lessard and Steve Baldwin, ISBN 0-07-135243-0, McGraw-Hill, 2000.

How can you not want to read a book that opens with a quote from a Guns&Roses song, "Do you know where you are? You're in the jungle, baby!"? *Net Slaves* is about the people who maintain the jungle that big game hunters come to exploit. The same jungle marketed as the digital age and the e-generation. This is the land of the "dot-coms" and future big-buck IPOs. Has hubris masked your role in this jungle? *Net Slaves* will set you straight. Exactly who are these net slaves? Well, take the 15 question quiz provided by the authors and determine your Internet exploitation quotient. Don't be shocked to find yourself among the new media caste; the only question is, what part of the jungle are you assigned to clean after?

The authors spent a year interviewing people who work for Internet-based companies. Based on their findings, they created 11 character composites: Garbagemen; Cops or Streetwalkers; Social Workers; Cab Drivers; Cowboys or Card Sharks; Fry Cooks; Gold Diggers or Gigolos; Priests or Madmen; Robots; Robber Barons; and Mole People.

For each composite the authors cite someone's real-life work experience—of course, in order to protect the innocent (and the guilty), names have been altered.

I was annoyed with David Zorn, Card Shark; his type does nothing but give the industry a bad reputation. The story of Ken Hussein, Robot, both saddened and angered me. I truly hope he and his family are doing better. How can anyone not feel sorry for Kellner after being taken in by Gigolo Mira? Jane, Cab Driver, learned the hard way that you have to roll with the blows to survive in the jungle. Finally, I must confess, I found the most disturbing of all profiles to be of Outis, a Mole Person.

For each profile the authors provide some social-economic statistics. How old is the average Social Worker? How much does it cost to hire a Cowboy? What are the career aspirations of the average Cab Driver? How do you know if a Robot is annoyed with you? You're a Garbageman; what are your chances of upward mobility? A lot of this is funny, but to leave it at that would be missing the point entirely. Every composite represents scores of real people's lives, and how they live doesn't necessarily match up with the glamour often associated with the high-tech industry.

My favorite profile is of Jason Barstow, a Madman. Barstow arrives on the scene on his Harley, ready to participate in a two-day seminar put on by the Earth Business Network. A former chicken farmer and former guitar player, Barstow now finds himself lecturing to a room full of CEOs. He begins by telling them about the 5 milligrams of LSD he bought the previous night, and proceeds to plant seeds of anxiety—did he spike their morning juice? As Barstow delivers his lecture on the future of e-commerce and builds to the climax, a frustrated Slim Clarkston of NetScathe blurts out, "Mr. Barstow, I want you to tell us the truth about your little prank." With the lecture over, Barstow returns his pass to the security desk. "How did it go?" asks the security guard. "Same bull," Barstow responds, "but they never seem to get tired of it."

Are these stories true? I don't know—it doesn't matter! What are true are the composites. This book is funny. It is also humbling. Most important, it is true. It was fun to read. After each chapter, I found myself wearing an undeniable mischievous grin as I scanned the office looking for the person I just read about; this is all in good fun as long as I remember one important thing: I'm in the book—and you are too. In my experiences, I've found that a certain animosity always exists between people who work call centers, programmers, Web designers, managers, and the like. *Net Slaves* reminds us that we are all in this jungle together.

—Neophytos Iacovou, eBenX Inc
diacovou@ebenx.com

Implementing IPSec

Implementing IPSec: Making Security work on VPNs, Internets, and Extranets, Elizabeth Kaufman and Andrew Newman, ISBN 0-471-34467-2 Wiley Computers Publishing, 1999.

Organization

The book is organized into four parts. The first three chapters of Part One should be nothing more than review for anyone who has been in networking for even a short time. Chapter 4, "Encrypting within the Law," analyzes current worldwide regulatory trends for encryption technologies and examines how existing laws will impact your ability to legally purchase and install IPSec products. Included is some good information that may help keep you on the right side of the laws pertaining to encryption. Encryption is an area of potential problems, especially when you are running your network between countries.

Part Two is a primer on the basic technological components of IPSec. Chapter 5, "A Functional Overview of IPv4," and its basic design characteristics should be old news to anyone who is seriously thinking of running any type of encryption on his/her network. Chapter Six is an overview of cryptographic technologies. Chapter 7 "The Basics of IPSec and Public Key Infrastructures (PKIs) Fundamental to Current IPSec Standards," has some good information pertaining to IPSec and its different components, but leaves out an explanation of its two basic modes of operation: *transport* and *tunnel.*

Part Three analyzes how and why the IPSec protocols can break existing IP networks, and should provide the reader with some good information. Chapter 8, "What Won't Work with IPSec," describes the root cause of IPsec performance problems and protocol conflicts. Chapter 9, "IPSec and PKI Rollout Considerations," discusses gateway-to-gateway, end host-to-gateway, and end host-to-end host configuration options and explains some of the policy elements of PKI.

Part Four provides some criteria for evaluating vendors and products; this information would be of little interest if you are unfamiliar with writing an RFI. Also included is some reference material, including an appendix, with a complete copy of the IPSec RFC (2401), "Security Architecture for the Internet Protocol." A glossary, which does *not* offer a description of IPSec, is included as well.

Who Should Read This Book

By trying to appeal to the technical as well as the nontechnical reader, the book has missed both. There are areas that will appeal to the reader with a limited networking background, as well as areas for the more technical. However, if you are the type of reader inclined to read the RFCs, you will find very little reason to read the remainder of the book. Overall the book does not provide enough information for any one group. Inclusion of RFC 2401 seems unnecessary considering how easily RFCs can be obtained from the Internet.

—Al Pruitt, CSG Systems, Inc
`al_pruitt@csgsystems.com`

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

Fragments

Scott Bradner Receives Postel Service Award

The Internet Society (ISOC) recently announced that noted Internet standards leader and Internet pioneer Scott O. Bradner has been awarded the prestigious *Jonathan B. Postel Service Award* for 2000. In presenting the award, Geoff Huston, Chair of ISOC, said, "Scott Bradner was introduced to many of us with his accurate and careful measurements of router performance. He has been a long standing participant in the *Internet Engineering Task Force* (IETF), and continues to serve on the *Internet Engineering Steering Group* (IESG) as the Area Director for Transport. He was a ISOC Trustee for six years from 1993 until 1999 and continues to serve as the Society's Vice-President for Standards. This is an impressive set of contributions and is worthy of recognition in Jon Postel's name as the 2000 recipient of the Jonathan B. Postel Service Award."

Don Heath, president and CEO of ISOC, said, "We established the award to honor the late Jon Postel by recognizing his unselfish and substantial contributions to the Internet over a 25 year period." He added, "Scott Bradner exemplifies the spirit of all that Jon brought to the Internet community and his outstanding contributions have made this year's choice an easy one. Scott's careful judgment and good humor has been a major contribution to many of the ISOC's activities, and we are pleased to be able to recognize his contributions in this unique fashion."

Bradner has been an active contributor to the IETF for over a decade, and has served as a Working Group Chair, the Area Director for Operations and currently serves as the Area Director for Transport. He also was the Director of the IPv6 area, and oversaw the process of refinement of a number of proposals into the definition of a coherent architecture for IPv6. Bradner has been the prime author of the current Internet Standards Process documents. He has also been an instructor at ISOC's *Network Training Workshops for Developing Countries* for many years, and has been a catalyst for the development of operationally robust Internet services in many areas of the world.

The Award is named for Dr. Jonathan B. Postel, an Internet pioneer and head of the organization that administered and assigned Internet names, protocol parameters, and Internet Protocol (IP) addresses. He was the primary architect behind what has become the *Internet Corporation for Assigned Names and Numbers* (ICANN), the successor organization to his work. The Award is presented at the Internet Society's annual INET Conference. It consists of an engraved crystal globe and US $20,000.00. Scott Bradner becomes the second recipient of the award. The first was presented posthumously to Dr. Postel in 1999.

The Internet Society is a non-profit, non-governmental, open membership organization whose worldwide individual and organization members make up a veritable "who's who" of the Internet industry. It provides leadership in technical and operational standards, policy issues, and education. ISOC hosts two annual Internet conferences, trains people from all over the world in networking technologies, conducts workshops for educators, and publishes an award-winning magazine, *OnTheInternet*. ISOC provides an international forum to address the most important economic, political, social, ethical and legal initiatives influencing the evolution of the Internet. This includes facilitating discussions on key policy decisions such as taxation, copyright protection, privacy and confidentiality, and initiatives towards self-governance of the Internet. ISOC created the Internet Societal Task Force as an ongoing forum for discussion, debate, and development of position papers, white papers, and statements on Internet related societal issues.

ISOC is the organizational home of the IETF, the Internet Architecture Board, the IESG, and the Internet Research Task Force—the standards setting and research arms of the Internet community. These organizations operate in an environment of bottom-up consensus building made possible through the participation of thousands of people from throughout the world. For more information, see **http://www.isoc.org/**

APNIC Policy Meeting

The *Asia Pacific Network Information Centre* (APNIC) will host an Open Policy Meeting October 25–27, 2000 in Brisbane, Australia. The meeting is open to anyone with an interest in Internet addressing issues. For more information see: **http://apnic.org**

APRICOT 2001

The *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will be held in Kuala Lumpur, Malaysia, February 26 to March 2, 2001. APRICOT is a forum that facilitates knowledge sharing among key Internet builders in the region, with peers and leaders from the Internet community worldwide. Since 1996, APRICOT has established itself as Asia Pacific's premier regional Internet Summit where related organisations converge and host their annual general meetings and other special events. The week-long summit comprises seminars, workshops, tutorials, conference sessions, Birds of a Feather (BOFs), and other forums, all geared towards spreading and sharing the knowledge required to operate the Internet within the Asia Pacific region. For more information see: **http://www.apricot2001.net**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

December 2000 Volume 3, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Numerous technologies have been developed to protect or isolate corporate networks from the Internet at large. These solutions incorporate security, either end-to-end (IP security, or IPSec), or at the Internet/intranet border (firewalls). A third class of systems allows a range of IP addresses to be used internally in a corporate network, while preserving IP address consumption through the use of a *single* public address. This latter class of device is called a *Network Address Translator* (NAT), and while many Internet engineers consider NATs to be "evil," they are nonetheless very popular. Combining IPSec, NATs, and firewalls can be quite challenging, however. In our first article Lisa Phifer explains the problem and offers some solutions.

Successful network design is the result of many factors. In addition to the basic building blocks of routers, switches and circuits, network planners must carefully consider how these elements are interconnected to form an overall system with as few single points of failure as possible. In our second article, Valdis Krebs looks at how lessons learned from social network analysis can be applied to the design of computer networks.

The current Internet grew out of several government-funded research efforts that began in the late 1960s. Today, basic technology development as well as research into new uses of computer networks continues in many research "testbeds" all over the world. Bob Aiken describes the past, present and future state of network research and research networks.

The online subscription system for this journal will be up and running in January at **www.cisco.com/ipj**. In addition to offering a subscription form, the system will allow you to select delivery options, update your mailing and e-mail address, and much more. Please visit our Web site and give it a try. If you encounter any difficulties, please send your comments to **ipj@cisco.com**.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

The Trouble with NAT

by Lisa Phifer, Core Competence

Those who are implementing virtual private networks often ask whether it is possible to safely combine *IP Security* (IPSec) and *Network Address Translation* (NAT). Unfortunately, this is not a question with a simple "yes" or "no" answer. IPSec and NAT can be employed together in some configurations, but not in others. This article explores the issues and limitations associated with combing NAT and "NAT-sensitive" protocols like IPSec. It examines configurations that do not work, and explains why. It illustrates methods for using NAT and IPSec together, and discusses an emerging protocol that may someday prove more IPSec friendly.

This article builds upon "IP Security and NAT: Oil and Water?"[1] and "Realm-Specific IP for VPNs and Beyond"[2], works previously published by *ISP-Planet*.

What Is Network Address Translation?

NAT was originally developed as an interim solution to combat IPv4 address depletion by allowing globally registered IP addresses to be reused or shared by several hosts. The "classic" NAT defined by RFC 1631[3] maps IP addresses from one realm to another. Although it can be used to translate between any two address realms, NAT is most often used to map IPs from the nonroutable private address spaces defined by RFC 1918[4], shown below.

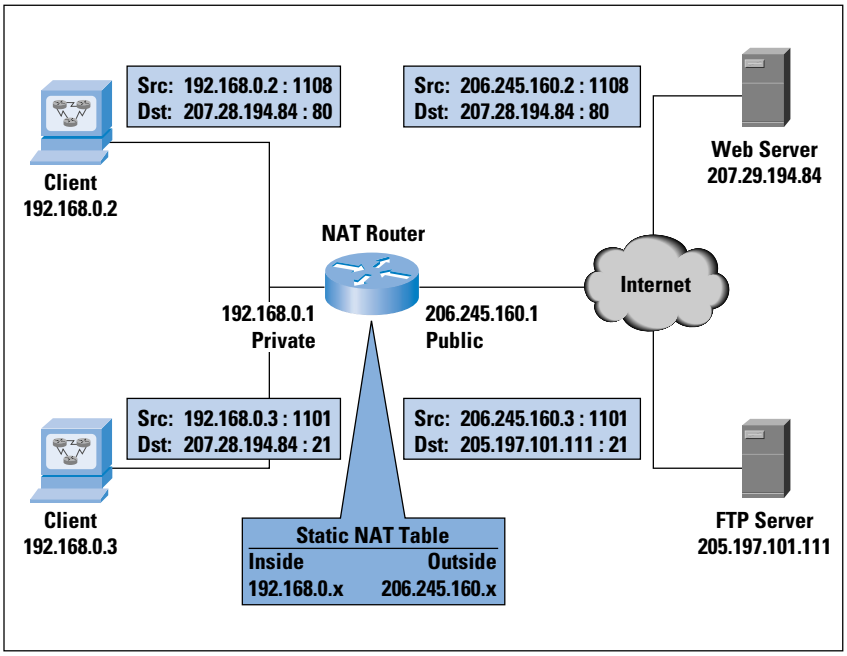
Class	Private Address Range
A	10.0.0.0 ... 10.255.255.255
B	172.16.0.0 ... 172.16.255.255
C	192.168.0.0 ... 192.168.255.255

These addresses were allocated for use by private networks that either do not require external access or require limited access to outside services. Enterprises can freely use these addresses to avoid obtaining registered public addresses. But, because private addresses can be used by many, individually within their own realm, they are nonroutable over a common infrastructure. When communication between a privately addressed host and a public network (like the Internet) is needed, address translation is required. This is where NAT comes in.

NAT routers (or NATificators) sit on the border between private and public networks, converting private addresses in each IP packet into legally registered public ones. They also provide transparent packet forwarding between addressing realms. The packet sender and receiver (should) remain unaware that NAT is taking place. Today, NAT is commonly supported by WAN access routers and firewalls—devices situated at the network edge.

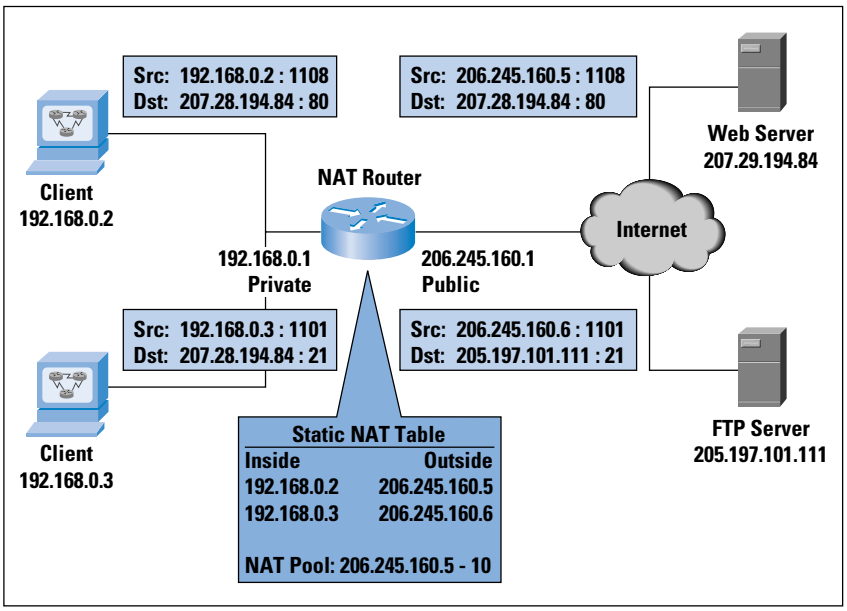
NAT works by creating bindings between addresses. In the simplest case, a one-to-one mapping may be defined between public and private addresses. Known as static NAT, this can be accomplished by a straightforward, stateless implementation that transforms only the network part of the address, leaving the host part intact. The payload of the packet must also be considered during the translation process. The IP checksum must, of course, be recalculated. Because TCP checksums are computed from a pseudo-header containing source and destination IP address (prepended to the TCP payload), NAT must also regenerate the TCP checksum.

Figure 1: Static NAT



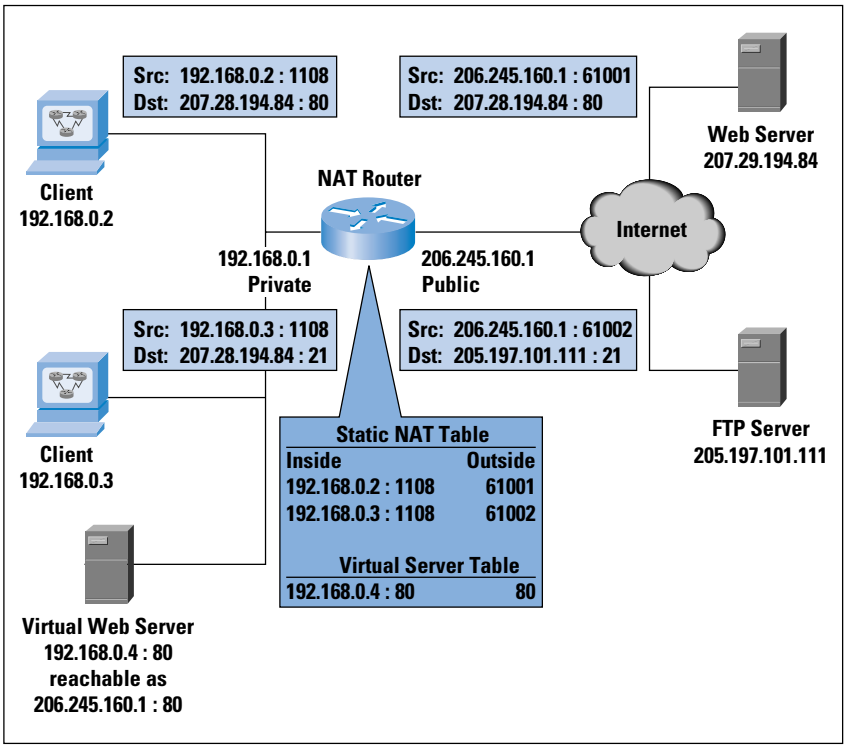
More often, a pool of public IP addresses is shared by an entire private IP subnet (dynamic NAT). Edge devices that run dynamic NAT create bindings "on the fly," building a NAT Table. Connections initiated by private hosts are assigned a public address from a pool. As long as the private host has an outgoing connection, it can be reached by incoming packets sent to this public address. After the connection is terminated (or a timeout is reached), the binding expires, and the address is returned to the pool for reuse. Dynamic NAT is more complex because state must be maintained, and connections must be rejected when the pool is exhausted. But, unlike static NAT, dynamic NAT enables address reuse, reducing the demand for legally registered public addresses.

Figure 2: Dynamic NAT



A variation of dynamic NAT known as *Network Address Port Translation* (NAPT) may be used to allow many hosts to share a single IP address by multiplexing streams differentiated by TCP/UDP port number. For example, suppose private hosts 192.168.0.2 and 192.168.0.3 both send packets from source port 1108. A NAPT router might translate these to a single public IP address 206.245.160.1 and two different source ports, say 61001 and 61002. Response traffic received for port 61001 is routed back to 192.168.0.2:1108, while port 61002 traffic is routed back to 192.168.0.3:1108.

Figure 3: NAPT



NAPT (masquerading) is commonly implemented on small Office/Home Office (SOHO) routers to enable shared Internet access for an entire LAN through a single public address. Because NAPT maps individual ports, it is not possible to "reverse map" incoming connections for other ports unless another table is configured. A virtual server table can make a server on a privately addressed DMZ reachable from the Internet via the public address of the NAPT router (one server per port). This is really a limited form of static NAT, applied to incoming requests.

In some cases, static NAT, dynamic NAT, NAPT, and even bidirectional NAT or NAPT may be used together. For example, an enterprise may locate public Web servers outside of the firewall, on a DMZ, while placing a mail server and clients on the private inside network, behind a NAT-ing firewall. Furthermore, suppose there are applications within the private network that periodically connect to the Internet for long periods of time. In this case:

- Web servers can be reached from the Internet without NAT, because they live in public address space.
- *Simple Mail Transfer Protocol* (SMTP) sent to the private mail server from the Internet requires incoming translation. Because this server must be continuously accessible through a public address associated with its *Domain Name System* (DNS) entry, the mail server requires static mapping (either a limited-purpose virtual server table or static NAT).
- For most clients, public address sharing is usually practical through dynamically acquired addresses (either dynamic NAT with a correctly sized address pool, or NAPT).
- Applications that hold onto dynamically acquired addresses for long periods could exhaust a dynamic NAT address pool and block access by other clients. To prevent this, long-running applications may use NAPT because it enables higher concurrency (thousands of port mappings per IP address).

Where is NAT used today? Outbound NAT is commonly employed by multihost residential users, teleworkers, and small businesses that share a single public IP for outbound traffic while blocking inbound session requests. In other words, small LANs connected via ISDN, *Digital Subscriber Line* (DSL), or cable modem.

Bidirectional static NAT/NAPT combinations are typically used by enterprises that host services behind a masquerading firewall. NAT can also be employed by enterprises wishing to insulate themselves from *Internet Service Provider* (ISP) address changes, or by those wanting to obscure private network topology for security reasons.

NAT-Sensitive Protocols

Our need to conserve IPv4 addresses has prompted many to overlook the inherent limitations of NAT, recognized in RFC 1631 but deemed acceptable for a short-term solution.

As noted previously, NAT regenerates TCP checksums. This, of course, requires the TCP header containing the checksum to be visible (that is, not encrypted). If only the TCP payload is encrypted and immutable between the application source and destination (for instance, *Secure Shell Protocol* [SSH], *Secure Sockets Layer* [SSL]), then the checksum in the TCP header can be recalculated without a visible TCP payload. But if the TCP header is encrypted (for instance, IPSec transport mode), the TCP checksum field in the TCP header cannot be modified.

Furthermore, many application protocols carry IP addresses in an application-level protocol. In such cases, an *Application-Level Gateway* (ALG) is needed to complete the translation. For example:

- Many *Internet Control Message Protocol* (ICMP) packets (for instance, "Destination Unreachable") carry embedded IP packets in ICMP payload. These require both address translation and checksum regeneration.
- A *File Transfer Protocol* (FTP) ALG is needed to rewrite IP addresses carried by FTP PORT and PASV control commands. In the IP header, these addresses are fixed-length words. Unfortunately, in the FTP protocol, these IP addresses are carried as human-readable, variable-length strings; rewriting can change the length of the TCP segment. If the segment is shortened, it can be padded. If the segment is lengthened, SEQ and ACK numbers must be transformed for the duration of the connection.
- Protocols like H.323 use multiple TCP connections or UDP streams to form "session bundles." If all connections in the bundle originate from the same end system, an ALG may be avoided. But H.323 presents other challenges, including ephemeral ports and embedded, ASN.1-encoded IP addresses in application payload.
- *NetBIOS over TCP/IP* (NBT) can be challenging to translate correctly because packet-header information is placed in NetBIOS payload at inconsistent offsets, and many embedded IP addresses are exchanged during an NBT session. Fortunately, most companies do not let NBT beyond their firewall anyway.
- *Simple Network Management Protocol* (SNMP) packets also carry IP addresses that identify trap source and object instance. Perhaps more important, dynamic NAT makes it impossible to uniquely identify hosts by IP address; public addresses are transient and shared. Remote management of private hosts can thus be impeded by NAT.
- Obviously DNS, responsible for domain name/IP address mapping, is impacted by NAT. From simple query handling to zone transfers, a robust DNS ALG is defined by RFC 2694[9].

NAT-sensitive protocols such as Kerberos, X-Windows, remote shell, Session Initiation Protocol (SIP), and others are further described in the Internet Draft *"Protocol Complications with the IP Network Address Translation"*[12]. Another Internet Draft, *"NAT Friendly Application Design Guidelines"*[13], explains how new application protocols can integrate smoothly with NAT. But there are still cases where ALGs simply cannot "fix" packets modified by NAT.

Impact of NAT on IPSec

The IPSec *Authentication Header* (AH)[5] is an example. AH runs the entire IP packet, including invariant header fields such as source and destination IP address, through a message digest algorithm to produce a keyed hash. This hash is used by the recipient to authenticate the packet. If any field in the original IP packet is modified, authentication will fail and the recipient will discard the packet. AH is intended to prevent unauthorized modification, source spoofing, and man-in-the-middle attacks. But NAT, by definition, modifies IP packets. Therefore, AH + NAT simply cannot work.

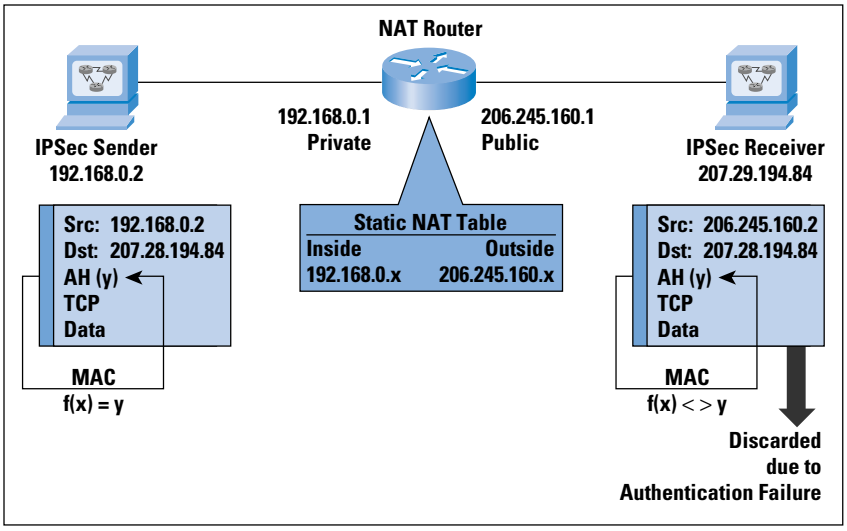


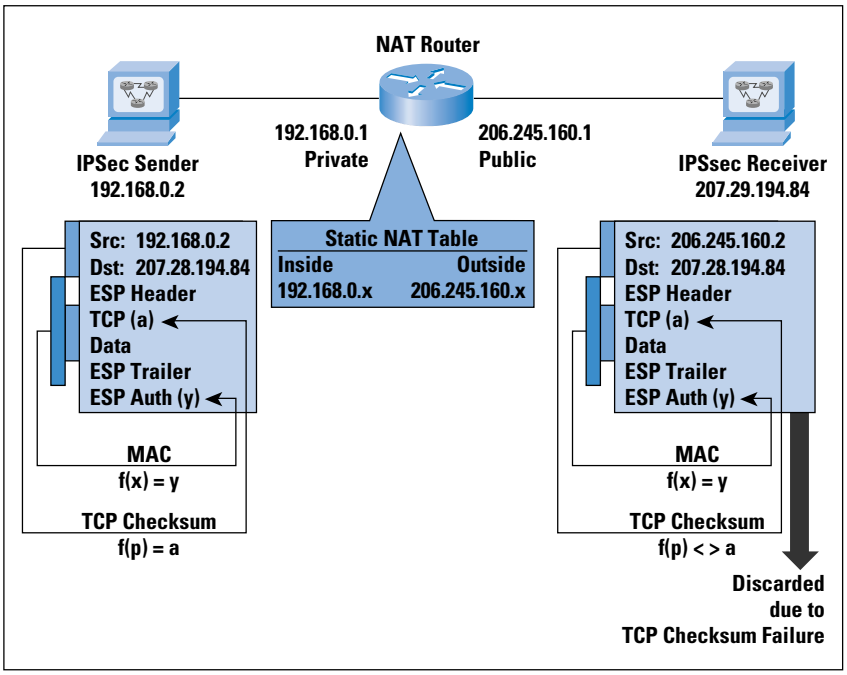
Figure 4: NAT vs. AH (Transport Mode)

The IPSec *Encapsulating Security Payload* (ESP)[6] also employs a message digest algorithm for packet authentication. But, unlike AH, the hash created by ESP does not include the outer packet header fields. This solves one problem, but leaves others.

IPSec supports two "modes." Transport mode provides end-to-end security between hosts, while tunnel mode protects encapsulated IP packets between security gateways—for example, between two firewalls or between a roaming host and a remote access server. When TCP or UDP are involved—as they are in transport mode ESP—there is a catch-22. Because NAT modifies the TCP packet, NAT must also recalculate the checksum used to verify integrity. If NAT updates the TCP checksum, ESP authentication will fail. If NAT does not update the checksum (for example, payload encrypted), TCP verification will fail.

If the transport endpoint is under your control, you might be able to turn off checksum verification. In other words, ESP can pass through NAT in tunnel mode, or in transport mode with TCP checksums disabled or ignored by the receiver.

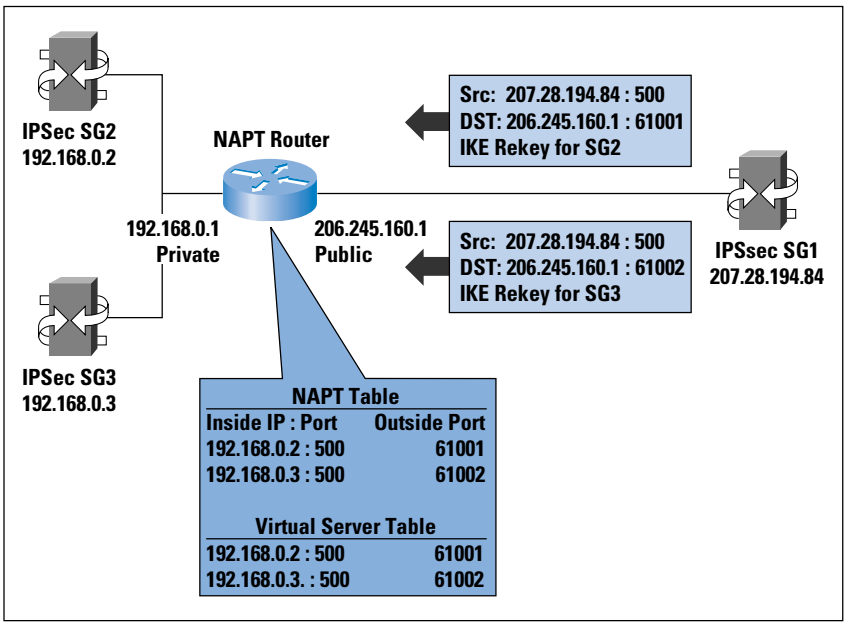
Figure 5: NAT vs. ESP (Transport Mode)



If we stick to ESP in tunnel mode or turn off checksums, there's still another obstacle: the *Internet Key Exchange* (IKE)[7]. IPSec-based *Virtual Private Networks* (VPNs) use IKE to automate security association setup and authenticate endpoints. The most basic and common method of authentication in use today is preshared key. Unfortunately, this method depends upon the source IP address of the packet. If NAT is inserted between endpoints, the outer source IP address will be translated into the address of the NAT router, and no longer identify the originating security gateway. To avoid this problem, it is possible to use another IKE "main mode" and "quick mode" identifier (for example, user ID or fully qualified domain name).

A further problem may occur after a *Security Association* (SA) has been up for awhile. When the SA expires, one security gateway will send a rekey request to the other. If the SA was initiated from the well-known IKE port UDP/500, that port is used as the destination for the rekey request. If more than one security gateway lies behind a NAPT router, how can the incoming rekey be directed to the right private IP address? Rekeys can be made to work by "floating" the IKE port so that each gateway is addressable through a unique port number, allowing incoming requests to be demultiplexed by the NAPT router.

Figure 6: NAT vs. IKE Rekey



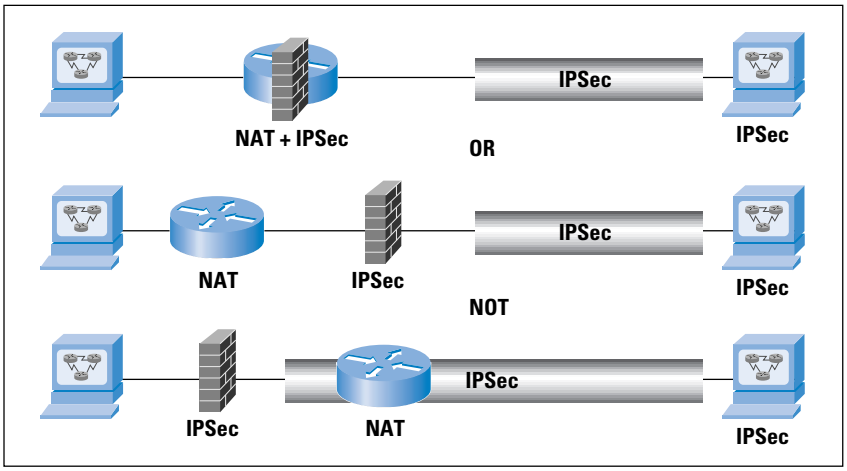
At this point, two things should be clear: (1) it is possible to find a "flavor" of IPSec that will run through NAT, but (2) one must do so with great care and attention to detail. Recent Internet Drafts[12] [14] have recorded these problems for further consideration, and RFC 2709[10] describes a security model for running tunnel-mode IPSec through NAT.

One Solution: Avoid the Problem

By far the easiest way to combine IPSec and NAT is to completely avoid these problems by locating IPSec endpoints in public address space. That is, NAT before IPSec; don't perform IPSec before NAT. This can be accomplished in two ways:

- Perform NAT on a device located behind your IPSec security gateway; or
- Use an IPSec device that also performs NAT.

Figure 7: Combining IPSec and NAT



Many routers, firewalls, security gateways, and Internet appliances implement IPSec and NAT in the same box. These products perform outbound address translation before applying security policies; the order is reversed for inbound packets. A typical "any-to-any" security policy is easily specified with such a product. Granular policies can be a bit more difficult because filters are often based on IP address, and care must be taken to avoid overlapping filters.

If you cannot avoid translating IPSec-protected traffic midstream, limit use of IPSec to tunnel-mode ESP and design security policies with care. If you simply cannot NAT before IPSec or require transport-mode ESP, there may still be hope. The *Internet Engineering Task Force* (IETF) is now defining *Realm-Specific IP* (RSIP), an alternative that may someday prove kinder to IPSec.

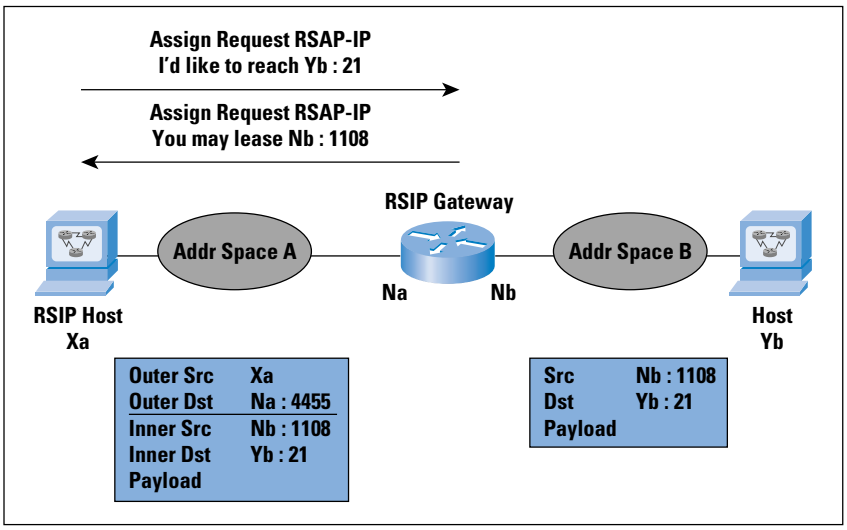
What Is RSIP?

RSIP[16] leases public IP addresses and ports to RSIP hosts located in private addressing realms. Unlike NAT, RSIP does not operate in stealth mode and does not translate addresses on the fly. Instead, RSIP allows hosts to directly participate concurrently in several addressing realms. Although RSIP does require host awareness, it avoids violating the end-to-end nature of the Internet. With RSIP, IP payload flows from source to destination without modifications that cripple IPSec AH and many other NAT-sensitive protocols.

RSIP gateways are multihomed devices that straddle two or more addressing realms, just as NAT-capable firewalls and routers do today. When an RSIP-savvy host wants to communicate beyond its own private network, it registers with an RSIP gateway. The RSIP gateway allocates a unique public IP address (or a shared public IP address and a unique set of TCP/UDP ports) and binds the private address of the RSIP host to this public address. The RSIP host uses this public source address to send packets to public destinations until its lease expires or is renewed.

But the RSIP host cannot send a publicly addressed packet as-is; it must first get the packet to the RSIP gateway. To do this, the host wraps the original packet inside a privately addressed outer packet. This "encapsulation" can be accomplished using any standard tunneling protocol: IP-in-IP, the *Generic Routing Encapsulation* (GRE), or the *Layer 2 Tunneling Protocol* (L2TP). Upon receipt, the RSIP gateway strips off the outer packet and forwards the original packet across the public network, toward its final destination.

Figure 8: RSIP



For simplicity, we talk about RSIP linking one private network to the public Internet, but RSIP can also be used to relay traffic between several privately addressed networks. An RSIP host can lease several different addresses as needed to reach different destinations networks. We've also focused on outgoing traffic, but an RSIP host can ask the RSIP gateway to "listen" and relay incoming packets addressed to a public IP and port.

Combining RSIP and IPSec

At first glance, RSIP sounds like a promising way for hosts to share public addresses while avoiding the pitfalls associated with applying NAT to IPSec traffic. But it turns out that RSIP extensions are needed to accommodate end-to-end IPSec[17].

Basic RSIP relies on unique port numbers to demultiplex arriving packets, but IPSec ESP encrypts port numbers. When several RSIP hosts use the same RSIP gateway to relay ESP, another discriminator is needed. Fortunately, every IPSec packet carries a unique *Security Parameters Index* (SPI), assigned during security association setup. Unfortunately, the SPI is guaranteed unique only for the responder. To enable demultiplexing, the tuple (SPI, protocol [AH or ESP], destination IP address) must also be unique at the initiating RSIP gateway.

A similar problem occurs during association setup with the IKE. IKE packets usually carry the well-known source port UDP/500. Using different source ports is the preferred solution, but if several RSIP hosts use the same RSIP gateway to relay IKE from port UDP/500, another discriminator is needed. Again, there is a convenient answer: every IKE packet carries the initiator cookie supplied in the first packet of an IKE session. The RSIP gateway can route IKE responses to the correct RSIP host using the tuple (initiator cookie, destination port [IKE], destination IP address). But rekeys may still be an issue.

To fix these problems, extensions have been proposed to allow RSIP hosts to register with an RSIP gateway for IPSec support, and allow hosts to request and receive unique SPI values along with leased IP addresses and ports.

Possible Applications for RSIP

RSIP specifications[16][17][18] are still at the Internet Draft stage. If and when RSIP matures, there may be a wide variety of applications:

- Residential power users and teleworkers with multihost LANs that share a single, publicly known IP address leased by an RSIP-enabled Internet appliance, DSL router, or cable modem;
- Small-to-midsize enterprise customers with dozens or hundreds of hosts, sharing a small pool of public IPs leased by an RSIP-enabled WAN access router or firewall;
- Multidwelling units (apartments, shared office buildings) with many private LANs, sharing public Internet access through an RSIP-enabled device;
- Hospitality networks (airports, hotels) where roaming hosts briefly lease the public IP(s) shared by the entire network;
- Remote access concentrators that use RSIP to lease private IP(s) to roaming corporate users that access the Internet via dynamically assigned public addresses; and
- Wireless devices (cell phones, personal digital assistants [PDAs]) that lease public IP(s) for "sticky sessions" that persist even when the mobile device moves from one location to another, updating its local access IP.

These scenarios, and the relationship of RSIP to IP multicast and differentiated services, are more fully explored in the RSIP framework[18].

Conclusion

Although NAT can be combined with IPSec and other NAT-sensitive protocols in certain scenarios, NAT tampers with end-to-end message integrity. RSIP—or whatever RSIP evolves into—may someday prove to be a better address-sharing solution for protocols that are adversely impacted by NAT. If RSIP fails to mature, another solution may be developed to broaden use of NAT with IPSec. Alternatives now under discussion within the IETF include UDP encapsulation and changes to IKE itself[14][15].

Despite its origin as a short-term solution, NAT is unlikely to disappear in the very near future. Until it does, understanding the relationship between NAT and IPSec and alternatives for safe combined deployment will remain an important aspect of VPN design.

References

[1] Phifer, L., "IP Security and NAT: Oil and Water?" *ISP-Planet,* June 15, 2000.
http://www.isp-planet.com/technology/nat_ipsec.html

[2] Phifer, L., "Realm-Specific IP for VPNs and Beyond," *ISP-Planet,* June 23, 2000.
http://www.isp-planet.com/technology/rsip.html

[3] Egevang, K. and Francis, P., "The IP Network Address Translator (NAT)," RFC 1631, May 1994.

[4] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G.J., and Lear, E., "Address Allocation for Private Internets," RFC 1918, February 1996.

[5] Kent, S. and Atkinson, R., "IP Authentication Header," RFC 2402, November 1998.

[6] Kent, S. and Atkinson, R., "IP Encapsulating Security Payload (ESP)," RFC 2406, November 1998.

[7] Harkins, D. and Carrel, D., "The Internet Key Exchange (IKE)," RFC 2409, November 1998.

[8] Srisuresh, P. and Holdrege, M., "IP Network Address Translator (NAT) Terminology and Considerations," RFC 2663, August 1999.

[9] Srisuresh, P., Tsirtsis, G., Akkiraju, P. and Heffernan, A., "DNS Extensions to Network Address Translators (DNS_ALG)," RFC 2694, September 1999.

[10] Srisuresh, P., "Security Model with Tunnel-Mode IPSec for NAT Domains," RFC 2709, October 1999.

[11] Tsirtsis, G. and Srisuresh, P., "Network Address Translation-Protocol Translation (NAT-PT)," RFC 2766, February 2000.

[12] Srisuresh, P. and Holdrege, M., "Protocol Complications with the IP Network Address Translator," Internet Draft, Work in Progress, July 2000.

[13] Senie, D., "NAT Friendly Application Design Guidelines," Internet Draft, Work in Progress, July 2000.

[14] Aboba, B., "NAT and IPSec," Internet Draft, Work in Progress, July 2000.

[15] Stenberg, M., Paavolainan, S., Ylonen, T., and Kivinen, T., "IPSec NAT-Traversal," Internet Draft, Work in Progress, July 2000.

[16] Borella, M. and Lo, J., "Realm-Specific IP: Protocol Specification," Internet Draft, Work in Progress, March 2000.

[17] Montenegro, G. and Borella, M., "RSIP Support for End-to-End IPSec," Internet Draft, Work in Progress, March 2000.

[18] Borella, M., Lo, J., Grabelsky, D., and Montenegro, G., "Realm-Specific IP: Framework," Internet Draft, Work in Progress, March 2000.

LISA PHIFER is vice president of Core Competence, Inc. **(www.corecom.com)**, a consulting firm specializing in Internet, network management, and security technologies. She earned her Master's Degree in Computer Science from Villanova University. A Bellcore award recipient for her work in ATM network operations, Lisa has been involved in the design and deployment of networking protocols for over 18 years. She represented Bellcore and Unisys in several industry-standards organizations, and has participated in The Internet Security Conference (TISC) since its inception. Lisa consults, teaches, and writes about a variety of technologies, including caching, load balancing, DSL, ISDN, IPSec, PKI, OSS, and VPNs. Her monthly column on virtual private networking is published by *ISP-Planet*. E-mail: **lisa@corecom.com**

The Social Life of Routers

Applying Knowledge of Human Networks to the Design of Computer Networks

by Valdis Krebs

We often forget that computer networks are put in place to support human networks—person-to-person exchanges of information, knowledge, ideas, opinions, insights, and advice. This article looks at a technology that was developed to map and measure human networks—social network analysis—and applies some of its principles and algorithms to designing computer networks. And as we see more peer-to-peer (P2P) models of computer-based networks, the P2P metrics in human network analysis become even more applicable.

Social network analysts look at complex human systems as an interconnected system of nodes (people and groups) and ties (relationships and flows)—much like an internetwork of routers and links. Human networks are often unplanned, emergent systems. Their growth is sporadic and self-organizing[1]. Network ties end up being unevenly distributed, with some areas of the network having a high density of links and other areas of the network sparsely connected. These are called "small world networks"[2]. Computer networks often end up with similar patterns of connections—dense interconnectivity within subnetworks, and sparser connections uniting subnetworks into a larger internetwork.

Social network researchers and consultants focus on *geodesics*—shortest paths in the network. Many of today's social network algorithms are based on a branch of mathematics called *graph theory*. Social network scientists have concentrated their work, and therefore their algorithms, in the following areas:

- Individual node centrality within a larger network—network dependency and load upon individual routers
- Overall path distribution—good connectivity without excessive routing tables
- Improving communication flow within and between groups—designing better topologies
- Network patterns surrounding ego networks—strategies for analyzing and manipulating individual router connections
- Analyzing information flow behavior of client organization—how computer networks can support human networks

One of the methods used to understand networks and their participants is to evaluate the location of actors in the network. Measuring the network location is finding the *centrality* of a node[3]. All network measures discussed here are based on geodesics—the shortest path between any two nodes. We will look at a social network, called the *kite network*, that effectively shows the distinction between the three most popular centrality measures—the ABCs—*A*ctivity, *B*etweenness, and Closeness.

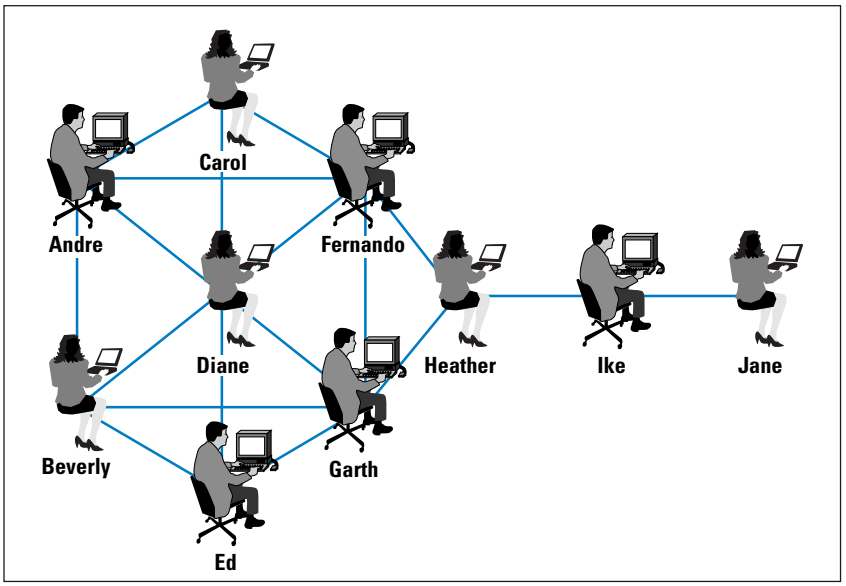
This model[4] was first developed by David Krackhardt, a leading researcher in social networks.

Activity

Figure 1 shows a simple social network. A link between a pair of nodes depicts a bidirectional information flow or knowledge exchange between two individuals. Social network researchers measure network activity for a node by using the concept of *degrees*—the number of direct connections a node has.

In this human network, Diane has the most direct connections in the network, making hers the most active node in the network with the highest degree count. Common wisdom in personal networks is "the more connections, the better." This is not always so. What really matters is where those connections *lead to*—and how they connect the otherwise unconnected![5] Here Diane has connections only to others in her immediate cluster—her clique. She connects only those who are already connected to each other—does she have too many redundant links?

Figure 1: Human Network



Betweenness

While Diane has many direct ties, Heather has few direct connections—fewer than the average in the network. Yet, in may ways, she has one of the best locations in the network—she is a boundary spanner and plays the role of broker. She is *between* two important constituencies, in a role similar to that of a border router. The good news is that she plays a powerful role in the network, the bad news is that she is a single point of failure. Without her, Ike and Jane would be cut off from information and knowledge in Diane's cluster.

Closeness

Fernando and Garth have fewer connections than Diane, yet the pattern of their ties allow them to *access* all the nodes in the network more quickly than anyone else. They have the shortest paths to all others—they are *close* to everyone else. Maximizing closeness between *all* routers improves updating and minimizes hop counts. Maximizing the closeness of only one or a few routers leads to counterproductive results, as we will examine below.

Their position demonstrates that when it comes to network connections, quality beats out quantity. Location, location, location—the golden rule of real estate also works in networks. In real estate it is geography—your physical neighborhood. In networks, it is your virtual location determined by your network connections—your network neighborhood.

Network Centralization

Individual network centralities provide insight into the individual's location in the network. The relationship between the centralities of all nodes can reveal much about the overall network structure. A very centralized network is dominated by one or a few very central nodes. If these nodes are removed or damaged, the network quickly fragments into unconnected subnetworks. Highly central nodes can become critical points of failure. A network with a low centralization score is not dominated by one or a few nodes—such a network has no single points of failure. It is resilient in the face of many local failures. Many nodes or links can fail while allowing the remaining nodes to still reach each other over new paths.

Average Path Length in Network

The shorter the path, the fewer hops/steps it takes to go from one node to another. In human networks, short paths imply quicker communication with less distortion. In computer networks, the signal degradation and delay is usually not an issue. Nonetheless, a network with many short paths connecting all nodes will be more efficient in passing data and reconfiguring after a topology change.

Average Path Length is strongly correlated with Closeness throughout the network. As the closeness of all nodes to each other improves (average closeness), the average path length in the network also improves.

Internetwork Topology

In the recent network design book, *Advanced IP Network Design*[6], the authors define a well-designed topology as the basis of a well-behaved and stable network. They further propose that "three competing goals must be balanced for good network design":

- Reducing hop count
- Reducing available paths
- Increasing the number of failures the network can withstand

Our social network algorithms can assist in measuring and meeting all three goals.

- Reducing the hop count infers minimizing the average path length throughout the network—maximize the closeness of all nodes to each other.
- Reducing the available paths leads to minimizing the number of geodesics throughout the network.
- Increasing the number of failures a network can withstand focuses on minimizing the centralization of the whole network.

On the following pages we examine various network topologies and evaluate them using social network measures while remembering these three competing goals of network design.

The models we examine do *not* cover hierarchical structures—with Core, Distribution, and Access layers—found in networks of hundreds or thousands of routers. We examine flat, nonhierarchical topologies such as those found in smaller internetworks, area subnetworks, or within core backbones. The topologies we model are the most commonly used—Star, Ring, Full Mesh, and Partial Mesh. We compute the social network measures on each of the topologies and discuss how the various measures help us meet the competing goals discussed above.

Star Topology

The Star topology, shown in Figure 2, has many advantages—but one glaring fault. The advantages include ease of management and configuration for the network administrators. For the Star, the three competing goals delineate as follows:

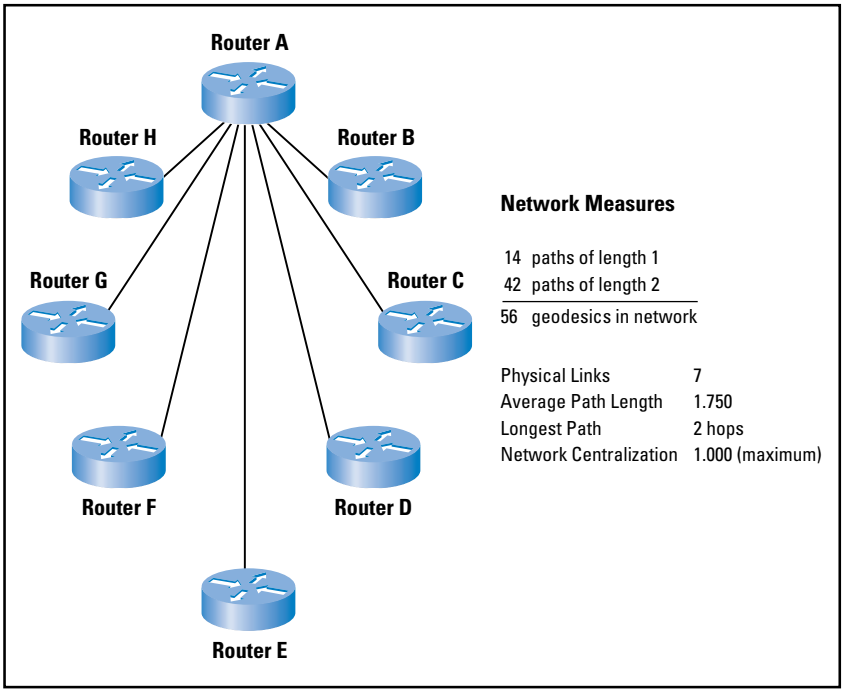
- *Reducing hop count:* The short average path length (1.75) throughout the network meets this goal well. Any router can *reach* any other router in two steps or less.
- *Reducing available paths:* The fact that there are a minimum number of possible available paths (56) to reach all other nodes—will not overload the routing tables, nor cause delays during routing table updates. It takes only seven bidirectional links to create the available paths.

- *Reducing network failures:* The network fails miserably if Router A goes down. Also, any link failure isolates the attached router—there are no multiple paths to reach each router.

Router A is not only a single point of failure—it is also a potential bottleneck—it will likely become overburdened with packet flows and routing updates as more routers are added in the star structure.

Router A receives the top score (1.000) in Activity, Betweenness, and Closeness. As a result, the network is very centralized around Router A from the perspective of all measures.

Figure 2: Routers in Star Topology



Ring Topology

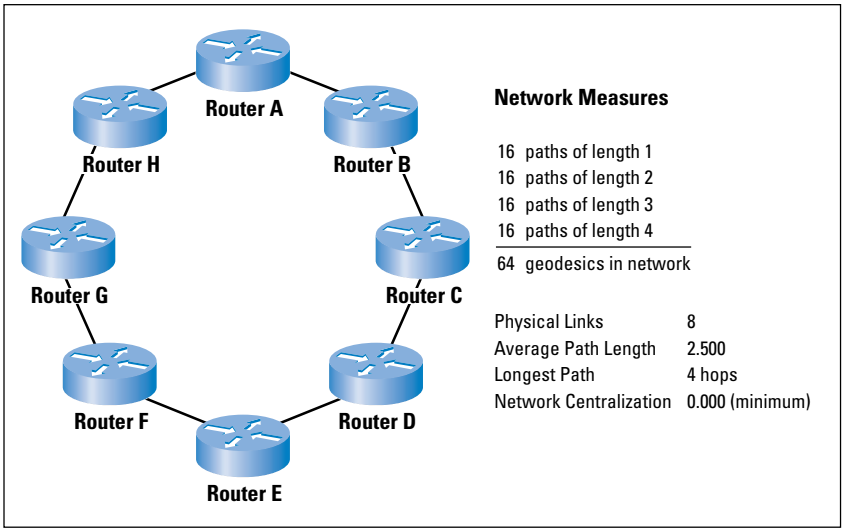
The Ring topology, shown in Figure 3, is an improvement over the Star. It has some of the same advantages, but does not eliminate all of the drawbacks of the Star. The advantages include ease of management and configuration for the network administrators—adding another router is very simple. Unlike the Star topology, the Ring provides some redundancy and, therefore, eliminates the single point of failure—all nodes have an alternate path through which they can be reached. Yet it is still vulnerable to both link and router failures. For the Ring, the three competing goals delineate as follows:

- *Reducing hop count:* The average path length of 2.5 is quite long for a small network of eight nodes. Some routers (that is, A and E) require four steps to reach each other! Many ring physical layers hide this complexity from the IP layers in order to make those hops invisible to routing protocols.

- *Reducing available paths:* This configuration has more geodesics (64) than Star, yet not significantly more to overload the routing tables, nor cause delays during table updates.
- *Reducing network failures:* Even though network centralization is at the minimum (no node is more central than any other), this network reaches failure quickly because of its weak redundancy. The Ring topology can withstand one link failure or one router failure and still keep a contiguous network. Two simultaneous failures can cause unreachable segments because of the lack of redundancy.

Most modern ring technologies such as *Synchronous Optical Network* (SONET) or the Cisco *Dynamic Packet Transport Protocol* (DPT) add a measure of redundancy by running a dual ring that heals itself if a link gets cut. The network "wraps" to avoid the downed line and operates at lower speed. A two-hop path can become a six-hop path if a single link fails. This can cause network congestion if the original dual ring was being used for data in all directions.

Figure 3: Routers in Ring Topology



Full Mesh Topology

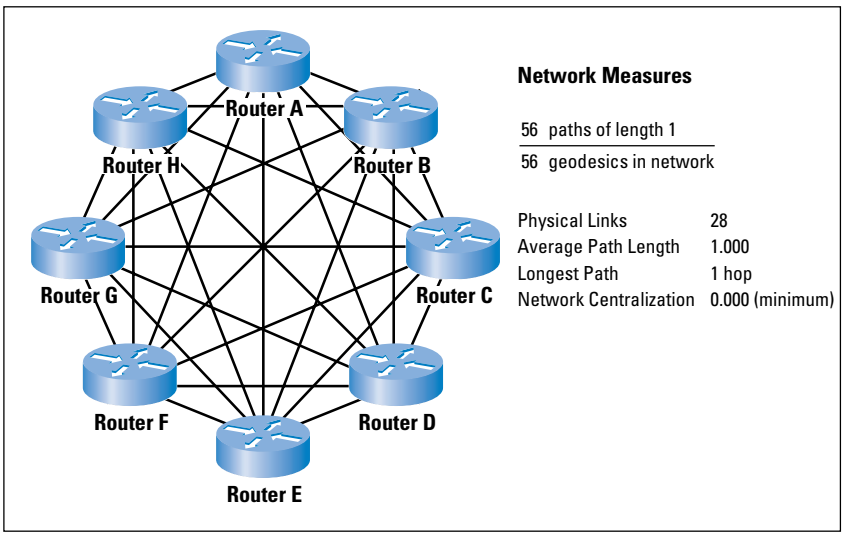
The Full Mesh topology has several big advantages and several faults. The advantages include short path length (one hop) to all other routers and maximum resilience to failure if links or routers start failing. The disadvantages revolve around the complexity created by this topology. For the Full Mesh, the three competing goals delineate as follows:

- *Reducing hop count:* The shortest path length possible is attained for all routes—all nodes can reach each other in one hop.
- *Reducing available paths:* There are a minimum number of possible available paths (56) to reach all other nodes. The routing entries will not overload the routing tables, nor cause delays during routing table updates.

- *Reducing network failures:* The network is not dependent upon any single node (network centralization = 0.000). This configuration represents the most robust topology available—chances are very slim that the number of failures necessary to fragment the network will actually occur within the same time period.

The disadvantages of the Full Mesh topology all focus on one glaring fault—there are too many physical links. If the routers are far apart, the link costs can quickly become prohibitively expensive because adding routers creates a geometrical explosion in links required—soon the routers do not have enough ports to support this topology. Administering the system and keeping an up-to-date topology map becomes more and more complex as routers are added. The network in Figure 4 has 28 two-way links. Double the routers, in a full mesh topology, and the link count increases by a factor greater than 4.

Figure 4: Routers in Full Mesh Topology



Partial Mesh Topology

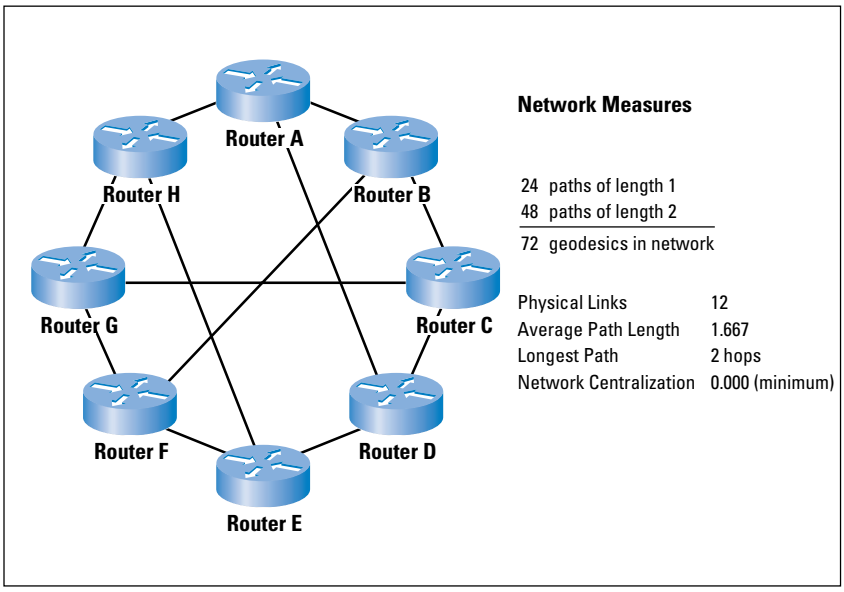
The Partial Mesh topology is quite different. It is the most difficult to build—there is no simple rule to follow (rule for Star: connect everyone to Router A; rule for Full Mesh: connect everyone to everyone). If built incorrectly, the partial mesh layout can have many of the disadvantages of the former topologies without many of the benefits. If built correctly, the opposite is true—more advantages, fewer disadvantages.

Building a successful partial mesh topology is where the interactive use of our social network measures really comes into play. The design below evolved after several iterations. With every iteration the average path length dropped until it appeared to reach a plateau where no further changes lowered the hop count without noticeably increasing the number of physical links. For the Partial Mesh, the three competing goals delineate as follows:

- *Reducing hop count:* The short average path length (1.667) throughout the network meets this goal well. Any router can *reach* any other router in two steps or less. Path length is less than that for the Star and Ring topologies.
- *Reducing available paths:* The number of available paths in the network (72) is the highest among all topologies, though not significantly more than the Ring topology. As the number of nodes in a network increases, this could become a problem—the average path length vs. path count trade-off needs to be closely monitored.
- *Reducing network failures:* Network centralization (0.000) is the same as for the Full Mesh topology—no router, nor link, is more important than any other. As nodes or links are removed from this network, it does not fragment quickly. Chances are slim that the number of failures necessary to fragment the network will actually occur within the same time period. Although we optimized our network centralization for this small "toy" network, we cannot expect this for most real networks. Yet, the goal remains to keep this metric as small as possible.

This topology in Figure 5 was built starting with a Ring topology—a simple architecture. A link was added and the network was remeasured. Was this structure better than the previous? If so, the current structure was kept and another link was added and the network was remeasured. This iterative process was continued until no further improvements happened after several changes. This process does not guarantee an *optimum* solution, yet it quickly converges on a *good* solution—even large networks improve quickly with just a few added links.

Figure 5: Routers in Partial Mesh Topology



A quirky aspect of networks is that sometimes you can subtract by adding—add a link to a network and reduce the average path length. The opposite also works, sometimes. You can add by subtracting—remove a tie and watch the average hop count grow. Yet, you never know for certain what effect adding or removing a link will have—it is neither a linear nor a local phenomenon. The size and direction of these changes depend upon the existing topology of the network and the location of the added or removed tie. It is key to have a model that allows quick what-if calculations.

Let's experiment with removing random ties—a situation similar to links between routers failing. If we remove the link between Router A and Router H in Figure 5, the number of geodesics in the network increases from 72 to 76, and the average path length increases to 1.815. Yet, removing a different link, G to F, reduces the the number of geodesics in the network from 72 to 66, while the average path length increases only to 1.727. If we are concerned about too many paths in the network, we can remove another link, B to C. This further decreases the number of shortest paths to 60, while reducing physical links to 10. This is very near the 56 paths in the very efficient star topology. Whereas the star is very vulnerable because of its single point of failure, this partial mesh, with the two links removed, is still robust. While the number of geodesics drops, the average path length creeps up slightly to 1.80 with the removal of the second link. Figure 5 has no paths greater than two hops. With the two links (G to F, B to C) removed, we now have 8 geodesics of three hops, while at the same time 12 fewer geodesics to load into routing tables, and two fewer physical links. It is a constant trade-off.

NSFnet Backbone

The NSFnet Backbone network, shown in Figure 6, connected the supercomputing centers in the USA in 1989. It is a partial mesh design that functions as a real-life example to test our social network algorithms.

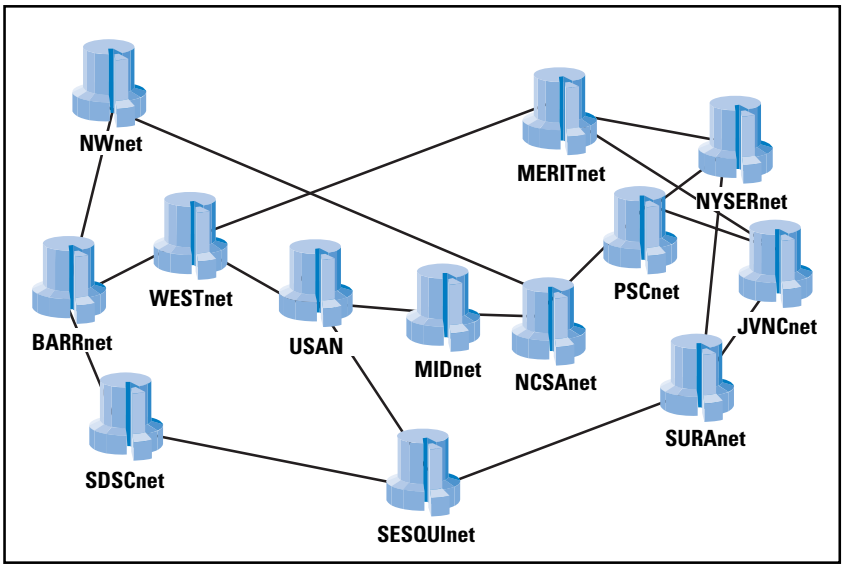


Figure 6: NSFnet in 1989

We remember our three competing goals for good internetwork design.

- Reducing hop count: average path length in steps/hops
- Reducing available paths: total geodesics in the network
- Increasing the number of failures the network can withstand: network centralization

What happens to these goals as we experience failures in the links or the nodes of the network? Table 1 shows the base metrics for Figure 6 and then shows what happens to the metrics, and our three goals, when five different failures occur.

Table 1: Possible Link and Node Failures

Scenario	Number of Geodesics in the Network	Network Centralization	Longest Path (hops)	Average Path Length (hops)
Original Design (Figure 6)	200	0.062	4	2.370
1) Node failure: NCSA	180	0.208	5	2.689
2) Node failure: MID	180	0.083	4	2.489
3) Node failure: JVNC	148	0.046	4	2.324
4) Link failure: NCSA–PSC	230	0.167	6	2.974
5) Link failure: USAN–MID	212	0.123	5	2.660
6) Link failure: MERIT–JVNC	192	0.069	4	2.458

The most damaging was link failure 4—the link failure between NCSA and PSC. This link is between two of the most central nodes in the network. If the flows between nodes are distributed somewhat evenly, then this link is one of the most traveled in the network.

The least damaging is node failure 3—the node failure at JVNC. In fact, this failure improved most metrics! By removing this node from the network, the number of network paths drops significantly, network centralization decreases, path length decreases slightly, and the longest path is still four hops.

The original NSFnet topology design is very efficient. I tried two different strategies to improve the network. The first strategy involved moving existing links to connect different pairs of routers. No obviously better topology was found by rearranging links among the routers. I was not able to find a better design that reduced both the number of geodesics and the average path length without significantly increasing the number of physical links in the network.

The second strategy is counter-intuitive, yet often networks respond well to this approach. It is the "subtracting by adding" approach described above. By adding new links in the right place in the network, we not only reduce the distance between nodes, we also decrease the number of geodesics in the network.

Because the NSFnet nodes had a maximum limit of three direct neighbors, I started connecting the nodes of Degree = 2. Options 1 through 3 show the various combinations and their effect on the total network. The improvements are minimal, yet each option offers specific strengths.

Option 2 offers more improvements than the others.

- The longest geodesic was reduced to three hops.
- The average path length was reduced throughout the network.
- The number of paths for the routers to remember was reduced slightly.
- Network centralization did not increase enough to noticeably affect the number of failures the network could withstand.

Table 2: Possible Network Improvements

Scenario	Number of Geodesics in the Network	Network Centralization	Longest Path (hops)	Average Path Length (hops)
Original Design (Figure 6)	200	0.062	4	2.370
Option 1 (add link: SDSC–MID)	202	0.071	4	2.287
Option 2 (add link: NW–DSC)	198	0.074	3	2.273
Option 3 (add link: NW–MID)	202	0.050	4	2.356

The improvement in Option 2 (add link: NW–SDSC) was actually implemented in the 1991 version of NSFnet—an excellent example of the "subtracting by adding" network dynamic. Networks are complex systems. How the network responds to change is based on the distribution and pattern of connections throughout the network.

Conclusion

In the real world we may not have the flexibility to experiment with our network model as we have with these examples. There will be more constraints. The information flows in your organization may require that specific pairs of routers have direct links—even if those connections would not be recommended by the algorithms we have been examining. Yet, when we have our "must-have" connections in place, we can experiment with the placement of the remaining connections using these social network metrics to indicate when we are getting close to a robust, yet efficient topology.

Given "initial conditions," social network methods can model our computer networks and suggest link changes[7] to form an effective topology that has a short average hop count, not too many paths, and just enough redundancy.

References

[1] Krebs V., "Visualizing Human Networks," *Release 1.0*, Esther Dyson's Monthly Report, February 1996.

[2] Watts D., Strogatz S., "Collective Dynamics of Small World Networks," *Nature,* 4 June 1998.

[3] Freeman L., "Centrality in Social Networks: A Conceptual Clarification," *Social Networks,* No. 1, 1979.

[4] Krackhardt D., "Assessing the Political Landscape: Structure, Cognition, and Power in Organizations," *Administrative Science Quarterly,* No. 35, 1990, page 351.

[5] Burt, Ronald S., *Structural Holes—The Social Structure of Competition,* ISBN 0674843711, Harvard University Press, 1992.

[6] Retana, A., Slice, D., White, R., *Advanced IP Network Design,* ISBN 1578700973, Cisco Press, 1999.

[7] Hagen G., Discussions with fellow network researcher, Guy Hagen, regarding combinatorial algorithms and models for recommending changes to improve the overall topology of a network.

VALDIS E. KREBS leads his own management consulting firm—orgnet.com He holds an undergraduate degree in Mathematics & Computer Science and a graduate degree in Human Resources. Since 1988 he has applied organizational network analysis to improve knowledge work within and between Fortune 500 firms such as IBM, Lucent, TRW, and supported consulting firms such as Ernst & Young, PricewaterhouseCoopers, and Booz-Allen-Hamilton. In addition to knowledge networks, he has applied these methodologies to mapping, measuring, and molding strategic alliances, communities of interest, emergent structures on the WWW, and internetworks. His work has been referenced in many publications, including the *Wall Street Journal, Entrepreneur, Training, PC Magazine, ZDNet, Corporate Leadership Council's Best Practices Reports, Knowledge Management, Across the Board, Business Week, HR Executive, Personnel Journal, FORTUNE,* and Esther Dyson's influential information industry newsletter, *Release 1.0.* He writes a regular column, "Working in the Connected World," for the *IHRIM Journal.* His Web site is at: **www.orgnet.com** and his e-mail is: **valdis@orgnet.com**

New Frontiers for Research Networks in the 21st Century

by Robert J. Aiken, Cisco Systems, Inc.

A famous philosopher, Yogi Berra, once said, "Prediction is hard. Especially the future."[1] In spite of this sage advice, we will still make an attempt at identifying the frontiers for research networks. By first examining and then extrapolating from the evolution and history of past research networks, we may be able to get an idea about the frontiers that face research networks in the future. One of the initial roles of the research network was to act as a testbed for network research on basic network protocols, mostly focusing on network Layers 1 through 4 (that is, the physical, data link, network, transport, and network management layers), but also including basic applications such as file transport and e-mail. During the early phases of the Internet, the commercial sector could not provide the network infrastructure sought by the research and education communities. Consequently, research networks evolved and provided backbone and regional network infrastructures that provided production-quality access to important research and education resources such as supercomputer centers and collaboratories[2]. Recent developments show that most research networks have moved away from being testbeds for network research and have evolved into production networks serving their research and education communities. It's time to make the next real evolutionary step with respect to research networks, and that is to shift our research focus toward maximizing the most critical of resources—*people*.

Given the growth and maturity of commercial service providers today, there may no longer be a pressing technical need for governments to continue to support pan-national backbone networks, or possibly even production-like national infrastructures, for Internet-savvy countries. Since commercially available *Virtual Private Networks* (VPNs) can now easily support many of the networked communities that previously required dedicated research networks, government and other supporting organizations can now support their research and education communities by providing the funding for backbone network services much as it does for telephony, office space, and computing capabilities; that is, as part of their research award. However, there may be valid social, political, and long-term economical reasons for continuing the support for such networks. For instance, a nation may decide that in order to ensure its economic survival in the future it wishes to accelerate the deployment and use of Internet technologies among its people, and thus the nation may decide to subsidize national research networks. In addition, it should be noted that VPNs often recreate the "walled" separation of communities, a scenario that was previously accomplished through the hard multiplexing of circuits.

But, in order to make technical advances in the e-economy, governments should now focus on supporting the evolution of intelligent and adaptable edge and access networks. These, in turn, will support the *Ubiquitous Computing* (UC) and persistent presence environments that will soon be an integral part of our future Internet-based economies.

The United States's recently expanded *National Science Foundation* (NSF)[3] research budget and the *Defense Advanced Projects Agency's* (DARPA's)[4] prior support of middleware research are good examples of moving in the right direction. The Netherland's Gigaport[5] project, which incorporates network and application research as well as an advanced technology access and backbone network infrastructure, is a good example of how visionary research networks are evolving.

Just as Internet technologies and network research have matured and evolved, so should the policies concerning the support of research networks. Policies need to be developed to again encourage basic network research and the development of new technologies. In addition, research networks need to encourage and accentuate new network capabilities in edge networks, on campus infrastructures, and in the end systems to support the humans in these new environments. This article focuses mainly on the future of research networks in e-developed nations; but, this is not to diminish the need or importance for e-developed nations to help encourage the same development in network-challenged nations.

Context and Definitions

Before delving into our discussion, we first need to define a few terms. These definitions will not only aid in our discussion, but may also help to highlight the role and function of various types of research networks. The most important terms to define are "network research" and "research network," both of which often get interchanged during discussions concerning policy, funding, and technology.

In this article, the term "network research" means long-term basic research on network protocols and technologies. The many types of network research can be categorized into three classes. The first category covers research on network transport infrastructure and generally includes research on the *Open System Interconnection* (OSI) Model Layers 1 through 4 (that is, the physical, data link, network, and transport layers) as well as research issues relating to the interconnection and peering of these layers and protocols. We will refer to this class of research as "transport services."

The second class consists of research covering what can nominally be referred to as "middleware"[6]. Middleware basically includes many of the services that were originally identified as network Layers 4 through 6. Layer 4 is included because of the need for interfaces to the network layer (sockets, TCP, and so on).

In addition, it nominally includes some components, such as e-mail gateways or directory services, which are normally thought of as being network applications, but which have subcomponents that may also be included in middleware. Given that the definition of middleware is far from an exact science, we shall say that middleware depends on the existence of the network transport services and supports applications.

The third area covers research on the real applications (for example, e-commerce, education, health care, and so on), network interfaces, network applications (for example, e-mail, Web, file transfer, and so on), and the use of networks and middleware in a distributed heterogeneous environment. Applications depend on both the middleware and transport layers. Advanced applications include *Electronic Persistence Presence* (EPP) and UC. EPP, or e-presence, describes a state of a person or application as always being "on the network" in some form or another. The concept of session-based network access will no longer apply. EPP assumes that support for UC and both mobile and nomadic networking exists. UC refers to the pervasive presence of computing and networking capabilities throughout all of our environments; that is, in automobiles, homes, and even on our bodies.

A "research network," on the other hand, is a production network; that is, one aspiring to the goal of 99.99999-percent "up time" at Layers 1 through 3, which supports various types of domain-specific application research. This application research is most often used to support the sciences and education, but can also be used in support of other areas of academic and economic endeavor. These networks are often referred to as *Research Networks* (RNs) or *Research and Education* (R&E) *Networks*. In this article, we further classify these RNs based on their general customer base. *Institutional Research Networks* (IRNs) support universities, institutes, libraries, data warehouses, and other "campus"-like networks. *National Research Networks* (NRNs)[7], such as the Netherland's Gigaport or Germany's DFN networks, support IRNs or affinity-based networks. *Pan National Research Networks* (PNRNs) interconnect and support NRNs. An example of a couple of current production PNRNs are Dante's Ten-155 and the NORDUNET[8] networks. In this article we will also classify the older *National Science Foundation Network's* (NSFNET's), *very-high-performance Backbone Network Service* (vBNS), CANARIE's CA*NET 3[9], and the Internet 2[10] Abilene networks as PNRNs because in terms of scale and policy they address the same issues of interconnecting a heterogeneous set of regionally autonomous networks (for example, NSFNET's regionals and Internet 2's Gigapops) as do the PNRNs.

A hybrid state of RN also exists. When we introduce one or more advanced technologies into a production system, we basically inject some amount of chaos into the system. The interplay between the new technologies and other existing technologies at various levels of the infrastructure, as well as scaling issues, can cause unanticipated results.

Research quality systems engineering and design is then required to address these anomalies. An example of this phenomenon is the problem encountered with ATM cell discard and its effect on TCP streams and subsequent retransmissions (that is, early packet discard and partial packet discard). The term *Virtual Private Network* (VPN) is used in this article in the classical sense; that is, a network tunneled within another network (for example, IP within IP, ATM virtual circuits [VCs], and so on), and it is not necessarily a security-based network VPN. *Acceptable Use Policy* (AUP) refers to the definition of the type of traffic or use that is allowed on a network infrastructure. *Conditions of Use* (COU) is basically another version of AUP.

Background

During the early phases of the evolution of research networks and the Internet, national research networks were building and managing backbone networks because there was a technical reason to do so. Governments supported these activities, because at the time the commercial sector Internet Service Providers (ISPs) could not do it and the expertise to do so resided within the R&E community. Much of the research or testing of this time still focused on backbone technologies as well as aggregation networks and architectures. Research networks started out by supporting longer-term risky network research and quickly evolved to support shorter-term no-risk production infrastructure.

The research during the *Advanced Research Projects Agency Network* (ARPANET) and early NSFNET phases of the Internet focused on basic infrastructure protocols and technologies. Now commodity services, these services are both easily and cost-effectively available from the commercial sector. We have come a long way since then. Except for a few universities and research centers, the commercial sector now dominates R&D in the backbone technology space. Commercially provided VPNs can now cost-effectively support most of the requirements of the R&E communities. Given the current domination of R&D in backbone technologies by the commercial sectors, as well as the need to address true end-to-end services, it is time that network research and research networks realign their focus onto the research and development of end-system and campus and edge network technologies. Most of the intelligence of the network (for example, *Quality of Service* [QoS], security, content distribution and routing, and so on) will live at the edges, and in some way will be oblivious to the backbone service over which it will operate. In addition, in order for applications to be able to make use of this network, intelligent RNs need to be able to provide the middleware and services that exist between the application and the transport systems. The real future for most RNs is in helping to analyze and identify, not necessarily run and manage, advanced network infrastructures for their R&E communities.

One of the problems faced by the R&E community is how to obtain support from their governments and other supportive organizations (both for-profit and nonprofit). In attempts to support advanced applications and end-user research, organizations and governments may be convinced into supporting RNs, which end up providing commodity services and competing with the commercial sector. One reason that this can occur is that governments often wish to see results very quickly in order to justify their support of the research community; but, by doing so they drive the recipient researchers and research network providers to focus on short-term results and abandon basic long-term research. This pressure from the supporting organizations can also force researchers to compete in a space—that is, transport layers—for which industry may be better suited and adapted in both scale and time. Another issue facing today's research networks is that many of the R&E community, who once would endure downtime and assume some risk in trade for being part of an experimental network, are now demanding full production-quality services from those same R&E networks. Subsequently, the RNs are then being precluded from aggressively pursuing and using really advanced technologies that may pose a risk. And finally, many times research networks, science communities, and researchers claim they are doing network research, when in reality they are not, because they wish to have decent network connectivity, and they assume that this is the only way to get funding and support for good network connectivity with which to support their real research objectives. All of these issues have driven RNs at all levels into difficult positions. RNs need to be able to again take risks if they are to push the envelope in adopting new technology. Likewise, it is also valid to provide production-quality network transport services to support research for middleware, network application (for example collaborative technologies), and R&E application (for example, medical, sciences, education, and so on) research. All of these requirements need to be addressed in the manner most expedient and cost-effective to the government or organization providing the support.

All research carries with it a certain amount of risk. There is theoretical and experimental research. Some research is subject to validation; some is *retrospective*—for example, examining packet traces to verify the existence of nonlinear synchronization—but some is *prospective* and involves reprogramming network resources, and any reprogramming is susceptible to bugs. The amount of risk often depends on the area of research undertaken. The lower down in the network structure that one performs experimental research, the more difficult it is to support this research and still maintain a production-like environment for the other researchers and applications; yet we need to provide support for all levels of experimental research, as described in MORPHNET[11]. The ideal environment would support applications that could easily migrate from a production network to one prototyping recent network research, and then back again if the experiment fails. Recent advances in optical networking show promise in realizing this goal, but many technical and policy-based challenges are yet to be addressed.

ARPANET and Early NSFNET Phase: 1980s

The ARPANET, one of the many predecessors of today's Internet, was a research project run by researchers as a sandbox where they could develop and test many of the protocols that are now integral components of the Internet. Because this was a research network that supported network research, there were times the network would "go down" and become unavailable. Although that was certainly not the goal, it was a reality when performing experimental network research. This was acceptable to all involved and allowed for the quick "research-to-production" cycle, now associated with the Internet, to develop. The management of the network with respect to policy was handled by the *Internet Activities Board* (IAB), which has since been renamed the *Internet Architecture Board,* and revolved around the actual use of the network as a research vehicle. The research focused mainly on Layers 1 through 4, and application research was secondary and used to demonstrate the underlying technologies.

At the end of the 1980s, the Internet and its associated set of protocols rapidly gained speed in deployment and use among the research community. This started the major shift away from research networks supporting experimental network protocols toward RNs supporting applications via production research networks; for example, the mission agencies' (that is, those agencies whose mission was fairly well focused in a few scientific areas) networks at the *Department of Energy* (DoE) (ESnet[12]) and NASA (NSInet). At the same time, the NSFNET was still somewhat experimental with the introduction and use of "home-grown" T1 and T3 routers, as well as with pioneering research on peering and aggregation issues associated with the hierarchical NSFNET backbone. It also focused on issues relating to the interconnection of the major agency networks and international networks at the *Federal Internet Exchanges* (FIXes), as well as the policy landscape of interconnecting commercial e-mail (MCIMail) with the Internet. The primary policy justification for supporting these networks (for example ESnet, NSInet, NSFNET) in the late 1980s was to provide access to scarce resources, such as supercomputer centers, although the NSFNET still supported network research, albeit on peering and aggregation.

In addition, the NSFNET was first in pioneering research on network measurement and characterization, leading to today's *Cooperative Association for Internet Data Analysis* (CAIDA) as well as to Surveyor installations on Abilene. As researchers became dependent on the network to support their research, the ability to introduce new and risky technologies into the network became more difficult, as shown by the second-phase T3 router upgrade for the NSFNET when many researchers vehemently complained about any "downtime."

At this time, there were still no commercial service providers from which to procure IP services to connect the numerous and varied sites of the NSFNET and other research networks. Hence there were still valid technical reasons for NRNs and R&E networks to exist and provide backbone services.

The policy decisions affecting the interconnection of the agency networks at the FIXes, as well as engineering international interconnectivity, were loosely coordinated by an ad hoc group of agency representatives called the *Federal Research Internet Coordinating Committee* (FRICC). The FRICC became the *Federal Networking Council* (FNC) in the early 1990s, and then became the *Large-Scale Network* (LSN) working group by the mid-1990s.

The FNC wisely left the management of the Internet protocols to the IAB, the *Internet Engineering Task Force* (IETF), and the *Internet Engineering Steering Group* (IESG); however, the FNC did not completely relinquish its responsibility, as evidenced by its prominent role in prodding the development of *Classless Interdomain Routing* (CIDR) and originating the work that led to new network protocols (for example, IPv6).

The Next-Generation NSFNET: Early 1990s

During the early 1990s, the Internet evolved and grew larger. It could no longer remain undetected on the government policy radar screen. Many saw the NSFNET and agency networks as competing with commercial *Internet Service Providers* (ISPs). Because of the charters of the agencies of the U.S.-based RNs (for example NSF, DoE, NASA), all traffic crossing their networks had to adhere to their respective AUPs. These AUPs prohibited any "commercial entity-to-commercial entity traffic" to use a U.S. government supported network as transit. In addition, the demand for generic Internet support for all types of research and education communities became much stronger, and at the same time there was growing support among the U.S. Congress and Executive branches to end the U.S. Federal Government support of the U.S. Internet backbone.

In response to these pressures and the responses to a NSF draft "New NSFNET" proposal, the NSF elected to get out of the business of being the Internet backbone within the United States. This policy change was the nexus for the design of the vBNS, *Network Access Points* (NAPs), and *Routing Arbiter* (RA) described in the ABF paper[13] by early 1992. The vBNS was meant to provide the NSF supercomputer sites a research network that was capable of providing the high-end network services required by the sites for their Metacenter, as well as to provide the capability for their researchers to perform network research because the centers were still the locus for network expertise. The NAPs were designed to enhance the AUP free interconnectivity of both commercial and R&E ISPs and to further evolve the interconnection of the Internet started by the FIXes and the *Commercial Internet eXchange* (CIX).

The research associated with NRNs is already evolving from dealing with mainly IP and transport protocol research to research addressing the routing and peering issues associated with a highly interconnected mesh of networks. Research was an integral part of the NAP and RA design, but it was now focused on peering of networks as opposed to the transport layer protocols themselves. Although this network was not official until 1995, commercial prototype AUP free NAPs (for example, MAE-EAST) immediately sprang up and hastened the transition to a commercial network. The network was transformed from a hierarchical network topology to a decentralized and distributed peer-to-peer model. It no longer existed for the sole purpose of connecting a large aggregation of R&E users to supercomputer centers and other "one-of-a-kind" resources. The NAPs and the "peering" advances associated with the NAPs constituted a very crucial step for the success of applications such as the *World Wide Web* (WWW) and the subsequent commercialization of the Internet because they provided the required seamless interconnected infrastructure. Although some ISPs, for example UUNET and PSInet, were quickly building out their infrastructure at that time, there still existed the need for PNRNs to act as brokers for acquiring and managing end-to-end IP services for their R&E customer base; it would not be much longer, however, before the ISPs had the necessary infrastructure in place to do this themselves.

The Internet 2 Phase: 1996–2000

The transition to the vBNS, NAP, and RA architecture became official early in 1995 and, as a result, the United States university community lost its government-subsidized production backbone. NSF-supported regionals had lost their support years earlier, and many had already transitioned to become commercial service providers, and the NSF "connections" program for tier 2 and lower schools persisted because it was felt (policy wise) that it was still valid to support such activities. The result of this set of affairs led to the creation of the Internet 2. Many of the top research universities in the United States felt that the then-current set of ISPs could not affordably provide adequate end-to-end services and bandwidth for the academic community's perceived requirements. As a result, the NSF decided to again support production-quality backbone network services for an elite set of research institutions. This was clearly a policy decision by NSF that had support from the U.S. Congress and Executive branches of government, even though in the early 1990s both Congress and the Executive branches were fairly vocal about not supporting such a network.

The initial phase was to expand to the vBNS and connect hundreds of research universities. The vBNS again changed from a research network, connecting a few sites and focusing on network and Metacenter research, back into a production research network. The vBNS is soon eclipsed by the OC-48 Abilene network. Gigapops, which are localized evolutions of NAPs, are used to connect the top R&E institutions to the Internet 2 backbones (that is, vBNS and Abilene).

These backbones were subject to COU as a way to restrict the traffic to that in direct support of R&E, much like the NSFNET was subject to its AUP.

The ISPs who complained so bitterly about unfair competition in the early 1990s no longer cared, because they had more business than they could handle in selling to corporate customers. An ironic spin on this scenario is that the business demands placed on the commercial ISPs by the late 1990s drove them to aggressively adopt new technologies to remain competitive. Not only were they willing to act as testbeds, they paid for that privilege since it gave them a competitive edge. The result is that in a lot of cases regarding the demonstration and testing of backbone-class technologies, the R&E community was time-wise behind the commercial sector. This situation is further aggravated by the fact that many, but not all, backbone network-savvy R&E folks went to work in industry. Another side effect of this transition is the loss of available network monitoring data. The data used by CAIDA, *The National Laboratory for Applied Network Research* (NLANR), and other network monitoring researchers had been gathered at the FIXes where most traffic used to pass. With the transition to a commercially dominated infrastructure, meaningful data becomes harder to obtain. In addition, as a result of the COU of the Internet 2 network, and the type of applications it supports (for example, trying to set bandwidth speed records), the traffic passing over its networks can no longer be assumed to be representative Internet data, and its value in this regard is diminished.

Another milestone is reached. ISPs have grown or merged so that they are offering both wide- and local-area network services, and anyone can now easily acquire national and international IP and transport services. The deployment and use of VPNs allows the commercial service providers (SPs) to provide and support various acceptable policy networks with differing AUP/COU on the same infrastructure. The technical need for most PNRNs or NRNs to exist to fulfill this function fades away. Researchers should now be able to specify wide-area network support as a line item in their research proposal budgets, just as they do for telephony and computing support. Most governments do not support separate research "Plain Old Telephone Service" (POTS) networks so that researchers can talk with one another. They provide funding in the grants to allow the researchers to acquire this from the commercial sector. However, valid technical reasons for selectively supporting some research networks still exist. A prime example is the CA*Net 3 network in Canada, which has been extremely aggressive in the adoption and use of preproduction optical networking technologies and infrastructure and has been instrumental in advancing our knowledge on this area.

During this evolution of research networks capabilities, network research is also going through its own evolution. DARPA starts focusing its research on optics, wireless, mobility, and network engineering as part of its Next-Generation Internet program. In addition, the research moves up the food chain of network layers. DARPA and DoE start supporting research on middleware. Globus[14], along with Legion[15], Condor[16], and POLDER[17], are major middleware research efforts that become the main impetus for GRIDs; and although they are focused mainly on seeking the holy grail of distributed computing, many of the middleware services they are developing are of value in a broader research and infrastructure context. The focus of network research and research networks now starts moving away from backbone transport services to research on advanced collaboratory, ubiquitous computing, mobile, nomadic, and EPP environments.

The policy management of the Internet now becomes an oxymoron and reflects the completion of the transition of the Internet to a distributed commercial Internet. Many organizations are now vying for a say in how the Internet evolves. Even the IETF is suffering from its own success. It now faces many of the same political challenges the ITU faced, that is, some commercial companies now try to affect the standards process for their own benefit by introducing standards contributions and only later disclosing the fact that they have filed patents on the technology in question. It is now much more difficult to make policy decisions regarding the future of Internet protocols, technologies, and architectures.

Future Frontiers

UC and EPP are the paradigm shifts at the user level that are already drastically altering our concept and understanding of networks. The scale, number, and complexity of networks supporting these new applications will far exceed anything we have experienced or managed in the past. Users will "be on the net" all the time, either as themselves or indirectly through agents and "bots." They will be mobile and nomadic. There will be "n" multiple instances of a user active on a network at the same time, and not necessarily from the same logical or geographical location. The frontiers associated with this new focus are many times more complex from a systems integration level than any work we have done in the past with backbone networks. This new frontier will provide new technical challenges at the periphery of the network; that is, the intelligent access and campus networks necessary to support these new environments. EPP and UC will drastically affect our research networks and application environments, much as the Web and its protocols drastically changed Internet and traffic patters in the 1990s.

The frontiers faced by research networks of the future will depend upon many technical and sociopolitical factors on a variety of levels. The sociopolitical frontiers can be divided into two different classes, one for e-developed nations who have already gone through the learning process

of building an Internet-based infrastructure, and another for the e-challenged nations who still face the challenges of building a viable network transport infrastructure. The developed nations need to now grapple with how they can encourage the next evolutionary phase of their Internet-based economies. Because of the fast evolution of technology, the technical need for subsidizing transport-based network infrastructure is no longer the pressing need it was in the 1990s. The future research network will most likely be nothing more than a VPN based on a commercial ISP "cloud" service that interconnects researchers. The *High Energy Physicists* (HEPs) have already proved that life as a VPN-based affinity group overlaid on production network services is a viable solution to providing for their network requirements. The *High-Energy Physics Network* (HEPnet)[18] is a virtual set of users and network experts using ESnet and other ISP VPN-based network services to support the HEP scientists. Although we still have some technical challenges associated with backbone network technology (for example, optics), there are now only a very small number of institutions and organizations capable of working with industry and making substantial contributions in this area.

The new technical challenges that need to be addressed now include how to build and deploy intelligent edge and campus networks, content delivery and routing, mobile/nomadic/wireless access to the Internet, and the support for both UC and EPP. The latter two require major advancements and will require a whole bevy of middleware that is both network aware and an integral component of an intelligent network infrastructure. This includes, but is not limited to, directories, locators, presence servers, call admission control services, self-configuring services, mobility, media servers, policy servers, bandwidth brokers, intrusion-detection servers, accounting, authentication, and access control. IRNs and RNs can contribute to our knowledge and growth of these new areas by acting as leaders in areas that tend to be more difficult for the commercial sector to address, for instance, the development and deployment of advanced end-to-end services that operate over one or more ISP-provided clouds. Examples include interdomain bandwidth broker services, multi *Public Key Infrastructure* (PKI) trust models, defining multisite policies and schemas for directory-based policy services, and developing scalable naming conventions.

In order for policy makers to make informed decisions on the evolution and support of Internet technologies and architectures, they will need access to a generic mix of real backbone network data. There still exists a dire need at this point for such data. Innovative solutions that respect the privacy and business concerns of all types of ISPs and RNs, while at the same time making available "scrubbed" data, need to be developed. In addition, with the new focus on edge and metro networks, we might be able to shift our monitoring attentions to this area as well in order to better understand traffic demands and patterns on these scales of networks. Network monitoring is only one of the challenges facing us.

As the scale and complexity of networks grows, even at the pico and body area network level, we will need to develop new techniques to support network modeling, simulation, and experimentation. The University of Utah is developing a test facility[19] comprising a large number of networked processors, the network equivalent of a supercomputer center, to be used experimentally in the design and development of new transport layer protocols.

Summary

"Being on the net" will change our way of doing e-everything, and the evolution of the underlying infrastructure will need to change in order to support this paradigm shift. The intelligence of the network will not only move to the periphery, but even beyond, to the personal digital assistant and body area network. Therefore, it is important that the goals and focus of the research networks also evolve. Leave the R&D associated with backbone networks mainly with the commercial sector because this is their raison d'etre. The research networks of the future will be mostly VPNs, with a few exceptions, as noted earlier in this article. Research networks need to focus on the new technologies at the periphery as well as the middleware necessary to support the advanced environments that will soon be commonplace. Many research networks will themselves become virtual, for example, HEPnet, providing expertise but not necessarily a network service.

Policy makers must adapt to address not only these substantial technical and architectural changes but also second-order policy issues such as security and privacy and how to ensure that we don't end up with a bifurcated digital economy of e-savvy and e-challenged communities.

E-developed nations have already been through the technology learning curve of implementing and deploying a transport infrastructure. The e-challenged nations, with respect to network infrastructure, still face these same challenges, and they have the benefit of taking advantage of the knowledge of the nations who have successfully made the transition. In order to speed up the deployment of Internet technologies and infrastructure in the e-challenged nations, it may be best to first create technologically educated people and then to provide them an economic and social environment where they can apply their knowledge and build the infrastructure. E-savvy nations should help by providing the "know-how." The *North Atlantic Treaty Organization* (NATO) has a joint program with the *Trans-European Research and Education Networking Association* (TERENA) to provide for the instruction of Eastern European nations on the use and deployment of Internet technology (that is, how to configure and manage routers).

In lieu of subsidizing networks in these nations, NATO and TERENA are providing the basic knowledge that these people need to build, manage, and evolve their own networks and infrastructure. This should be the model to consider for e-developing nations. This is not to diminish the challenges of building network infrastructure in some areas where there is no such infrastructure, and perhaps in some of these areas working with other utility infrastructure providers might advance this cause.

Disclaimer

The ideas, comments, and projections proffered in this article are the sole opinions of the author, and in no way represent or reflect official or unofficial positions or opinions on the part of Cisco Systems, Inc. This article is based on my experience designing and managing operational international research networks, as well as being a program manager for network research, during the formative years of the Internet (that is, my tenure as a program manager for the United States Government's National Science Foundation and the Department of Energy), and my recent experience within Cisco working with next-generation Internet projects and managing its University Research Program. Many of the examples that I cite in this work are based on the development and deployment of the U.S.-based Internet and research networks, although the lessons learned in the United States may also be illuminating elsewhere.

Gratitude

I would like to thank my friend and colleague, Dr. Stephen Wolff, of the Office of the CTO, Cisco Systems Inc., for many good suggestions with respect to improving the content and presentation of this article; but, mostly for his good-humored authentication of my history and facts.

References

[0] This article was presented at the third Global Research Village Conference organized jointly by the Organization for Economic Cooperation and Development (OECD) and the Netherlands in Amsterdam, December 6–8, 2000.

[1] This is also attributed to the famous Physicist Niels Bohr.

[2] Wulf, William A. 1988. "The National Collaboratory—A white paper," Appendix A. In "Towards a National Collaboratory," Unpublished report of a National Science Foundation invitational workshop. Rockefeller University, New York, March 17–18, 1989.

[3] **http://www.nsf.gov/**

[4] **http://www.darpa.mil/**

[5] **http://www.gigaport.nl/**

[6] **Draft-aiken-middleware-reqndef-01.txt**, Internet Draft, Work in Progress, May 1999, **http://www.anl.gov/ECT/Public/research/morphnet.html**

[7] See **http://www.dante.org/** and **http://www.terena.nl/** for full lists of European research networks.

[8] **http://www.nordu.net/**

[9] **http://www.canarie.ca/**

[10] **http://www.internet2.org/**

[11] "Architecture of the Multi-Modal Organizational Research and Production Heterogeneous Network (MORPHnet)," Aiken, et al, ANL-97/1 technical report, and 1997 Intelligent Network and Intelligence in Networks Conference.
http://moat.nlanr.net/Papers/iinren.ps

[12] **http://www.es.net/**

[13] "NSF Implementation Plan for an Interagency Interim NREN," (aka Architecture for vBNS, NAPs and RAs), Aiken, Braun, and Ford, GA A21174, May 1992.

[14] **http://www.globus.org/**

[15] **http://www.cs.virginia.edu/~legion/**

[16] **http://www.cs.wisc.edu/condor/**

[17] **http://www.science.uva.nl/projects/polder/**

[18] **http://www.hep.net/hepnrc.html**

[19] **http://www.cs.utah.edu/flux/testbed/**

ROBERT J. AIKEN has an MS in Computer Science from Temple University. He is the Manager of the Cisco University Research Program. Prior to joining Cisco, Bob was the network and security research program manager for DoE's HPCC program and Next-Generation Internet (NGI) initiative. He was a program manager at the National Science Foundation (NSF), and with colleagues Peter Ford and Hans-Werner Braun coauthored the conceptual design and architecture of the second-generation National Science Foundation Network (NSFNET) (vBNS, Network Access Points [NAPs], and the Routing Arbiter [RA]), which enabled the commercialization of the then-U.S.-federally supported Internet. Before his NSF tenure, he served as DoE's ESnet program manager and was the creator and manager of the ESnet Network Information and Services group. Prior to his career in networking, Bob was responsible for managing supercomputers and coding their operating systems. His academic experience includes being an Assistant Professor of Computer Science at Hood College in Maryland, an adjunct Professor at California State University, Hayward, and the Manager of Technology Services at Gettysburg College in Pennsylvania. E-mail: **raiken@cisco.com**

Book Review

Intrusion Detection

Network Intrusion Detection—An Analyst's Handbook, by Stephen Northcutt, ISBN 0735708681, New Riders Publishers, 1999.

Network security and the ability to detect intrusion attempts has become extremely important in today's networks, regardless of size. I was looking for a book that would get technical on the details in these matters. Laura Chappell, the guru of packet-level information (**www.packet-level.com**), recommended this book to me. I should have realized what I was getting into at that point. I purchased the book, which was a bit expensive for its size at $39.99, and eagerly began reading it.

Mr. Northcutt starts out with a good discussion on how Kevin Mitnick conducted his famous attack. The book presents some very good information on a variety of topics, intermixed with personal observations and opinion. This made for an enjoyable read. If you are considering getting an *Intrusion Detection System* (IDS), then this book will provide you with some valuable insight and guidelines to consider from a recognized industry expert in this field. Mr. Northcutt is affiliated with The *System Administration, Networking, and Security* (SANS) *Institute* (**www.sans.org**).

Be aware that this book is not for the faint of heart. You will dive into the depths of packets and intrusion detection rather quickly, and never look back. This is both good and bad. I prefer an easy-to-read technical book, but the level of technical knowledge required to make sense of many of the examples is rather extensive. This includes how the many trace examples are presented in rather specialized fashion; in addition, the touted "detailed" explanations varied in usefulness quite a bit.

The book was marketed as a training aid; however, I suspect most readers need to be quite experienced to benefit from it. I admit I had to read many sections more than once in order to grasp the finer points being conveyed. I am confident that many readers have already echoed this sentiment to the author and publisher, since the second edition of this book was published in September 2000 and the page count has doubled, with only a modest price increase. I put it on my Christmas list!

—Tom Thomas, Mentor Technologies Group
tothomas@mentortech.com

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

Fragments

New Top-Level Domains

On November 16, 2000 The board of directors of the *Internet Corporation for Assigned Names and Numbers,* (ICANN) announced its selections for registry operators for new top level domains. The applications selected for further negotiation are the following:

.aero	Societe Internationale de Telecommunications Aeronautiques SC, (SITA)
.biz	JVTeam, LLC
.coop	National Cooperative Business Association, (NCBA)
.info	Afilias, LLC
.museum	Museum Domain Management Association, (MDMA)
.name	Global Name Registry, LTD
.pro	RegistryPro, LTD

The ICANN staff will now work through the end of the year to negotiate registry agreements with the applicants selected. The proposed schedule for completion of negotiations is December 31, 2000. The negotiated registry agreements must then be approved by the board of directors. Following that approval, the ICANN board will forward its recommendations to the U.S. Department of Commerce for implementation. For more on the history of ICANN's new TLD application process, please see **http://www.icann.org/tlds/** Multimedia archives of the annual meeting can be reviewed at **http://cyber.law.harvard.edu/icann/la2000/**

ICANN is a technical coordination body for the Internet. Created in October 1998 by a broad coalition of the Internet's business, technical, academic, and user communities, ICANN is assuming responsibility for a set of technical functions previously performed under U.S. government contract by IANA and other groups. Specifically, ICANN coordinates the assignment of the following identifiers that must be globally unique for the Internet to function: Internet domain names, Internet Protocol address numbers, and protocol parameter and port numbers. In addition, ICANN coordinates the stable operation of the Internet's root server system. As a non-profit, private-sector corporation, ICANN is dedicated to preserving the operational stability of the Internet; to promoting competition; to achieving broad representation of global Internet communities; and to developing policy through private-sector, bottom-up, consensus-based means. ICANN welcomes the participation of any interested Internet user, business, or organization. See **http://www.icann.org**

ISOC Launches Platinum Membership Level

The *Internet Society* (ISOC) is pleased to announce its *Platinum Sponsorship Program,* The Platinum program, which is in addition to and distinct from ISOC's standard organizational membership categories, provides interested organizations with the ability to designate support for specific areas of ISOC's work.

The initial participants, who also helped define the program, included Cisco, IBM, Microsoft, Nortel, RIPE NCC and SoftComca.com. APNIC has since joined the list of Platinum sponsors. Platinum level sponsors contribute $100,000 annually, with non-profit organizations eligible for funding at half that amount.

The Platinum program was initially developed to bolster support for the standards activities of ISOC, specifically ISOC's support of the *Internet Engineering Task Force* (IETF). Recently the program was expanded beyond Standards to include the three remaining areas of ISOC activities: Education & Training, Public Policy, and Member Services. As a result, participants in the Platinum program can now earmark their contribution for any of these four functional areas, or choose to allocate support for multiple areas, should they so desire.

ISOC is dependent upon individual and organizational members for its funding. ISOC believes that allowing contributors to designate where their money will be spent through the Platinum program enhances the Society's ability to undertake activities in these four areas, and, at the same time, provides an attractive support option for many organizations. ISOC will provide a report on the use of funds to each Platinum-Level sponsor at the end of each year. More information on the Platinum-Level Support Program can be found at:
`http://www.isoc.org/isoc/membership/platinum.shtml`

More information on ISOC's standard membership categories is available from: **`http://www.isoc.org/orgs/benefits.shtml`**

100 Million Internet Hosts

The Internet reached 100,000,000 hosts on 2 November 2000, according to John S. Quarterman, founder of Matrix.Net, a provider of Internet performance, measurement and intelligence. From its humble beginnings of 4 sites in the western United States in December 1969, the Internet has now reached over 150 countries and is nearly pole to pole. "This is an impressive achievement," said Quarterman. "We have been tracking the growth and development of the Internet for this entire decade. If this kind of growth continues, we will hit 1,000,000,000 hosts in 2006." For more information, see **`http://www.matrix.net/`**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

Bulk Rate Mail
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

March 2001 Volume 4, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

The rapid growth of the Internet has led to numerous changes to the underlying technologies. In the early days, host names and their corresponding IP addresses were kept in a flat text file ("**HOSTS.TXT**"), updated weekly by the Network Information Center at SRI International. In the mid 1980s it became clear that this method of name/address mapping would not scale, and a new distributed lookup mechanism was designed and deployed. This new method, known as the *Domain Name System* (DNS), has proven successful even in the face of millions of Internet hosts.

Another result of Internet growth is the potential for depletion of the IP Version 4 (IPv4) 32-bit address space. In the early 1990s, this became a matter of great focus for the Internet Engineering Task Force (IETF). The "short-term" fix for this problem was to abandon the original concept of A, B and C address classes and introduce *Classless Interdomain Routing* (CIDR), which consumes addresses in a much more efficient manner—that is to say, more slowly. Address consumption has also been slowed by the use of *Network Address Translation* (NAT) and private address space. Predictions for when the Internet will finally run out of IPv4 addresses varies. The long-term solution is to replace IPv4 with IPv6 which uses 128 bits for addressing.

One area of Internet growth that is currently causing some concern among ISPs is the growing size of the routing table that each router participating in the *Border Gateway Protocol* (BGP) must keep in memory. Our first article, by Geoff Huston, is a detailed look at this problem. Geoff takes an historical look at the BGP routing table, and discusses ways to address some of the issues.

In our March 2000 issue, Geoff Huston wrote an article entitled "Quality of Service—Fact or Fiction?" that discussed the prospects for achieving QoS on an Internet-wide scale. In this issue, Bill Stallings looks at QoS in the LAN environment, which is generally easier to control than the Internet as a whole. LAN QoS has been standardized in IEEE 802.1D which is the subject of this article.

We apologize for the delay in getting our online subscription system up and running. It should be available in the very near future. Meanwhile, please continue to use **ipj@cisco.com** for any subscription questions or to give feedback on anything you read in this journal.

—Ole J. Jacobsen, Editor and Publisher

ole@cisco.com

Analyzing the Internet BGP Routing Table

by Geoff Huston, Telstra

The Internet continues along a path of seemingly inexorable growth, at a rate that has, at a minimum, doubled in size each year. How big it needs to be to meet future demands remains an area of somewhat vague speculation. Of more direct interest is the question of whether the basic elements of the Internet can be extended to meet such levels of future demand, whatever they may be. To rephrase this question, are there inherent limitations in the technology of the Internet—or its architecture of deployment—that may impact the continued growth of the Internet to meet ever-expanding levels of demand?

Numerous potential areas can be searched for such limitations, including the capacity of transmission systems, the switching capacity of routers, the continued availability of addresses, and the capability of the routing system to produce a stable view of the overall topology of the network. This article examines the Internet routing system and the longer-term growth trends that are visible within this system.

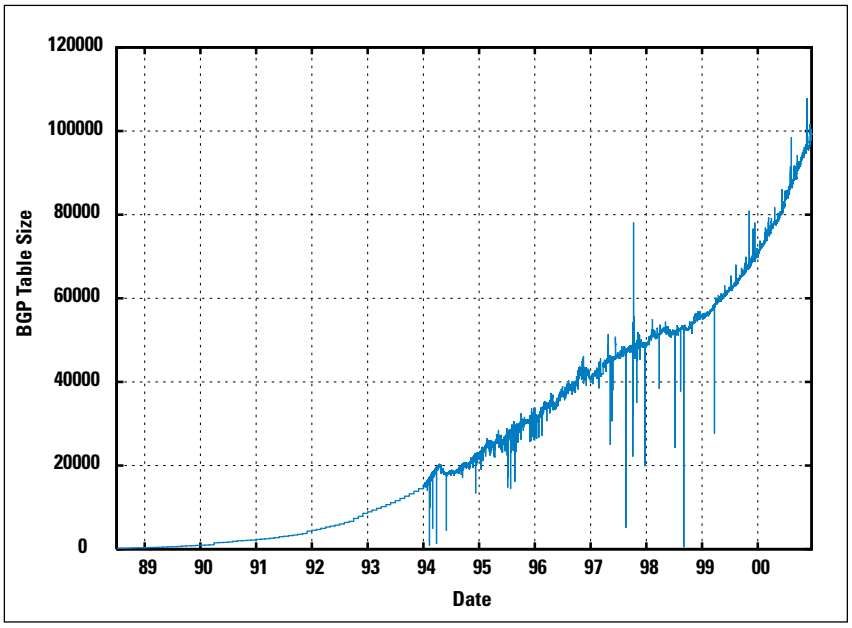
The structure of the global Internet can be likened to a loose coalition of semi-autonomous constituent networks. Each of these networks operates with its own policies, prices, services, and customers. Each network makes independent decisions about where and how to secure the supply of various components that are needed to create the network service. The cement that binds these networks into a cohesive whole is the use of a common address space and a common view of routing. Integrity of routing within each constituent network, or *Autonomous System* (AS), is maintained through the use of an interior routing protocol (or *Interior Gateway Protocol,* or IGP). The collection of these networks is joined into one large routing domain through the use of an inter-network routing protocol (or *Exterior Gateway Protocol,* or EGP).

When the scaling properties of the Internet were studied in the early 1990s, two critical factors identified in the study were, not surprisingly, routing and addressing[1]. As more devices connect to the Internet, they consume addresses, and the associated function of maintaining reachability information for these addresses implies ever-larger routing tables. The work in studying the limitations of the 32-bit IPv4 address space produced many outcomes, including the specification of IPv6, as well as the refinement of techniques of *Network Address Translation* (NAT) intended to allow some degree of transparent interaction between two networks using different address realms. Growth in the routing system is not directly addressed by these approaches, because the routing space is the cross product of the complexity of the topology of the network, multiplied by the number of autonomous domains of connectivity policy multiplied by the base size of a routing-table entry. When a network advertises a block of addresses into the exterior routing space, this entry is generally carried across the entire exterior routing domain of the

Internet. To measure the characteristics of the global routing table, it is necessary to establish a point in the default-free part of the exterior routing domain and examine the *Border Gateway Protocol* (BGP) routing table that is visible at that point.

Measurements of the size of the routing table were somewhat sporadic in the beginning, and many measurements were taken at approximately monthly intervals from 1988 until 1992 at Merit[2]. This effort was resumed in 1994 by Erik-Jan Bos at Surfnet in the Netherlands, who commenced measuring the size of the BGP table at hourly intervals at the start of that year. This measurement technique was adopted by the author in 1997, using a measurement point located at the edge of AS 1221 in Australia, again using an hourly interval for the measurement[6]. The result of these efforts is that we now have a detailed view of the dynamics of the Internet routing-table growth that spans 13 years (Figure 1).

Figure 1: BGP Table Growth 1988–2000



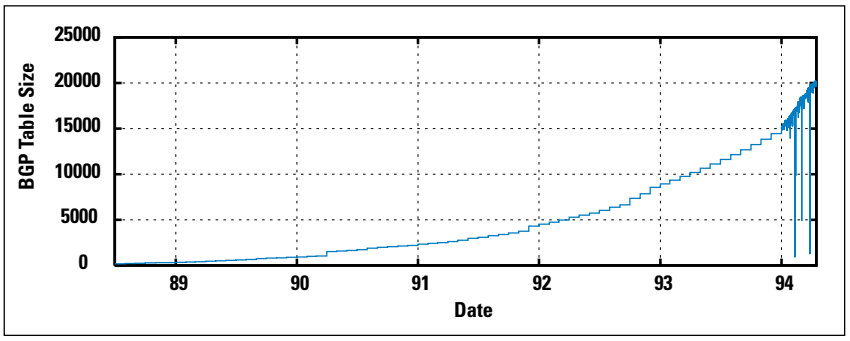
BGP Table Growth

At a gross level, there appear to be four distinct phases of growth visible in this data.

Pre-CIDR Growth

The initial characteristics of the routing-table size from 1988 until April 1994 show definite characteristics of exponential growth (Figure 2). Much of this growth can be attributed to the growth in deployment of the historical Class C address space (/24 address prefixes). Unchecked, this growth would have lead to saturation of the BGP routing tables in nondefault routers within a few years. Estimates of the time at which this would have happened vary somewhat, but the overall observation was that the growth rates were exceeding the growth in hardware and software capability of the deployed network at that time.

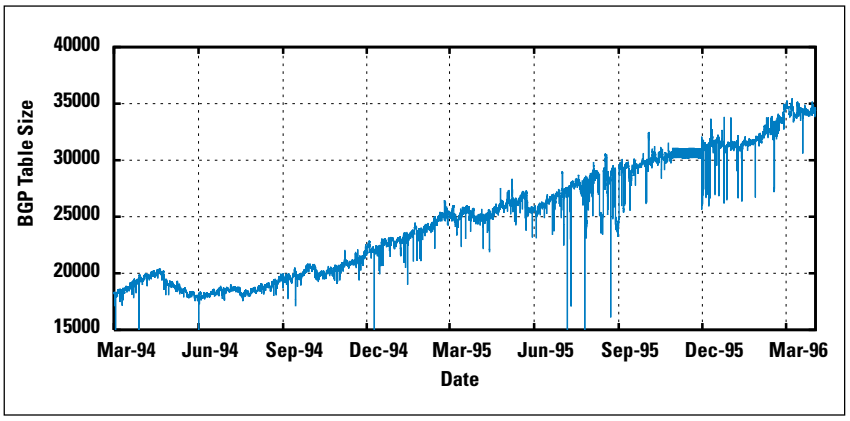
Figure 2: BGP Table Growth 1988–1994



CIDR Deployment

The response from the engineering community was the introduction of routing software that dispensed with the requirement for the Class A, B, and C address delineation, replacing this scheme with a routing system that carried an address prefix and an associated prefix length. A concerted effort was undertaken in 1994 and 1995 to deploy *Classless Interdomain Routing* (CIDR), based on encouraging deployment of the CIDR-capable version of the BGP protocol, BGP4. The effects of this effort are visible in the routing table (Figure 3). Interestingly enough, the efforts of the *Internet Engineering Task Force* (IETF) CIDR Deployment Working Group are visible in the table, with downward movements in the size of the routing table following each IETF meeting.

Figure 3: BGP Table Growth 1994–1995

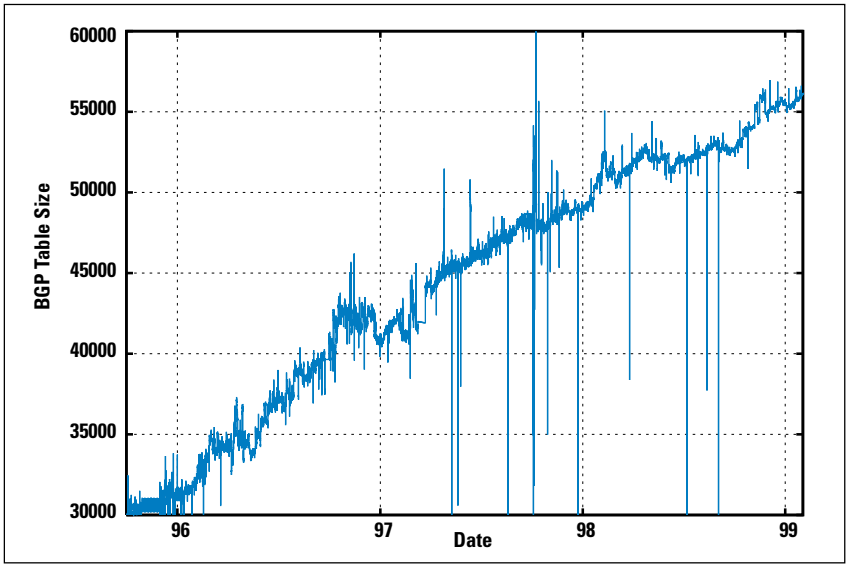


The intention of CIDR was one of supporting an address architecture termed "provider address aggregation," where a network provider is allocated an address block from the address registry, and announces this entire block into the exterior routing domain. Customers of the provider use a suballocation from this address block, and these smaller routing elements are aggregated by the provider and not directly passed into the exterior routing domain. During 1994, the size of the routing table remained relatively constant at approximately 20,000 entries as the growth in the number of providers announcing address blocks was matched by a corresponding reduction in the number of address announcements as a result of CIDR aggregation.

CIDR Growth

For the next four years until the start of 1998, CIDR proved remarkably effective in damping unconstrained growth in the BGP routing table. While other metrics of Internet size grew exponentially during this period, the BGP table grew at a linear rate, adding about 10,000 entries per year. (Figure 4). Growth in 1997 and 1998 was even lower than this linear rate. Although the reasons behind this are somewhat speculative, it is relevant to note that this period saw intense aggregation within the *Internet Service Provider* (ISP) industry, and in many cases this aggregation was accompanied by large-scale renumbering to fit within provider-based aggregated address blocks. During this period, credit for this trend also must be given to Tony Bates, whose weekly reports of the state of the BGP address table, including listings of further potential for route aggregation, provided considerable incentive to many providers to improve their levels of route aggregation[4].

Figure 4: BGP Table Growth 1995–1998

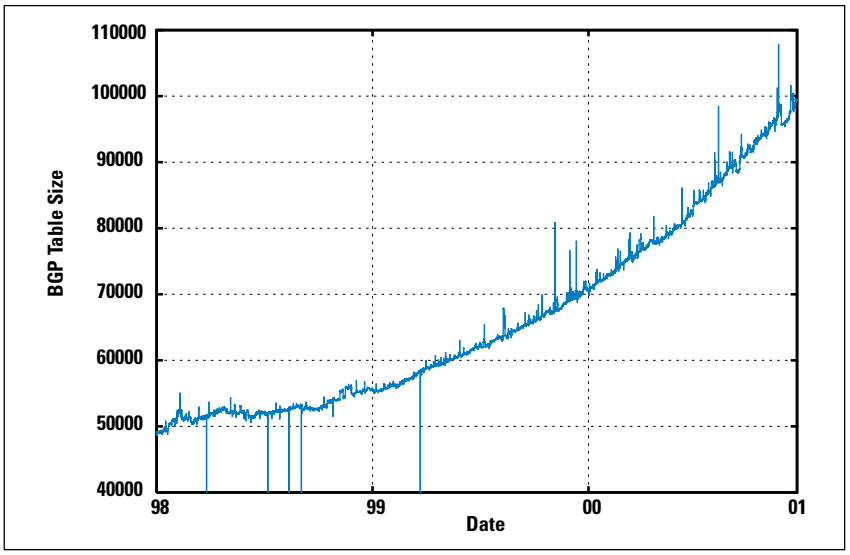


A close examination of the table reveals a greater level of stability in the routing system at this time. The short-term (hourly) variation in the number of announced routes decreased, both as a percentage of the number of announced routes and in absolute terms. One of the other benefits of using large aggregate address blocks is that an instability at the edge of the network is not immediately propagated into the routing core. The instability at the last hop is absorbed at the point at which an aggregate route is used in place of a collection of more specific routes. This, coupled with widespread adoption of BGP route flap damping, has been every effective in reducing the short-term instability in the routing space. It has been observed that whereas the absolute size of the BGP routing table is one factor in scaling, another is the processing load imposed by continually updating the routing table in response to individual route withdrawals and announcements. The encouraging picture from this table is that the levels of such dynamic instability in the network have been reduced considerably by a combination of route flap damping and CIDR.

Current Growth

In late 1998, the trend of growth in the BGP table size changed radically, and the growth for the past two years is again showing all the signs of a reestablishment of exponential growth. It appears that CIDR has been unable to keep pace with the levels of growth of the Internet. (Figure 5). Once again the concern is that this level of growth, if sustained, will outstrip the capability of hardware, or current capability of the BGP routing protocol, or possibly both.

Figure 5: BGP Table Growth 1998–2000



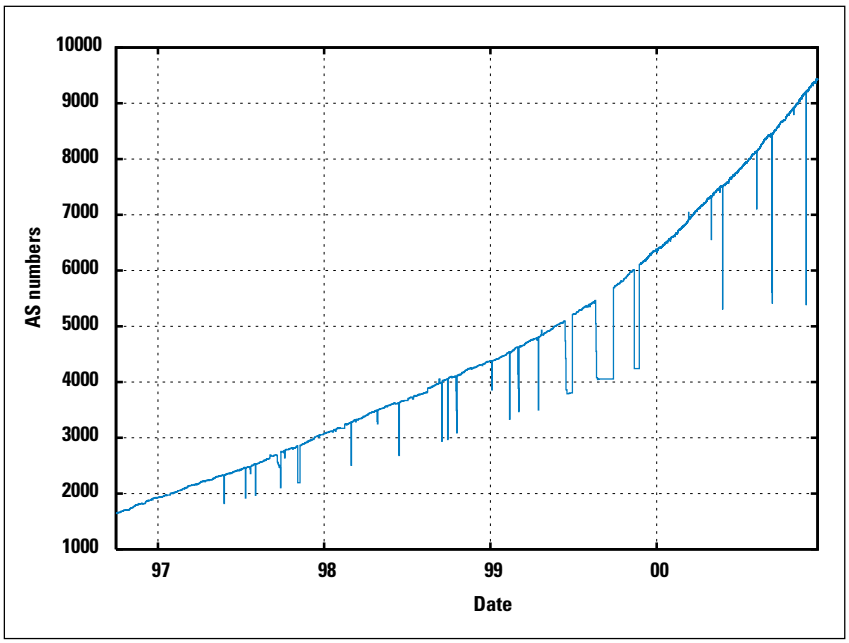
Related Measurements Derived from BGP Table

The level of analysis of the BGP routing table has been extended in an effort to identify the reasons for this resumption of exponential growth. Current analysis includes measuring the number of ASs in the routing system, and the number of distinct AS paths, the range of addresses spanned by the table, and the average span of each routing entry.

AS Number Consumption

Each network that is multihomed within the topology of the Internet and wishes to express a distinct external routing policy must use an AS to associate its advertised addresses with such a policy. In general, each network is associated with a single AS, and the number of ASs in the default-free routing table tracks the number of entities that have unique routing policies. There are some exceptions to this, including large global transit providers with varying regional policies, where multiple ASs are associated with a single network, but such exceptions are relatively uncommon. The trend of AS number deployment over the past four years is also exponential (Figure 6). The growth in the number of ASs can be correlated with the growth in the amount of address space spanned by the BGP routing table. At the end of 2000, the span of advertised addresses is growing at an annual rate of 7 percent, while the number of ASs is growing by 51 percent. Each AS is, on average advertising smaller address ranges. This points to increasingly finer levels of routing detail being announced into the global routing domain, a trend that causes some level of concern.

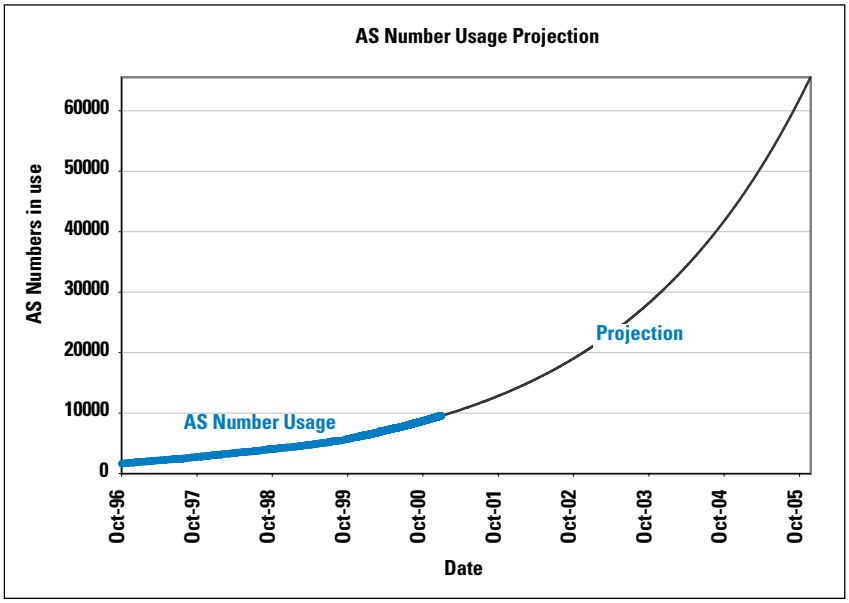
Figure 6: AS Number Deployment



This is a likely result of an increasingly dense interconnection mesh, where an increasing number of networks are moving from a single-homed connection into multihoming and peering. The spur for this may well be the declining unit costs of communications bearer services.

If this rate of growth continues, the 16-bit AS number set will be exhausted by late 2005 (Figure 7). Work is under way within the IETF to modify the BGP protocol to carry AS numbers in a 32-bit field[5]. Although the protocol modifications are relatively straightforward, the major responsibility rests with the operations community to devise a transition plan that will allow gradual transition into this larger AS number space.

Figure 7: AS Number Projections

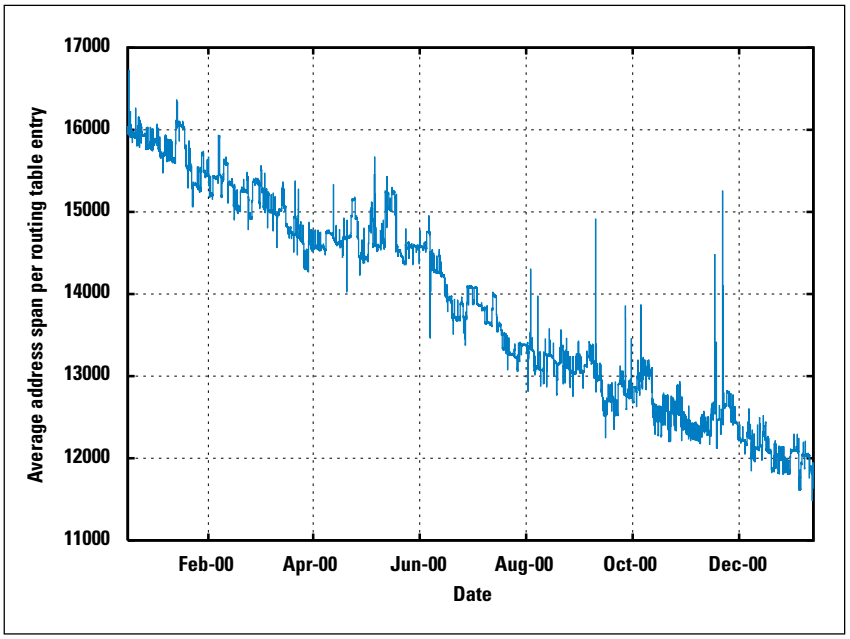


Average Prefix Length of Advertisements

The intent of CIDR aggregation was to support the use of large aggregate address announcements in the BGP routing table. To check whether this is still the case, researchers have tracked the average span of each BGP announcement for the past 12 months. The data indicates a decline in the average span of a BGP advertisement from 16,000 individual addresses in November 1999 to 12,100 in December 2000 (Figure 8). This corresponds to an increase in the average prefix length from /18.03 to /18.44. Separate observations of the average prefix length used to route traffic in operation networks in late 2000 indicate an average length of 18.1[8]. Again, this trend is cause for concern because it implies the increasing spread of traffic over greater numbers of increasingly finer forwarding-table entries. This, in turn, has implications for the design of high-speed core routers, particularly when extensive use is made of cached forwarding entries within the switching subsystem.

One potential scenario is that the size of the advertisement continues to decrease. With the widespread use of address translation gateway systems, such as NAT, and the continued concern over the finite nature of the IPv4 address pool, this is certainly a highly likely scenario. Projections of the average prefix length of advertisements using current trends in the number of BGP table entries and the total address span advertised in the BGP table indicate a lengthening of the average prefix length of advertisements by 1 bit length every 29 months. This has implications in the lookup algorithms used in routing design, depending on the space/time trade-offs used in the lookup algorithm design. This trend implies that either lookups need to search deeper through the prefix chain to find the necessary forwarding entry, requiring faster memory subsystems to perform each lookup, or the lookup table needs to be both larger and more sparsely populated, increasing the requirements for high-speed memory within the router forwarding subsystem.

Figure 8: Average Span of BGP Advertisement



Prefix Length Distribution

In addition to looking at the average prefix length, the analysis of the BGP table also includes an examination of the number of advertisements of each prefix length.

An extensive effort was introduced in the mid-1990s to move away from extensive use of the Class C space and to encourage providers to advertise larger address blocks. This has been reinforced by the address registries who have used provider allocation blocks of /19 and, more recently, /20. These measures were introduced when there were approximately 20,000 to 30,000 entries in the BGP table. It is interesting to note that five years later, of the 96,000 entries in the routing table, about 53,000 entries have a /24 prefix. In absolute terms, the /24 prefix set is the fastest-growing prefix set in the entire BGP table.

The routing entries of these smaller address blocks also show a much higher level of change on an hourly basis. Although a large number of BGP routing points perform route flap damping, there is still a very high level of announcements and withdrawals of these entries in this particular area of the routing table when viewed using a perspective of route updates per prefix length. Given that the number of these small prefixes is growing rapidly, there is cause for some concern that the total level of BGP flux, in terms of the number of announcements and withdrawals per second, may be increasing, despite the pressures from flap damping. This concern is coupled with the observation that, in terms of BGP stability under scaling pressure, it is not the absolute size of the BGP table that is of prime importance, but the rate of dynamic path recomputations that occur in the wake of announcements and withdrawals. Withdrawals are of particular concern because of the number of transient intermediate states that the BGP distance-vector algorithm explores in processing a withdrawal. Current experimental observations indicate a typical convergence time of about 2 minutes to propagate a route withdrawal across the BGP domain[7]. An increase in the density of the BGP mesh, coupled with an increase in the rate of such dynamic changes, does have serious implications in maintaining the overall stability of the BGP system as it continues to grow.

The registry allocation policies also have had some impact on the routing-table prefix distribution. The original registry practice was to use a minimum allocation unit of a /19, and the 10,000 prefix entries in the /17 to /19 range are a consequence of this policy decision. More recently, the allocation policy now allows for a minimum allocation unit of a /20 prefix, and the /20 prefix is used by about 4000 entries; in relative terms, this is one of the fastest-growing prefix sets.

The number of entries corresponding to very small address blocks (smaller than a /24), although small in number as a proportion of the total BGP routing table, is the fastest growing in relative terms. The number of /25 through /32 prefixes in the routing table is growing faster, in terms of percentage change, than any other area of the routing table. If prefix length filtering were in widespread use, the practice of announcing a very small address block with a distinct routing policy would have no particular beneficial outcome, because the address block would not be passed throughout the global BGP routing domain and the propagation of the associated policy would be limited in scope. The growth of the number of these small address blocks, and the diversity of AS paths associated with these routing entries, points to a relatively limited use of prefix-length filtering in today's Internet. In the absence of any corrective pressure in the form of widespread adoption of prefix-length filtering, the very rapid growth of global announcement of very small address blocks is likely to continue.

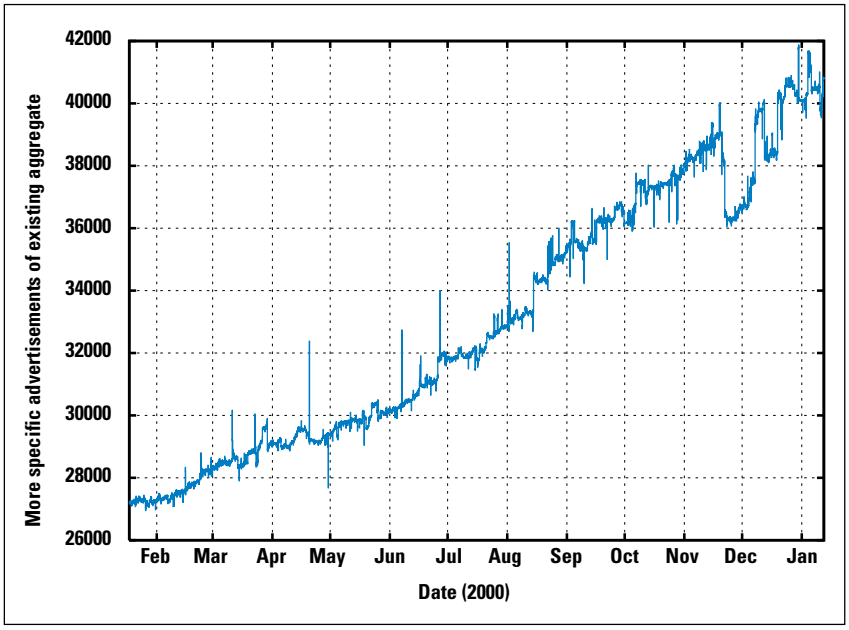
Aggregation and Holes

With the CIDR routing structure, it is possible to advertise a more specific prefix of an existing aggregate. The purpose of this more specific announcement is to punch a "hole" in the policy of the larger aggregate announcement, creating a different policy for the specifically referenced address prefix. Another use of this mechanism is not to promulgate a different connectivity policy, but to perform some rudimentary form of load balancing and mutual backup for multihomed networks. In this model, a network may advertise the same aggregate advertisement along each connection, but then advertise a set of specific advertisements for each connection, altering the specific advertisements such that the load on each connection is approximately balanced. The two forms of holes can be readily discerned in the routing table—while the approach of policy differentiation uses an AS path that is different from the aggregate advertisement, the load balancing and mutual backup configuration uses the same AS path for both the aggregate and the specific advertisements.

Although it is difficult to understand whether the use of such specific advertisements was intended to be an exception to a more general rule or that it was not intended to be within the original intent of CIDR deployment, there appears to be very widespread use of this mechanism within the routing table. Approximately 37,500 advertisements, or 37 percent of the routing table, is being used to punch policy holes in existing aggregate announcements (Figure 9). Of these, the overall majority of about 30,000 routes use distinct AS paths, so that once more we are seeing a consequence of finer levels of granularity of connection policy in a densely interconnected space.

Although long-term data is not available for the relative level of such advertisements as a proportion of the full routing table, the growth level does strongly indicate that policy differentiation at a fine level within existing provider aggregates is a significant driver of overall table growth.

Figure 9: More Specific Advertisements

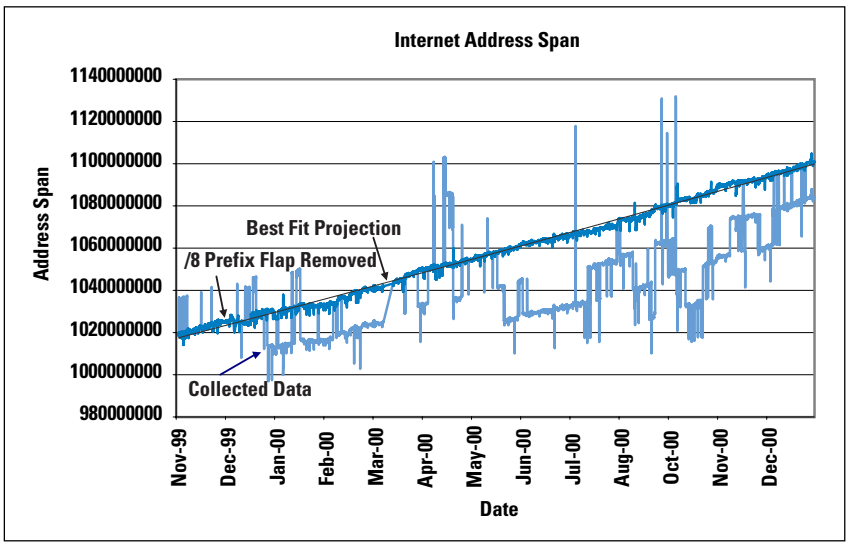


Address Consumption

A decade ago there were two major concerns over scaling of the Internet, and of the two, the consumption of address space was considered to be the more immediate and compelling threat to the continued viability of the network to sustain growth.

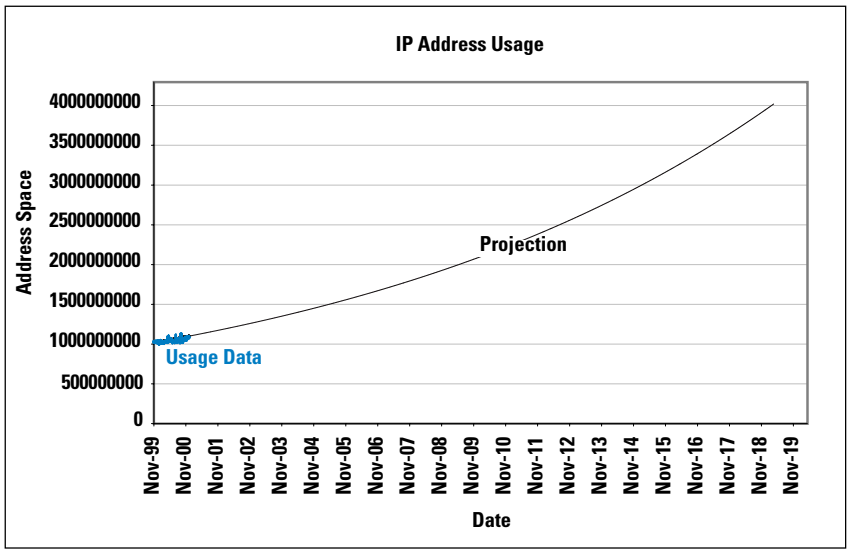
Within the scope of this exercise, it has been possible to track the total span of address space covered by BGP routing advertisements. Over the period from November 1999 until December 2000, the span of address space has grown from 1.02 billion addresses to 1.06 billion. However, numerous /8 prefixes are periodically announced and withdrawn from the BGP table, and if the effects of these prefixes are removed, the final value of addresses spanned by the table is approximately 1.09 billion addresses (Figure 10).

Figure 10: Total Address Space



This is an annual growth rate of a little less than 7 percent, and at that rate of address deployment, the IP Version 4 address space will be able to support another 19 years of such growth (Figure 11). Compared to the 42-percent growth in the number of routing advertisements, it would appear that much of the growth of the Internet in terms of growth in the number of connected devices is occurring behind various forms of NATs. In terms of solving the perceived finite nature of the address space identified just under a decade ago, the Internet appears so far to have embraced the approach of using NATs, irrespective of their various perceived functional shortcomings[3]. This observation also supports the observed increase of smaller address fragments supporting distinct policies in the BGP table, because such small address blocks encompass arbitrarily large networks located behind one or more NAT gateways.

Figure 11: Address Space Projection



Anomalies

A common space such as the inter-provider domain is not actively managed by any single entity, and various anomalies appear in the routing table from time to time.

One notable event occurred in late 1997, when some large prefixes were deconstructed into a massive set of /24 prefixes and this set was inadvertently passed into the inter-provider BGP domain. The BGP table graphs show a sudden upswing in the number of routing table entries from 50,000 entries to about 78,000 entries. It could have been higher, except that a commonly used routing hardware platform at the time ran into table memory exhaustion at that number of table entries, and further promulgation of additional routing entries ceased. Numerous other anomalies also exist in the table, including the presence of a /31 prefix and several hundred /32 prefixes.

Although many of these anomalies can be attributed to configuration errors of various forms, the underlying observation is that there are no universally used strong filters on what can broadcast into the BGP routing space. Considering the distributed nature of this table and the critical role that it plays in supporting the global Internet, this can be considered a significant current vulnerability. One potential response is to make more use of authentication measures. A validity check could be a precondition to accepting any route advertisement, allowing the receiver of the advertisement a means to check that the origin AS intended to advertise this route. This would create greater resiliency against inadvertent leaks of large sets of advertisements into the broader inter-domain space. It would also improve the resiliency of the BGP domain against some forms of deliberate attack.

Conclusions

There are strong parallels between the BGP routing space and the condition commonly referred to as "The Tragedy Of The Commons." The BGP routing space is simultaneously everyone's problem, because it impacts the stability and viability of the entire Internet, and no one's problem, in that no single entity can be considered to manage this common resource.

In other common resource domains, when the value of the resource is placed under threat because of damaging exploitative practices, the most typical form of corrective action is through the imposition of a consistent set of policies and practices intended to achieve a particular outcome. The vehicle for such an imposition of policies and practices is most commonly that of regulatory fiat. In a globally distributed space such as the BGP table, it is a challenging task to identify the source and authority of such potential regulatory activity.

Multihomed Small Networks

It would appear that one of the major drivers of the recent growth of the BGP table is that of small networks multihoming with numerous peers and numerous upstream providers. In the appropriate environment where numerous networks are in relatively close proximity, using peer relationships can reduce total connectivity costs, as compared to using a single upstream service provider. Equally significantly, multihoming with numerous upstream providers is seen as a means of improving the overall availability of the service. In essence, multihoming is seen as an acceptable substitute for upstream service resiliency.

This has a potential side effect: When multihoming is seen as a preferable substitute for upstream provider resiliency, the upstream provider cannot command a price premium for proving resiliency as an attribute of the provided service, and, therefore, has little incentive to spend the additional money required to engineer resiliency into the network. The actions of the multihomed network clients then become self-fulfilling.

One way to characterize this behavior is that service resiliency in the Internet is becoming the responsibility of the customer, not the service provider.

In such an environment resiliency still exists, but rather than being a function of the bearer or switching subsystem, resiliency is provided through the function of the BGP routing system. The question is not whether this is feasible or desirable in the individual case, but whether the BGP routing system can scale adequately to continue to undertake this role.

A Denser Interconnectivity Mesh

The decreasing unit cost of communications bearers in many part of the Internet is creating a rapidly expanding market in exchange points and other forms of inter-provider peering. The deployment model of a single-homed network with a single upstream provider is rapidly being supplanted by a model of extensive interconnection at the edges of the Internet. The underlying deployment model assumed by CIDR assumed a different structure, more akin to a strict hierarchy of supply providers. The business imperatives driving this denser mesh of interconnection in the Internet are irresistible, and the casualty in this case is the CIDR-induced dampened growth of the BGP routing table.

Traffic Engineering via Routing

Further driving this growth in the routing table is the use of selective advertisement of smaller prefixes along different paths in an effort to undertake traffic engineering within a multihomed environment. Although considerable effort is being undertaken to develop traffic-engineering tools within a single network using *Multiprotocol Label Switching* (MPLS) as the base flow management tool, inter-provider tools to achieve similar outcomes are considerably more complex when using such switching techniques. At this stage, the only tool being used for inter-provider traffic engineering is that of the BGP routing table, further exacerbating the growth and stability pressures being placed on the BGP routing domain.

The effects of CIDR on the growth of the BGP table have been outstanding, not only because of their initial impact in turning exponential growth into a linear growth trend, but also because CIDR was effective for far longer than could have been reasonably expected in hindsight. The current growth factors at play in the BGP table are not easily susceptible to another round of CIDR deployment pressure within the operator community. It may well be time to consider how to manage a BGP routing table that has millions of small entries, rather than the expectation of tens of thousands of larger entries.

We started this journey over ten years ago when considering the scaling properties of addressing and routing. It is perhaps fitting that we tie the two concepts back together again as we consider the future of the BGP inter-provider routing space. The observation that the BGP growth pressures are largely due to an uptake in multihoming and the associated advertisement of discrete connectivity policies by increasingly smaller networks at the edge of the network has a corollary for address allocation policy. In such a ubiquitous environment of multihomed networks, we will also need to review how address blocks are allocated to network providers, because the concept of provider-based address allocation that assumes a relatively strict hierarchical supply structure is becoming less and less relevant in today's Internet.

References

[1] D. Clark, L., Chapin, V. Cerf, R. Braden, R. Hobby, "Towards the Future Internet Architecture," RFC 1287, December 1991.

[2] V. Fuller, T. Li, J. Yu, and K. Varadhan, "Supernetting: an Address Assignment and Aggregation Strategy," RFC 1338, June 1992.

[3] T. Hain, "Architectural Implications of NAT," RFC 2993, November 2000.

[4] T. Bates, "The CIDR Report," updated weekly at: **http://www.employees.org/~tbates/cidr-report.html**

[5] E. Chen, Y. Rekhter, "BGP Support for Four-Octet AS Number Space," work in progress, currently published as an Internet Draft: **draft-chen-as4bytes-00.txt**, November 2000.

[6] "BGP Table Report" updated hourly at **http://www.telstra.net/ops/bgp**

[7] C. Labovitz, A. Ahuja, "The Impact of Internet Policy and Topology on Delayed Routing Convergence—Update to This Work," ISMA Winter 2000 Workshop, CAIDA, December 2000.

[8] Peter Lothberg, personal communication.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Scientist in the Internet area for Telstra. He is also a member of the Internet Architecture Board, and is the Secretary of the Internet Society Board of Trustees. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@telstra.net**

LAN QoS

by William Stallings

A typical organization's on-premise network configuration has multiple *Local-Area Networks* (LANs) connected by bridges or Layer 2 switches. The LANs may all be of one type (for example, Ethernet) or may be of mixed types (for example Ethernet, Token Ring, wireless). In either case, the issue of *Quality of Service* (QoS) arises.

User Priority and Access Priority

The first attempt to deal with LAN QoS in a standardized fashion appears in the original version of IEEE 802.1D, which is a specification that defines the protocol architecture for bridges and Layer 2 switches, which operate at the *Media Access Control* (MAC) level. IEEE 802.1D deals with the interconnection of LANs with the same MAC protocol and with LANs with different MAC protocols. In addition to passing MAC frames from one LAN to another across the bridge, the bridge is able to pass parameters from software that controls the incoming port to the software that controls the outgoing port. Two of these parameters are *user_priority* and *access_priority*.

The *user_priority* and *access_priority* parameters relate to the problem of how to handle priorities. In the case of IEEE 802.3 (Ethernet) and 802.11 (wireless LAN), priority is not supported. Other 802 LAN types support up to eight levels of priority. The *user_priority* value provided to the MAC- layer entity at the incoming port is derived from the incoming MAC frame; in the case of an incoming frame with no priority value, a value of *unspecified* is used. The *user_priority* value issued to the MAC entity at the outgoing port is to be placed in the outbound MAC frame for LAN types that provide a priority field. The *access_priority* refers to the priority used by a bridge MAC entity to access a LAN for frame transmission. We may not want the *access_priority* to be equal to the *user_priority* for several reasons:

- A frame that must go through a bridge has already suffered more delay than a frame that does not have to go through a bridge; therefore, we may wish to give such a frame a higher access priority than the requested user priority.
- It is important that the bridge not become a bottleneck. Therefore, we may wish to give all frames being transmitted by a bridge a relatively high priority.

The rules for handling priorities can now be summarized. The *user_priority* is determined from the priority field of the incoming frame and placed in the priority field of the outbound frame. Priorities are not used to transmit 802.3 and 802.11 MAC frames, and the frames themselves have no priority field. Therefore, if the outbound frame is 802.3 or 802.11, any incoming priority field (from a frame that has such a field) is ignored. If the incoming frame is 802.3 or 802.11 and the outbound frame requires a priority field, then the priority field in the outbound frame is set to a default *user_priority* value. If both incoming and outbound frames carry a priority field, then the priority field in the outbound MAC frame is set equal to the priority field in the inbound MAC frame.

The *access_priority* is also determined from the priority field of the incoming frame. For incoming 802.3 and 802.11 frames, a *user_priority* of 0 (lowest priority) is assumed. Table 1 shows the access priorities assigned to outgoing MAC frames for each of the LAN types, as a function of incoming user priority value. For 802.3 and 802.11, there is no access priority mechanism and, therefore, a priority of 0 is used. For 802.4 and 802.6, there are eight available access priorities, so the incoming user priority is mapped to the outgoing access priority using equality. IEEE 802.12 permits only two priority levels; half of the possible user priority values are mapped into each of these levels. For the two Token Ring types (802.5 and Fiber Distributed Data Interface [FDDI]), although eight priority levels are available, the highest priority (level 7) is not used in bridge forwarding. The reason for this restriction is that the token-passing protocol reserves priority 7 for its use in transmitting frames needed to manage the token-passing process, such as recovering from a frame loss.

Table 1: Outbound Access Priorities

User Priority	Outbound Access Priority per MAC Method						
	802.3	802.4	802.5	802.6	802.11	802.12	FDDI
0	0	0	0	0	0	0	0
1	0	1	1	1	0	0	1
2	0	2	2	2	0	0	2
3	0	3	3	3	0	0	3
4	0	4	4	4	0	4	4
5	0	5	5	5	0	4	5
6	0	6	6	6	0	4	6
7	0	7	6	7	0	4	6

802.3 = CSMA/CD
802.4 = Token bus
802.5 = Token ring
802.6 = DQDB (Distributed Queue, Dual Bus) MAN
802.11 = Wireless LAN
802.12 = Demand priority (100VG-AnyLAN)
FDDI = Fiber Distributed Data Interface (token ring)

Traffic Classes

These rules, summarized in Table 1, are effective in communicating a priority requested by a user and in obtaining access to a LAN in competition with other devices also attempting to transmit on that LAN. However, the rules do not directly provide guidance concerning the relative priority with which frames are to be handled by a bridge. For example, consider a bridge connected to a Token Ring on one side and an Ethernet on the other, and suppose that the bridge receives a large volume of traffic from the Token Ring so that a number of frames are buffered waiting to be transmitted onto the Ethernet. Should the bridge transmit these frames in the order in which they were received, or should the bridge account for the user priority of all waiting frames in determining which frame to transmit next? Consideration of this issue led to the development of a new concept, *traffic class,* which is incorporated in the 1998 version of IEEE 802.1D. This new material is sometimes referred to as 802.1p in the literature. This was the designation when the traffic-class standard was in draft form. In the 802 scheme, a lowercase letter refers to a supplement to an existing standard and an uppercase letter refers to a base standard. Thus 802.1D is a base standard defining bridge operation, and 802.1p is a supplement to the earlier version of 802.1D. With the publication of the 1998 version, the traffic-class supplement was incorporated into 802.1D, and the designation 802.1p is no longer used.

The goal of the traffic-class addition to 802.1D is to enable Layer 2 switches and bridges to support time-critical traffic, such as voice and video, effectively. In the remainder of this article, we begin with an overview of the use of traffic classes in bridges. Next, we examine the mapping of user priorities into traffic classes. Finally, we look at the larger issue of QoS in an internet that includes bridges as well as routers and other Layer 3 switches.

The 1998 version of IEEE 802.1D distinguishes three concepts:

- *User priority:* The user priority is a label carried with the frame that communicates the requested priority to downstream nodes (bridges and end systems). Typically, the user priority is not modified in transit through bridges, unless a mapping is needed for the use of a different number of priority levels by different MAC types. Thus, the user priority has end-to-end significance across bridged LANs.
- *Access priority:* The access priority is used, on LANs that support priority, to compete for access to the shared LAN with frames from other devices (end systems and other bridges) attached to the same LAN. For example, the token-passing discipline in a Token Ring network enables higher-priority frames to gain access to the ring ahead of lower-priority frames when frames from multiple stations are waiting to gain access. When both the incoming and outbound LAN are of the same MAC type, the bridge assigns an access priority equal to the incoming user priority. Otherwise, the bridge must perform a mapping as defined in Table 1.

- *Traffic class:* A bridge can be configured so that multiple queues are used to hold frames waiting to be transmitted on a given outbound port, in which case the traffic class is used to determine the relative priority of the queues. All waiting frames at a higher traffic class are transmitted before any waiting frames of a lower traffic class. As with access priority, traffic class is assigned by the bridge on the basis of incoming user priority.

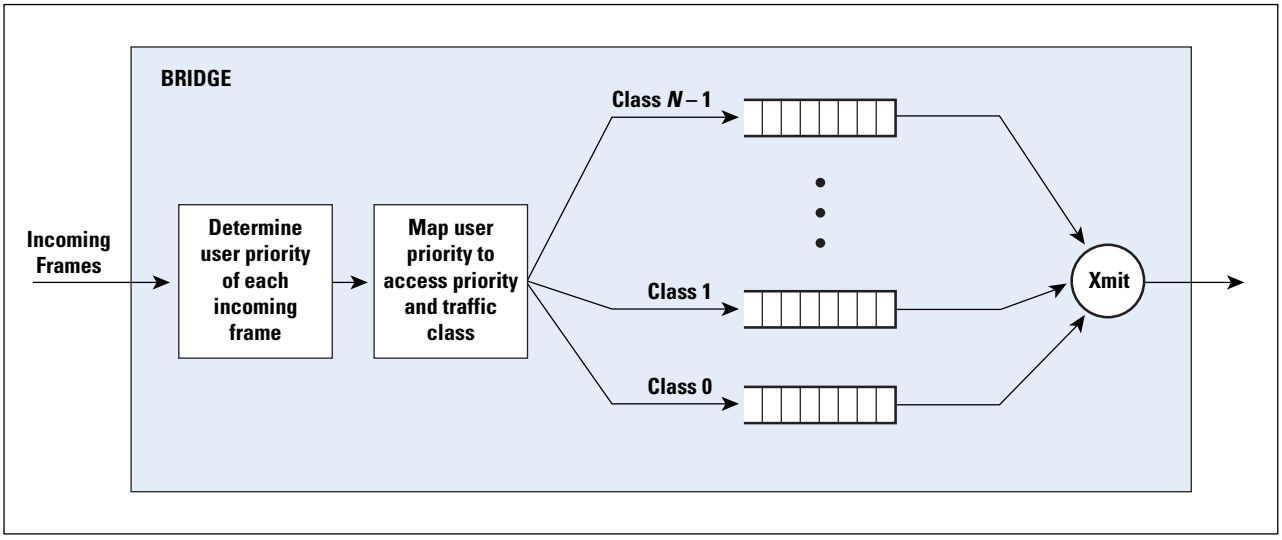
The significance of traffic classes can be seen by recognizing that a frame experiences two types of delay at a bridge:

- *Queuing delay:* The time that a frame waits until it becomes first in line for transmission on the outbound port. This delay is determined by the queuing discipline used by the bridge. The simplest scheme is first-in, first-out (FIFO). Traffic classes permit more sophisticated schemes.
- *Access delay:* The delay that a frame experiences waiting for permission to transmit on the LAN, in competition with frames from other stations attached to the same LAN. This delay is determined by the MAC protocol used (for example Token Ring, Carrier Sense Multiple Access Collision Detect [CSMA/CD]).

The total delay experienced by a frame at a bridge is the sum of its queuing delay and its access delay.

Figure 1 illustrates the mechanism used to support traffic classes at a bridge. A bridge may support up to eight different traffic classes on any outbound port by implementing up to eight distinct queues, or buffers, for that port. A traffic-class value is associated with each queue, ranging from a low of 0 to a high of $N - 1$, where N is the number of traffic classes associated with a given outbound port ($N \leq 8$).

Figure 1: IEEE 802.1 D Traffic Class Operation



On a given output port with multiple queues, the rules for transmission follow:

1. A frame may be transmitted from a queue only if all queues corresponding to numerically higher values of traffic class are empty. For example, if there is a frame in queue 0, it can be transmitted only if all the other queues at that port are currently empty.
2. Within a given queue, the order of frame transmission must satisfy the following: The order of frames received by this bridge and assigned to this outbound port shall be preserved for:
    - Unicast frames with a given combination of destination address and source address
    - Multicast frames for a given destination address

In practice, a FIFO discipline is typically used. Thus, a strict priority mechanism is used. It follows that during times of congestion, lower-priority frames may be stuck indefinitely at a bridge that devotes its resources to moving out the higher-priority frames.

Mapping of User Priority to Traffic Class

IEEE 802.1D provides guidance on the mapping of user priorities into traffic classes. Table 2 shows the recommended mapping. We can make two comments immediately:

1. The mapping is based on the user priority associated with the frame, which, as was mentioned earlier, has end-to-end significance. However, the 802.3 and 802.11 frame formats do not include a priority field, meaning that this end-to-end information could be lost. To address this issue, the bridge is able to reference the priority field contained in a tag header defined in IEEE 802.1Q, which deals with virtual LANs. The 802.1Q specification defines a tag header of 32 bits that is inserted after the source and destination address fields of the frame header. This tag header includes a 3-bit priority field. Thus, if 802.1Q is in use by Ethernet and wireless LAN sources, a user priority can be defined that stays with the frame from source to destination.
2. Outbound ports associated with MAC methods that support only a single access priority, such as 802.3 and 802.11, can support multiple traffic classes. Recall that the traffic class deals with queuing delay, while the access priority deals with access delay.

To understand the reason for the mappings recommended in Table 2, we need to consider the types of traffic that are associated with each traffic class. IEEE 802.1D provides a list of traffic types, each of which can benefit from simple segregation from the others. In descending importance, these types include:

- Network control (7): Both time critical and safety critical, consisting of traffic needed to maintain and support the network infrastructure, such as routing protocol frames.

- Voice (6): Time critical, characterized by less than 10-ms delay, such as interactive voice.
- Video (5): Time critical, characterized by less than 100-ms delay, such as interactive video.
- Controlled load (4): Non-time-critical but loss sensitive, such as streaming multimedia and business-critical traffic. A typical use is for business applications subject to some form of reservation or admission control, such as capacity reservation per flow.
- Excellent effort (3): Also non-time-critical but loss sensitive, but of lower priority than controlled load. This is a best-effort type of service that an information services organization would deliver to its most important customers.
- Best effort (2): Non-time-critical and loss insensitive. This is LAN traffic handled in the traditional fashion.
- Background (0): Non-time-critical and loss insensitive, but of lower priority than best effort. This type includes bulk transfers and other activities that are permitted on the network but that should not impact the use of the network by other users and applications.

Only seven traffic types are defined in IEEE 802.1D. The standard leaves as spare an eighth type, which could be used for traffic of more importance than background but less importance than best effort. The numbers in parentheses in the preceding list are the traffic-class values corresponding to each traffic type if there are eight queues and hence eight traffic classes available at a given output port.

Table 2: Recommended User Priority to Traffic Class Mapping

		Number of Available Traffic Classes							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
User Priority	0 (default)	0	0	0	1	1	1	1	2
	1	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	1
	3	0	0	0	1	1	2	2	3
	4	0	1	1	2	2	3	3	4
	5	0	1	1	2	3	4	4	5
	6	0	1	2	3	4	5	5	6
	7	0	1	2	3	4	5	6	7

We can now address the issue of the mapping between user-priority and traffic-class value. If eight traffic class values are available (eight queues at this output port), the obvious mapping would be equality; that is, a user priority of K would map into traffic class K for $0 \leq K < 7$. This obvious mapping is not desirable because of the treatment of default priorities. For 802.3 and 802.11, which do not use priorities, the de-

fault user priority is 0. For other MAC types, such as 802.5, if the user does not specify a priority, the MAC level assigns a default value of 0. The 802.1D standard points out that using a different default value would result in some confusion and probably a lack of interoperability. However, the logical default traffic type is best effort. The solution proposed by 802.1D is to map a user priority of 0 to traffic-class value 2. When there are eight traffic class values available, then user-priority values 1 and 2 map to traffic-class values 0 (background) and 1 (spare value), respectively.

This solution is reflected in Table 2, which shows the mapping of user priority to traffic class when there are eight available traffic classes. The table also shows the mapping when there are fewer traffic classes. To understand the entries in this table, we need to consider the way in which 802.1D recommends grouping traffic types when fewer than eight queues are configured at a given output port. Table 3 shows this grouping. The first row in the table shows that if there is only one queue, then all traffic classes are carried on that queue. This is obvious. If there are two queues (second row), 802.1D recommends assigning network control, voice, video, and controlled load to the higher-priority queue, and excellent effort, best effort, and background to the lower-priority queue. The reasoning supplied by the standard follows: To support a variety of services in the presence of bursty best-effort traffic, it is necessary to segregate time-critical traffic from other traffic. In addition, further traffic that is to receive superior service and that is operating under admission control also needs to be separated from the uncontrolled traffic. The allocation of traffic types to queues for the remaining rows of the table can be explained similarly.

Table 3: Suggested Traffic Types

		Traffic Types							
Number of Queues	1	**BE** (EE, BK, VO, CL, VI, NC)							
	2	**BE** (EE, BK)				**VO** (CL, VI, NC)			
	3	**BE** (EE, BK)				**CL** (VI)		**VO** (NC)	
	4	**BK**		**BE** (EE)		**CL** (VI)		**VO** (NC)	
	5	**BK**		**BE** (EE)		**CL**	**VI**	**VO** (NC)	
	6	**BK**		**BE**	**EE**	**CL**	**VI**	**VO** (NC)	
	7	**BK**		**BE**	**EE**	**CL**	**VI**	**VO**	**NC**
	8	**BK**	**—**	**BE**	**EE**	**CL**	**VI**	**VO**	**NC**
		1	**2**	**0**	**3**	**4**	**5**	**6**	**7**
		User Priority							

Note: In each entry, the boldface type is the traffic type that has driven the allocation of types to classes.

BK = Background
BE = Best Effort
EE = Excellent Effort
CL = Controlled Load
VI = Video (<100 ms latency and jitter)
VO = Voice (<10 ms latency and jitter)
NC = Network Control

Internet Traffic Quality of Service

The user-priority and traffic-class concepts enable MAC-level bridges and Layer 2 switches to implement a traffic-handling policy within a bridged collection of LANs that gives preference to certain types of traffic. These concepts are needed because these bridges and switches cannot see "above" the MAC layer and hence cannot recognize or utilize QoS indications in higher layers such as IP. However, it is often the case that traffic from a bridged set of LANs must cross Wide-Area Networks (WANs) that make use of QoS functionality. An example of this is an ATM network, which provides for user-specified QoS. Another example is an IP-based internet, which can provide IP-level QoS. Some means is needed for mapping between traffic classes and QoS for such configurations. This is an evolving area of technology and standardization, but a general picture can be provided.

In the case of IP-based internets, the IP *Type-of-Service* (ToS) field provides a way to label traffic with different QoS demands. The ToS field is preserved along the entire path from source to destination through, potentially, multiple routers. Fortunately, the mapping from traffic class to ToS is straightforward. The ToS field includes a 3-bit Precedence subfield. A router connecting a LAN to an internet can be configured to read the Layer 2 Traffic-Class field and copy that into the ToS Precedence field in one direction, and copy the 3-bit Precedence field into the User Priority field in the other direction.

In the case of an ATM connection, a bridge or Layer 2 switch might be connected to a LAN on one side and an ATM network on the other, using the ATM network to link to other remote LANs. For local LAN traffic arriving at the bridge, the bridge must match the user priority level with the appropriate ATM service class and other ATM parameters. For this purpose, the bridge can consult a mapping table whose settings have been predefined through the policy controls of network management software. An appropriate virtual connection is used to carry the traffic. If the traffic exits the ATM network at another LAN, the bridge on that end can map incoming traffic from each virtual connection into the appropriate traffic class and user priority.

References

A more detailed discussion of bridges, Layer 2 switches, and IEEE 802.1D is contained in [1]. The IEEE 802.1 working group is at **`http://grouper.ieee.org/groups/802/1/index.html`**.

[1] Stallings, W., *Local and Metropolitan Area Networks, Sixth Edition,* Prentice Hall, 2000.

WILLIAM STALLINGS is a consultant, lecturer, and author of over a dozen books on data communications and computer networking. He has a PhD in computer science from M.I.T. His latest book is *Local and Metropolitan Area Networks, Sixth Edition* (Prentice Hall, 2000). His home in cyberspace is **`WilliamStallings.com`** and he can be reached at **`ws@shore.net`**

Book Reviews

E-mail Books

Essential Email Standards: RFCs and Protocols Made Practical by Pete Loshin, ISBN 0-471-34597-0, John Wiley & Sons, Inc., 2000. **www.wiley.com**

Internet Email Protocols: A Developer's Guide, by Kevin Johnson, ISBN 0-201-43288-9, Addison-Wesley, 1999. **www.awl.com**

Deciding when to write a book about an exciting new technology is pretty easy. At first issuance of the standards for it, or emergence of a market for it, out will come the requisite texts. In 1993, when the commercial Internet started to surface, Marshall Rose produced *The Internet Message: Closing The Book With Electronic Mail* [Prentice Hall, 1993]; it's an excellent introduction to the core e-mail services. As the market grew, Rose and David Strom issued a more operations-oriented effort, *Internet Messaging; From Desktop to the Enterprise* [Prentice Hall, 1998]. For anyone serious about e-mail technology and operations, it remains required reading.

But what about straight technology exposition when the standards that have been in use for more than 20 years keep getting modified? In the case of Internet mail, this dilemma has been exacerbated by an extended recent effort to coalesce documentation for the service, compiling and clarifying the contents of many independent *Internet Engineering Task Force* (IETF) documents into two, one for the transfer service and one for the mail object definition. The best time to publish a book on the subject would be at the issuance of the two revisions. Unfortunately, the IETF effort has taken perhaps 3 years longer than expected, and Wiley and Addison-Wesley decided the market needed these books earlier. Hence the authors were faced with a juggling act, referring to original specifications, with appropriate nods to the new—but unstable—drafts.

Comprehensive Introductions

This tactical caveat notwithstanding, Peter Loshin's *Essential Email Standards: RFC and Protocols Made Practical* and Kevin Johnson's *Internet Email Protocols: A Developer's Guide* are credible and reasonably thorough. They introduce the reader to the technical details of Internet mail. Loshin adds detail about the standards culture that produced the specification. Johnson adds a bit of programming detail. No textbook on a technology should be used as the primary reference by someone building products, of course; and these are no exception. These are comprehensive introductions.

With such books, the criteria are simple. I look for helpful overall organization, clear language, and accurate content. These two books qualify. They summarize and restate the basic descriptions of services, data formats, protocol commands, and responses associated with the various standards.

Extra points are assigned when a book comes with commentary that provides some insight into the technical philosophy or operational pragmatics of the technology. Pleasantly, both books have a bit of these extras, too. Such texts typically also have minor technical errors; and these fit that profile, too. Since the reader is not using the book as an implementation reference, the occasional, small errors cause no harm.

Loshin's effort is 330 hardbound pages. Johnson's is about a third longer, softbound. Both books cover the core services of *Submit, Simple Mail Transfer Protocol Service Extensions* (ESMTP), the *Post Office Protocol* (POP), the *Internet Mail Access Protocol* (IMAP), RFC 822, and *Multipurpose Internet Mail Extensions* (MIME), that is, posting, relaying, and accessing e-mail, as well as description of the e-mail object. Both also discuss security. Submit is a recent spinoff from SMTP, for local user-relay posting. It began as a clone of ESMTP, but on a different port, and will permit service-to-service relaying functionality to diverge from the local, first-hop posting process. The market treats POP and IMAP as essentially competitive protocols, and both books explain their details adequately. I wish they had made the very simple architectural point that POP does last-hop delivery, to the user's PC-based message store, whereas IMAP is primarily for user access to a message store on a remote system. That is, one is for simply dumping an entire message queue onto the waiting user machine, whereas the other is for ongoing and interaction with portions of message data. On the other hand, an example of Loshin's extra credit is for noting that ISPs are reticent to support IMAP—they have not yet discovered that they could make money being a small business' back-office data store—whereas corporations like IMAP because it is an open standard that permits replacing proprietary workgroup message stores.

E-mail address resolution can be a bit tricky, requiring general understanding of the Domain Name Service and specific cleverness with MX "routing" records. Johnson devotes a useful, but very terse 2+ pages to the topic. Loshin allocates a 8+ pages.

Security

As with every other aspect of Internet standards making, e-mail security is problematic because no IETF-originated security protocol has yet gained wide deployment and use. Oddly continuing the peculiarity of security as a topic, both books are a little off-beat, albeit differently. Johnson provides a relatively extensive introduction to basic security technology, including descriptions of various algorithms, as well as a listing of the types of security attacks that can occur. He also discusses enhancements to the basic e-mail protocols for invoking security mechanisms. Loshin has a more functional systems orientation concerning overall e-mail security architecture. Although Loshin does not usually spend much time on ancient history, for some reason in this chapter he discusses two IETF failures of *Privacy Enhanced Mail* (PEM) and *MIME Object Security Services* (MOSS).

Both discuss *Pretty Good Privacy* (PGP), and PGP is certainly the long-standing popular choice among the technical community. Johnson discusses it in some detail; Loshin's coverage is minimal. *Secure MIME* (S/MIME) has support from major industry software vendors. Loshin treats it equally as tersely as he treats PGP. Johnson barely mentions it.

Standards

Loshin spends the first 50 pages on the Internet standards community, process, and documents. His book also covers Internet News (NNTP) and some work involving standard data for business cards (vCard) and calendaring and scheduling (iCalendar). Besides being interesting topics, these last two were probably included because the Internet Mail Consortium acquired intellectual property rights to the precursor work and highlights the topics on its Web page. Loshin also ends with a chapter about the future, where he adds the topics of instant messaging and message tracking, based on continuing IETF standards work. An included CD-ROM contains a copy of the book, with Web links to cited documents such as RFCs.

Johnson's forays beyond the core services discuss messaging filtering and mailing-list processing, UNIX file issues, and generic, terse descriptions of some programming languages. He also discusses the *Internet Message Support Protocol* (IMSP), the *Application Configuration Access Protocol* (ACAP), and the *Lightweight Directory Access Protocol* (LDAP), protocols for accessing user configuration data. Obviously he intends that the reader take seriously the "Developer's" reference in the book title.

The Differences

Perhaps it is the programmer's orientation that caused Johnson to be so thorough with his discussions. This includes discussion of e-mail protocols that are not standards and not in use. Loshin in far more selective and reflective. And therein lies the easy distinction between the two efforts. Loshin gives an understanding of a portion of application space, providing the basic technical details tidbits of useful insight. Johnson is more mechanical and more detailed; in effect he chooses to be less selective and more detailed in what he dumps on the reader, letting the reader decide what is useful.

—Dave Crocker, Brandenburg InternetWorking
dcrocker@brandenburg.com

Wireless and Mobile Network Architectures

Wireless and Mobile Network Architectures, by Yi-Bing Lin and Imrich Chlamtac, ISBN 0-471-39492-0, John Wiley & Sons, 2001.

Paging through this book, my first impressions are that it uses very little math and that it is a comprehensive standards-based overview of practical wireless systems. The authors' multidisciplinary tack—systems, networks, and services—is evidenced by their conceptual approach to engineering design issues and their straightforward explanations of implementation issues. The primary concern of the book as a whole is: "How does it all fit together?"

Organization

The authors divide the book into five major units. The first three units covered their topics well and enhanced my understanding of wireless communications. However, the final two units fell short of my expectations. Coverage of the *Wireless Application Protocol* (WAP) and other up-and-coming issues in wireless networking was patchy and unbalanced.

The "PCS Network Management" section provides an overview of the concepts, definitions, and procedures used in current wireless network implementations. Basic roaming concepts including handoff geometry, detection, and queuing schemes are briefly discussed. An understanding of foundational engineering concepts is assumed as the authors provide detailed algorithmic descriptions of hard and soft handoff message flows.

The "IS-41 Mobile Systems" section provides an introductory overview of *Signaling System 7* (SS7) as a supporting protocol for the IS-41 mobile communications protocol. The importance of integration between these two protocols is presented in practical example format. Intersystem handoff and authentication techniques applicable to IS-41 are then discussed. Included in this section is a functional overview of network signaling for *Personal Access Communications* (PACS) networks as related to IS-41. However, a general understanding of the PACS radio system is assumed.

GSM

Global System for Mobile Communication (GSM) systems are the largest focus of this book. A full ten chapters are dedicated to the concepts and applications of this technology. The section appropriately starts with a high-level overview of the GSM system architecture and moves through mobility management and roaming. Here, the authors present several alternative roaming concepts aimed at reducing the cost of roaming service. Additionally, mobile number portability mechanisms and costs are also addressed. Likewise, significant attention is given to the technical aspects of GSM networks and their integration with data networks. Full chapters are dedicated to describing the GSM network signaling software platform (MAP), operations, administration, and management functions, Voice over IP integration, and General Packet Radio Service over GSM.

For the student, *Wireless and Mobile Network Architectures* is a capstone reference that ties together several courses worth of technical information with a practical focus toward real-world applications. For professional IT managers, engineers, and software developers, it is a practical and handy tutorial for getting up-to-speed on second-generation wireless and mobile technologies.

Questions

Each chapter ends with a set of very open-ended and thought-provoking analysis and design questions. Reading the chapter does not necessarily prepare you to do in-depth design; rather, you gain enough knowledge to sketch out a basic approach to solving the problem. It is obvious that many of the problems would require interdisciplinary collaboration to arrive at a tenable solution. Members of such a team would contribute different perspectives based on their particular area of expertise.

Worthwile Reference

This book assumes that the reader has mastered the basics in the field of mobile communications and is seeking to implement a practical design. Throughout the book are many easy-to-follow algorithmic or flow-chart explanations of various wireless communications processes. However, the information gleaned from these treatments tended to be more about functionality than design. Although a worthwhile reference, this book is by no means "all you need to design and implement a mobile services network."

—Albert C. Kinney
kinney@ieee.org

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

Fragments

ICANN Launches At-Large Membership Study

The *Internet Corporation for Assigned Names and Numbers* (ICANN) recently announced that it was commencing a comprehensive study of the structure of its At Large membership. The study will be conducted by an *At Large Membership Study Committee* that will make recommendations to ICANN's Board of Directors on how individuals can effectively participate in ICANN's policy development, deliberations and actions for technical coordination of the Internet.

Mr. Carl Bildt, the former Prime Minister of Sweden and noted United Nations envoy, will serve as Chair of the nine member Study Committee. An international statesman and information technology advisor, Bildt's current duties include Special Envoy of the Secretary General of the United Nations to the Balkans, Member of Parliament of Sweden, and Advisor and Board Member of several Internet and technology-related corporations.

"The Board's approval of the Study Committee and Carl Bildt's selection as Chair is a demonstration of ICANN's commitment to finding an effective way for the perspectives of individuals in every country to be heard and given due consideration," said Vint Cerf, Chairman of the ICANN Board of Directors. "We are extremely fortunate to have someone with Carl Bildt's international consensus building experience to lead this critical effort."

The Committee, which is chartered to seek input from all interested parties and to work toward a broad consensus on ICANN's At Large membership, will use multiple mechanisms for input, including public forums, mailing lists, and a public website. The Committee will encourage the participation of organizations and individuals worldwide, including the development of independent studies and analyses from across the global Internet's constituencies.

"ICANN's actions affect the whole world's Internet users, and I look forward to the challenging task of forging a consensus on the best method for representing this ever-growing constituency," said Bildt. "This will be an international cooperative effort, and I am counting on the participation of a diversity of Internet stakeholders that have an interest in ICANN to help us deliver a workable solution."

The Board invited Charles Costello and Pindar Wong to serve as the Committee's Vice-Chairs. Costello is director of the Carter Center's Democracy Program, and served as an outside monitor for ICANN's At Large elections held last year. Wong served as an ICANN Director and Vice Chairman of the Board during 1999–2000. He also is an active Internet policy leader in the Asia Pacific Region, and Chairman of Verifi (Hong Kong) Ltd., an Internet infrastructure consultancy. The remaining members of the committee are Pierre Dandjinou, Esther Dyson, Oliver Iteanu, Ching-Yi Lu, Thomas Niles, and Oscar Robles.

ICANN also announced the appointment of Denise Michel as the Committee's Executive Director. Ms. Michel has extensive experience in both private and public sector technology policy development, having served previously on the staff of the U.S. National Science Foundation, the American Electronics Association and the U.S. Department of Commerce. From 1993–1995, she was Sr. Technology Advisor to the Secretary of Commerce, Mr. Ronald Brown.

Following public comment, the Board also adopted a charter for the study to ensure a consistent base of expectations on the scope and details of the study committee's work. ICANN has posted the charter on its website at:
http://www.icann.org/committees/at-large-study/charter-22jan01.htm

For more information about the At Large Membership Study Committee, see: **http://www.atlargestudy.org/**

Correction

In the article "The Trouble with NAT," which appeared in our previous issue, a table of private nonroutable IP addresses taken from RFC 1918 was shown. The table contained an error, as pointed out by a couple of our readers. The correct table appears below.

Class	Private Address Range
A	10.0.0.0 … 10.255.255.255
B	172.16.0.0 … 172.31.255.255
C	192.168.0.0 … 192.168.255.255

Upcoming Events

The Internet Society (ISOC) will hold its annual conference INET in Stockholm, Sweden, June 5–8, 2001. For more information, see:
http://www.isoc.org/inet2001/

Just before INET, The Internet Corporation for Assigned Names and Numbers (ICANN) will hold its meeting in the same venue. The dates are June 1–4, 2001 and you can find more information at:
http://www.icann.org/calendar.htm

The Internet Engineering Task Force (IETF) will next meet in London, England, August 5–10. For more information, see:
http://www.ietf.org

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

June 2001 Volume 4, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

A user of a laptop computer "on the road" typically connects to the Internet in one of two ways. The oldest, and most common method, is to dial into an ISP's network and obtain an IP address using the *Point-to-Point Protocol* (PPP). The other method involves attaching the laptop to a local network (usually via Ethernet) and obtaining an IP address through the *Dynamic Host Configuration Protocol* (DHCP). The "local network" could be anything from the high-speed connection provided in some hotels, to an enterprise network at some corporation or other institution. In all cases, the IP address is fixed for the duration of the network session, and the routing of packets from the laptop back to its "home" network remains a relatively straight-forward task (ignoring NATs, firewalls and other complexities for the moment). Suppose however, the mobile computer is using a wireless connection and traveling between several networks over a short period of time. In this scenario one would still like to maintain network connectivity in a seamless manner. The IETF has been working on Mobile IP to address this problem. Mobile IP is the subject of our first article by Bill Stallings.

The art of cryptography is certainly not new, but its use in computer-communications is a more recent phenomena. The *Data Encryption Standard* (DES) has been widely used since it was standardized in 1977. The strength of a particular encryption scheme depends on the key length and the sophistication of the mathematics involved in transforming the so-called cleartext to the encrypted form. As computers have become more powerful it is now possible to systematically "guess" the 56-bit DES keys in a matter of hours, thus a new encryption standard is needed. This new standard, known as the *Advanced Encryption Standard* (AES), is described by Edgar Danielyan.

Many aspects of computer networking can be described as "controversial," that is, there are strongly held opinions about a particular technology or its use. In this issue we begin a new series of articles labelled "Opinion," hoping to bring out some of the different views held by members of the networking community. We hope you will take issue with some of these columns and send us your own opinion piece. We begin the series with an article by Geoff Huston entitled "The Middleware Muddle." Let us know what you think by sending your comments to **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Mobile IP

by William Stallings

In response to the increasing popularity of palm-top and other mobile computers, Mobile IP was developed to enable computers to maintain Internet connectivity while moving from one Internet attachment point to another. Although Mobile IP can work with wired connections, in which a computer is unplugged from one physical attachment point and plugged into another, it is particularly suited to wireless connections.

The term "mobile" in this context implies that a user is connected to one or more applications across the Internet, that the user's point of attachment changes dynamically, and that all connections are automatically maintained despite the change. This scenario is in contrast to a user, such as a business traveler, with a portable computer of some sort who arrives at a destination and uses the computer notebook to dial into an *Internet Service Provider* (ISP).

In this latter case, the user's Internet connection is terminated each time the user moves, and a new connection is initiated when the user dials back in. Each time an Internet connection is established, software in the point of attachment (typically an ISP) is used to obtain a new, temporarily assigned IP address. For each application-level connection (for example, *File Transfer Protocol* [FTP], Web connection), this temporary IP address is used by the user's correspondent. A better term for this kind of use is "nomadic."

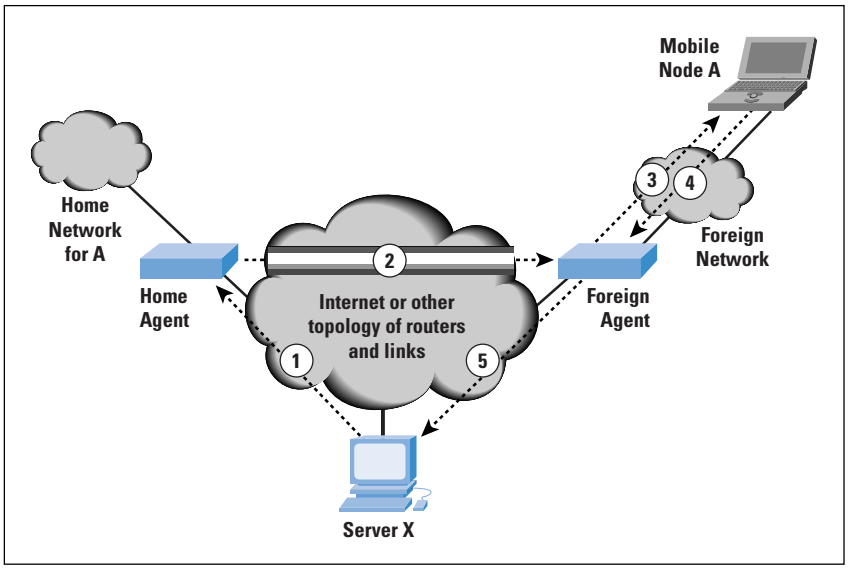
We begin with a general overview of Mobile IP and then look at some of the details.

Operation of Mobile IP

Routers make use of the IP address in an IP datagram to perform routing. In particular, the *network portion* of an IP address is used by routers to move a datagram from the source computer to the network to which the target computer is attached. Then the final router on the path, which is attached to the same network as the target computer, uses the *host portion* of the IP address to deliver the IP datagram to the destination. Further, this IP address is known to the next higher layer in the protocol architecture. In particular, most applications over the Internet are supported by *Transmission Control Protocol* (TCP) connections. When a TCP connection is set up, the TCP entity on each side of the connection knows the IP address of the correspondent host. When a TCP segment is handed down to the IP layer for delivery, TCP provides the IP address. IP creates an IP datagram with that IP address in the IP header and sends the datagram out for routing and delivery. However, with a mobile host, the IP address may change while one or more TCP connections are active.

Figure 1 shows in general terms how Mobile IP deals with the problem of dynamic IP addresses. A mobile node is assigned to a particular network, known as its *home network*. Its IP address on that network, known as its *home address,* is static. When the mobile node moves its attachment point to another network, that is considered a *foreign network* for this host. When the mobile node is reattached, it makes its presence known by registering with a network node, typically a router, on the foreign network known as a *foreign agent*. The mobile node then communicates with a similar agent on the user's home network, known as a *home agent,* giving the home agent the *care-of address* of the mobile node; the care-of address identifies the foreign agent's location. Typically, one or more routers on a network will implement the roles of both home and foreign agents.

Figure 1: Mobile IP Scenario



When IP datagrams are exchanged over a connection between the mobile node (A) and another host (server X in Figure 1), the following operations occur:

1. Server X transmits an IP datagram destined for mobile node A, with A's home address in the IP header. The IP datagram is routed to A's home network.
2. At the home network, the incoming IP datagram is intercepted by the home agent. The home agent encapsulates the entire datagram inside a new IP datagram, which has the A's care-of address in the header, and retransmits the datagram. The use of an outer IP datagram with a different destination IP address is known as *tunneling*.
3. The foreign agent strips off the outer IP header, encapsulates the original IP datagram in a network-level *Protocol Data Unit* (PDU) (for example, a LAN *Logical Link Control* [LLC] frame), and delivers the original datagram to A across the foreign network.

4. When A sends IP traffic to X, it uses X's IP address. In our example, this is a fixed address; that is, X is not a mobile node. Each IP datagram is sent by A to a router on the foreign network for routing to X. Typically, this router is also the foreign agent.
5. The IP datagram from A to X travels directly across the Internet to X, using X's IP address.

To support the operations illustrated in Figure 1, Mobile IP includes three basic capabilities:

- *Discovery:* A mobile node uses a discovery procedure to identify prospective home agents and foreign agents.
- *Registration:* A mobile node uses an authenticated registration procedure to inform its home agent of its care-of address.
- *Tunneling:* Tunneling is used to forward IP datagrams from a home address to a care-of address.

Figure 2 indicates the underlying protocol support for the Mobile IP capability. The registration protocol communicates between an application on the mobile node and an application in the home agent, and hence uses a transport-level protocol. Because registration is a simple request/response transaction, the overhead of the connection-oriented TCP is not required, and, therefore, the *User Datagram Protocol* (UDP) is used as the transport protocol. Discovery makes use of the existing *Internet Control Message Protocol* (ICMP) by adding the appropriate extensions to the ICMP header. ICMP is a connectionless protocol well suited for the discovery operation. Finally, tunneling is performed at the IP level.

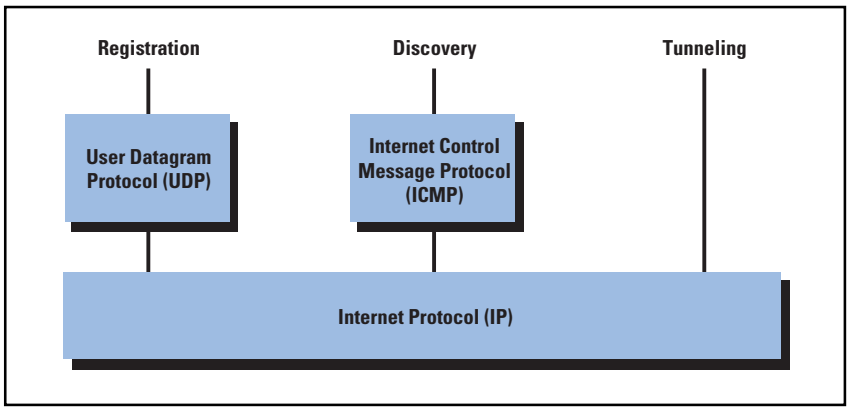


Figure 2: Protocol Support for Mobile IP

Discovery

The discovery process in Mobile IP is very similar to the router advertisement process defined in ICMP. Accordingly, agent discovery makes use of ICMP router advertisement messages, with one or more extensions specific to Mobile IP.

The mobile node is responsible for an ongoing discovery process. It must determine if it is attached to its home network, in which case IP datagrams may be received without forwarding, or if it is attached to a foreign network.

Because handoff from one network to another occurs at the physical layer, a transition from the home network to a foreign network can occur at any time without notification to the network layer (that is, the IP layer). Thus, discovery for a mobile node is a continuous process.

For the purpose of discovery, a router or other network node that can act as an agent periodically issues a router advertisement ICMP message with an advertisement extension. The router advertisement portion of the message includes the IP address of the router. The advertisement extension includes additional information about the role of the router as an agent, as discussed subsequently. A mobile node listens for these *agent advertisement messages*. Because a foreign agent could be on the home network of the mobile node (set up to serve visiting mobile nodes), the arrival of an agent advertisement does not necessarily tell the mobile node that it is on a foreign network. The mobile node must compare the network portion of the router IP address with the network portion of its own home address. If these network portions do not match, then the mobile node is on a foreign network.

The *agent advertisement extension* follows the ICMP router advertisement fields and consists of the following fields:

- *Type:* 16, indicates that this is an agent advertisement.
- *Length:* (6 + 4*N*), where *N* is the number of care-of addresses advertised.
- *Sequence number:* The count of agent advertisement messages sent since the agent was initialized.
- *Lifetime:* The longest lifetime, in seconds, that this agent is willing to accept a registration request from a mobile node.
- *R:* Registration with this foreign agent is required (or another foreign agent on this network). Even those mobile nodes that have already acquired a care-of address from this foreign agent must reregister.
- *B:* Busy. The foreign agent will not accept registrations from additional mobile nodes.
- *H:* This agent offers services as a home agent on this network.
- *F:* This agent offers services as a foreign agent on this network.
- *M:* This agent can receive tunneled IP datagrams that use minimal encapsulation, explained subsequently.
- *G:* This agent can receive tunneled IP datagrams that use *Generic Routing Encapsulation* (GRE), explained subsequently.
- *Y:* This agent supports the use of Van Jacobson header compression, an algorithm defined in RFC 1144 for compressing fields in the TCP and IP headers.
- *Care-of address:* The care-of address or addresses supported by this agent on this network. There must be at least one such address if the F bit is set. There may be multiple addresses.

There may also be an optional *prefix-length extension* following the advertisement extension. This extension indicates the number of bits in the router address that define the network number. The mobile node uses this information to compare the network portion of its own IP address with the network portion of the router. The fields include the following:

- *Type:* 19, indicates that this is a prefix-length advertisement.
- *Length: N,* where *N* is the value of the Num Addrs field in the ICMP router advertisement portion of this ICMP message. In other words, this is the number of router addresses listed in this ICMP message.
- *Prefix length:* The number of leading bits that define the network number of the corresponding router address listed in the ICMP router advertisement portion of this message. The number of prefix length fields matches the number of router address fields (*N*).

Foreign agents are expected to periodically issue agent advertisement messages. If a mobile node needs agent information immediately, it can issue an ICMP router solicitation message. Any agent receiving this message will then issue an agent advertisement.

As was mentioned, a mobile node may move from one network to another because of some handoff mechanism, without the IP level being aware of it. The agent discovery process is intended to enable the agent to detect such a move. The agent may use one of two algorithms for this purpose:

- *Use of Lifetime field:* When a mobile node receives an agent advertisement from a foreign agent that it is currently using or that it is now going to register with, it records the Lifetime field as a timer. If the timer expires before the agent receives another agent advertisement from the agent, then the node assumes that it has lost contact with that agent. If, in the meantime, the mobile node has received an agent advertisement from another agent and that advertisement has not yet expired, the mobile node can register with this new agent. Otherwise, the mobile node should use agent solicitation to find an agent.
- *Use of network prefix:* The mobile node checks whether any newly received agent advertisement is on the same network as the current care-of address of the node. If it is not, the mobile node assumes that it has moved and may register with the agent whose advertisement the mobile node has just received.

The discussion so far has involved the use of a care-of address associated with a foreign agent; that is, the care-of address is an IP address for the foreign agent. This foreign agent will receive datagrams at this care-of address, intended for the mobile node, and then forward them across the foreign network to the mobile node. However, in some cases a mobile node may move to a network that has no foreign agents or on which all foreign agents are busy.

As an alternative, the mobile node may act as its own foreign agent by using a *colocated care-of address*. A colocated care-of address is an IP address obtained by the mobile node that is associated with the current interface to a network of that mobile node.

The means by which a mobile node acquires a colocated address is beyond the scope of Mobile IP. One means is to dynamically acquire a temporary IP address through an Internet service such as *Dynamic Host Configuration Protocol* (DHCP). Another alternative is that the colocated address may be owned by the mobile node as a long-term address for use only while visiting a given foreign network.

Registration

When a mobile node recognizes that it is on a foreign network and has acquired a care-of address, it needs to alert a home agent on its home network and request that the home agent forward its IP traffic. The registration process involves four steps:

1. The mobile node requests the forwarding service by sending a registration request to the foreign agent that the mobile node wants to use.
2. The foreign agent relays this request to the home agent of that mobile node.
3. The home agent either accepts or denies the request and sends a registration reply to the foreign agent.
4. The foreign agent relays this reply to the mobile node.

If the mobile node is using a colocated care-of address, then it registers directly with its home agent, rather than going through a foreign agent.

The registration operation uses two types of messages, carried in UDP segments. The *registration request message* consists of the following fields:

- *Type:* 1, indicates that this is a registration request.
- *S:* Simultaneous bindings. The mobile node is requesting that the home agent retain its prior mobility bindings. When simultaneous bindings are in effect, the home agent will forward multiple copies of the IP datagram, one to each care-of address currently registered for this mobile node. Multiple simultaneous bindings can be useful in wireless handoff situations to improve reliability.
- *B:* Broadcast datagrams. Indicates that the mobile node would like to receive copies of broadcast datagrams that it would have received if it were attached to its home network.
- *D:* Decapsulation by mobile node. The mobile node is using a colocated care-of address and will decapsulate its own tunneled IP datagrams.
- *M:* Indicates that the home agent should use minimal encapsulation, explained subsequently.

- *V:* Indicates that the home agent should use Van Jacobson header compression, an algorithm defined in RFC 1144 for compressing fields in the TCP and IP headers.
- *G:* Indicates that the home agent should use GRE encapsulation, explained subsequently.
- *Lifetime:* The number of seconds before the registration is considered expired. A value of zero is a request for deregistration.
- *Home address:* The home IP address of the mobile node. The home agent can expect to receive IP datagrams with this as a destination address, and must forward those to the care-of address.
- *Home agent:* The IP address of the mobile node home agent. This informs the foreign agent of the address to which this request should be relayed.
- *Care-of address:* The IP address at this end of the tunnel. The home agent should forward IP datagrams that it receives with the mobile node home address to this destination address.
- *Identification:* A 64-bit number generated by the mobile node, used for matching registration requests to registration replies and for security purposes, as explained subsequently.
- *Extensions:* The only extension so far defined is the authentication extension, explained subsequently.

The *registration reply message* consists of the following fields:

- *Type:* 3, indicates that this is a registration reply.
- *Code:* Indicates result of the registration request.
- *Lifetime:* If the code field indicates that the registration was accepted, the number of seconds before the registration is considered expired. A value of zero indicates that the mobile node has been deregistered.
- *Home address:* The home IP address of the mobile node.
- *Home agent:* The IP address of the mobile node home agent.
- *Identification:* A 64-bit number used for matching registration requests to registration replies.

The only extension so far defined is the authentication extension, explained subsequently.

A key concern with the registration procedure is security. Mobile IP is designed to resist two types of attacks:

1. A node may pretend to be a foreign agent and send a registration request to a home agent so as to divert traffic intended for a mobile node to itself.
2. A malicious agent may replay old registration messages, effectively cutting the mobile node from the network.

The technique that is used to protect against such attacks involves the use of message authentication and the proper use of the identification field of the registration request and reply messages.

For purposes of message authentication, each registration request and reply contains an *authentication extension* with the following fields:

- *Type:* Used to designate the type of this authentication extension.
- *Length:* 4 plus the number of bytes in the authenticator.
- *Security parameter index (SPI):* An index that identifies a security context between a pair of nodes. This security context is configured so that the two nodes share a secret key and parameters relevant to this association (for example, authentication algorithm).
- *Authenticator:* A code used to authenticate the message. The sender inserts this code into the message using a shared secret key. The receiver uses the code to ensure that the message has not been altered or delayed. The authenticator protects the entire registration request or reply message, any extensions prior to this extension, and the type and length fields of this extension.

The default authentication algorithm uses keyed MD5 to produce a 128-bit message digest. For Mobile IP, a "prefix+suffix" mode of operation is used. The MD5 digest is computed over the shared secret key, followed by the protected fields from the registration message, followed by the shared secret key again. Three types of authentication extensions are defined:

- *Mobile-home:* This extension must be present and provides for authentication of the registration messages between the mobile node and the home agent.
- *Mobile-foreign:* The extension may be present when a security association exists between the mobile node and the foreign agent. The agent will strip this extension off before relaying a request message to the home agent and add this extension to a reply message coming from a home agent.
- *Foreign-home:* The extension may be present when a security association exists between the foreign agent and the home agent.

Note that the authenticator protects the identification field in the request and reply messages. As a result, the identification value can be used to thwart replay types of attacks. As was mentioned, the identification value enables the mobile node to match a reply to a request. Further, if the mobile node and the home agent maintain synchronization so that the home agent can distinguish a reasonable identification value from a suspicious one, then the home agent can reject suspicious messages. One way to do this is to use a timestamp value. As long as the mobile node and home agent have reasonably synchronized values of time, the timestamp will serve the purpose. Alternatively, the mobile node could generate values using a pseudorandom number generator. If the home agent knows the algorithm, then it knows what identification value to expect next.

Tunneling

When a mobile node is registered with a home agent, the home agent must be able to intercept IP datagrams sent to the mobile node home address so that these datagrams can be forwarded via tunneling. The standard does not mandate a specific technique for this purpose but references *Address Resolution Protocol* (ARP) as a possible mechanism. The home agent needs to inform other nodes on the same network (the home network) that IP datagrams with a destination address of the mobile node in question should be delivered (at the link level) to this agent. In effect, the home agent steals the identity of the mobile node in order to capture packets destined for that node that are transmitted across the home network.

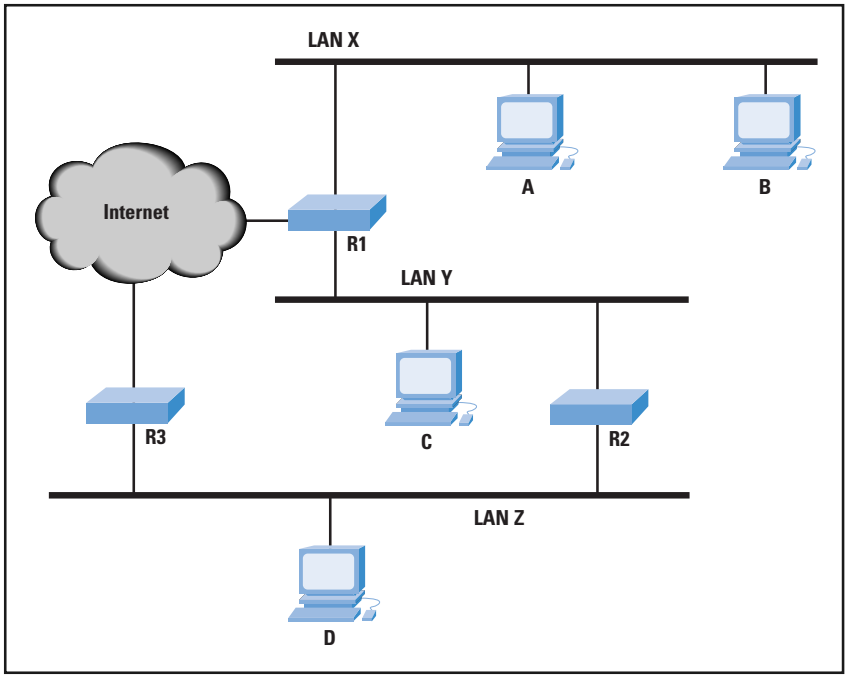


Figure 3: A Simple Internetworking Example

For example, suppose that R3 in Figure 3 is acting as the home agent for a mobile node that is attached to a foreign network elsewhere on the Internet. That is, there is a host H whose home network is LAN Z that is now attached to some foreign network. If host D has traffic for H, it will generate an IP datagram with H's home address in the IP destination address field. The IP module in D recognizes that this destination address is on LAN Z and so passes the datagram down to the link layer with instructions to deliver it to a particular *Media Access Control* (MAC)-level address on Z. Prior to this time, R3 has informed the IP layer at D that datagrams destined for that particular address should be sent to R3. Thus, the MAC address of R3 is inserted by D in the destination MAC address field of the outgoing MAC frame. Similarly, if an IP datagram with the mobile node home address arrives at router R2, it recognizes that the destination address is on LAN Z and will attempt to deliver the datagram to a MAC-level address on Z. Again, R2 has previously been informed that the MAC-level address it needs corresponds to R3.

For traffic that is routed across the Internet and arrives at R3 from the Internet, R3 must simply recognize that for this destination address, the datagram is to be captured and forwarded.

To forward an IP datagram to a care-of address, the home agent puts the entire IP datagram into an outer IP datagram. This is a form of encapsulation, just as placing an IP header in front of a TCP segment encapsulates the TCP segment in an IP datagram. Three options for encapsulation are allowed for Mobile IP and we will review the first two of the following options:

- *IP-within-IP encapsulation:* This is the simplest approach, defined in RFC 2003.
- *Minimal encapsulation:* This approach involves fewer fields, defined in RFC 2004.
- *Generic routing encapsulation (GRE):* This is a generic encapsulation procedure, defined in RFC 1701, that was developed prior to the development of Mobile IP.

In the IP-within-IP encapsulation approach, the entire IP datagram becomes the payload in a new IP datagram (Figure 4a). The inner, original IP header is unchanged except to decrement *Time To Live* (TTL) by 1. The outer header is a full IP header. Two fields (indicated as unshaded in the figure) are copied from the inner header. The version number is 4, the protocol identifier for IPv4, and the type of service requested for the outer IP datagram is the same as that requested for the inner IP datagram.

Figure 4a: Mobile IP Encapsulation

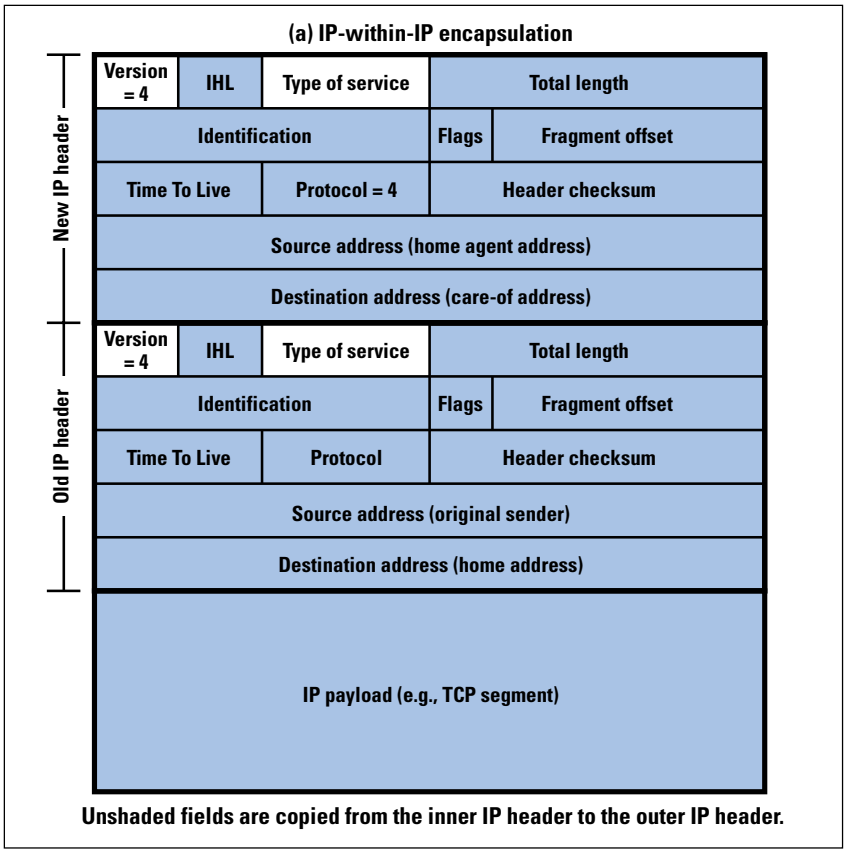
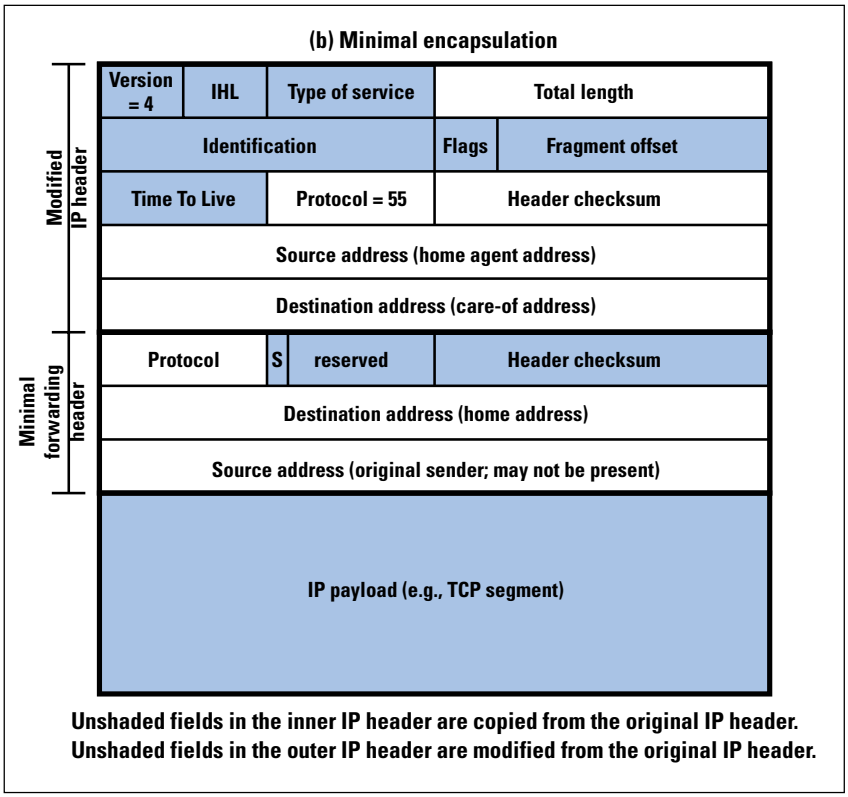


Figure 4b: Mobile IP Encapsulation



In the inner IP header, the source address refers to the host that is sending the original datagram, and the destination address is the home address of the intended recipient. In the outer IP header, the source and destination addresses refer to the entry and exit points of the tunnel. Thus, the source address typically is the IP address of the home agent, and the destination address is the care-of address for the intended destination.

Example: Consider an IP datagram that originates at server X in Figure 1 and that is intended for mobile node A. The original IP datagram has a source address equal to the IP address of X and a destination address equal to the IP home address of A. The network portion of A's home address refers to A's home network, so the datagram is routed through the Internet to A's home network, where it is intercepted by the home agent. The home agent encapsulates the incoming datagram with an outer IP header, which includes a source address equal to the IP address of the home agent and a destination address equal to the IP address of the foreign agent on the foreign network to which A is currently attached. When this new datagram reaches the foreign agent, it strips off the outer IP header and delivers the original datagram to A.

Minimal encapsulation results in less overhead and can be used if the mobile node, home agent, and foreign agent all agree to do so. With minimal encapsulation, the new header is inserted between the original IP header and the original IP payload (Figure 4b). It includes the following fields:

- *Protocol:* Copied from the Destination Address field in the original IP header. This field identifies the protocol type of the original IP payload and thus identifies the type of header that begins the original IP payload.
- *S:* If 0, the original source address is not present, and the length of this header is 8 octets. If 1, the original source address is present, and the length of this header is 12 octets.
- *Header checksum:* Computed over all the fields of this header.
- *Original destination address:* Copied from the Destination Address field in the original IP header.
- *Original source address:* Copied from the Source Address field in the original IP header. This field is present only if the S bit is 1. The field is not present if the encapsulator is the source of the datagram (that is, the datagram originates at the home agent).

The following fields in the original IP header are modified to form the new outer IP header:

- *Total length:* Incremented by the size of the minimal forwarding header (8 or 12).
- *Protocol:* 55; this is the protocol number assigned to minimal IP encapsulation.
- *Header checksum:* Computed over all the fields of this header; because some of the fields have been modified, this value must be recomputed.
- *Source address:* The IP address of the encapsulator, typically the home agent.
- *Destination address:* The IP address of the exit point of the tunnel. This is the care-of address and may be either the IP address of the foreign agent or the IP address of the mobile node (in the case of a colocated care-of address).

The processing for minimal encapsulation is as follows. The encapsulator (home agent) prepares the encapsulated datagram with the format of Figure 4b. This datagram is now suitable for tunneling and is delivered across the Internet to the care-of address. At the care-of address, the fields in the minimal forwarding header are restored to the original IP header and the forwarding header is removed from the datagram. The total length field in the IP header is decremented by the size of the minimal forwarding header (8 or 12) and the header checksum field is recomputed.

References

Reference [1] is a good survey article on mobile IP; a somewhat less technical, more business-oriented description from the same author is [2]. For greater detail, see [3]. The August 2000 issue of *IEEE Personal Communications* contains numerous articles on enhancements to the current Mobile IP standard. The Web site of the IETF Working Group on Mobile IP, which contains current RFCs and Internet Drafts is at:

`http://ietf.org/html.charters/mobileip-charter.html`

[1] Perkins, C., "Mobile IP," *IEEE Communications Magazine,* May 1997.

[2] Perkins, C., "Mobile Networking through Mobile IP," *IEEE Internet Computing,* January-February 1998.

[3] Perkins, C., *Mobile IP: Design Principles and Practices,* ISBN 0-201-63469-4, Prentice Hall PTR, 1998.

[4] Solomon, J., *Mobile IP: The Internet Unplugged,* ISBN 0138562466, Prentice Hall PTR, 1998.

WILLIAM STALLINGS is a consultant, lecturer, and author of over a dozen books on data communications and computer networking. He also maintains a computer science resource site for CS students and professionals at **`WilliamStallings.com/StudentSupport.html`**. He has a PhD in computer science from M.I.T. His latest book is *Wireless Communications and Networks* (Prentice Hall, 2001). His home in cyberspace is **`WilliamStallings.com`** and he can be reached at **`ws@shore.net`**

Goodbye DES, Welcome AES

by Edgar Danielyan

Much has changed since introduction of the *Data Encryption Standard* (DES)[2] in 1977. Our hardware is faster, we have more memory, and the use of computer networks in all areas of human activity is increasing. The widely used DES has, on several occasions, been proven to be inadequate for many applications—especially those involving the transmission of sensitive information over public networks such as the Internet, where the entire transmission may be intercepted and cryptanalyzed. Specialized hardware has been built that can determine the 56-bit DES key in a few hours. These considerations, and others, have signaled that a new standard algorithm and longer keys are necessary.

Fortunately, in January 1997, the U.S. *National Institute of Standards and Technology* (NIST) announced that it's time for a new encryption standard: the *Advanced Encryption Standard* (AES). They formalized their requirements and issued a call for candidate algorithm nominations in September 1997. The deadline for submissions was June 1998, when a total of 15 algorithms were submitted for consideration. This article shows why DES is outdated and should not be used for any purposes that require serious encryption. It also provides a brief description of the soon-to-come replacement of DES, the Advanced Encryption Standard.

Data Encryption Standard

Published as the U.S. Federal Information Processing Standard 46 in 1977, DES is still widely used, despite being proven inadequate for use in many applications. It is a symmetric block cipher (shared secret key), with its block size fixed at 64 bits. There are four defined modes of operation, with the *Electronic Code Book* (ECB) mode being the most widely used[1]. Additionally, DES has been incorporated into numerous other standards, such as American Bankers Association's *Protection of Personal Identification Numbers in Interchange Standard, Management and Use of Personal Identification Numbers Standard, Key Management Standard,* and three ANSI standards, *Data Encryption Algorithm* (DEA), *Standard for Personal Identification Number (PIN) Management and Security,* and *Standard for Financial Institution Message Authentication*[3]. In particular, DES is also specified as an approved algorithm in the *IP Security Architecture* (IPSec) standard[9], which is used in the equipment from many different suppliers.

Key Length

Key length is one of the two most important security factors of any encryption algorithm—the other one being the design of the algorithm itself. DES uses a 64-bit block for the key; however, 8 of these bits are used for odd parity and are, therefore, not counted in the key length. The effective key length is then calculated as 56 bits, giving 2^{56} possible keys. A true 64-bit key has 256 times as many keys, whereas a 128-bit key is 2^{72} times "better" than a 56-bit key. As if this was not enough, DES also has so-called *weak* and semi-*weak* keys. During the encryption process, the key is used to generate two values that are used for separate purposes during the process. These 16 weak and semi-weak keys will produce values that don't appear to be random. They will give outputs of all-ones, all-zeros, or distinguishable patterns of ones and zeros. It is generally recognized that these 16 key values should not be used. The key length was known to be a factor in trusting DES soon after DES was published. For this reason, people started exploring the use of multiple encryption passes and multiple keys. *Triple DES* (3DES) is a way of using DES encryption three times.

The most common method is to first encrypt the data block with one key. The output of this operation is run through the decryption process with a second key, and the output of that operation is run through the encryption process again with the first key. This process makes the effective key length 112 bits long. Again, the problem with weak and semi-weak keys remains. The disadvantage of Triple DES is that it is about one-third as fast as DES when processing data. This effort just slightly extended the life of DES while a suitable alternative could be found.

Breaking the DES

In addition to the brute-force key search (for example, trying every possible key in order to recover the plaintext—for DES that would be 2^{56} keys), there is also a technique known as *cryptanalysis*, which may be used to find the key or the plaintext. Essentially, there are two publicized ways to cryptanalyze DES: *differential* and *linear*. Discovered by Biham and Shamir in 1990, differential cryptanalysis was previously unknown to the public. In short, differential cryptanalysis looks at the difference between pairs of ciphertext and uses the information about these differences to find the key. Linear cryptanalysis, discovered by M. Matsui, on the other hand, uses a method called *linear approximations* to analyze block ciphers (not only DES). Because some internal structures used in DES are not designed to be strong against linear cryptanalysis, it is quite effective when used against DES. To show that the DES is inadequate and should not be used in important systems anymore, RSA Data Security[7] sponsored a challenge to see how long it would take to decrypt successively more difficult algorithms (see **`http://www.rsasecurity.com/rsalabs/challenges`** for more information). Two organizations played key roles in breaking the DES: the distributed.net and the *Electronic Frontier Foundation* (EFF).

distributed.net

distributed.net[6] is a worldwide distributed computing network. Started in 1997, the company now has thousands of participants who are contributing their idle computing power to provide an equivalent of about 160,000 Pentium II computers working in parallel. The company's mission statement says, in particular:

"We will deploy our software to form an immense, globally distributed computer that solves large-scale problems and provides an accessible pool of computational power to projects that need it. This deployment will also demonstrate the real-world utility of both distributed computing in general and our software in particular."

It may be said that they are doing well: projects undertaken and successfully completed by distributed.net include the CS Cipher, DES III, DES II 2, and RC5-56 challenges. At the time of writing, distributed.net is working on two projects: breaking RC5 with a 64-bit key and finding *Optimal Golomb Rulers* (OGRs). The idea behind distributed.net is that it is possible to distribute chunks of data over the Internet to be processed in parallel by participating computers during their idle time. The results of these calculations are then sent to a central computer that coordinates the distributed computation. The same principle is used by the SETI (Search for Extraterrestrial Intelligence) @ Home project.

Electronic Frontier Foundation

The EFF's DES cracking computer was designed by Cryptography Research, Advanced Wireless Technologies, and the EFF[5]. The design was based upon theoretical work by Michael Wiener[10]. It checked 90 billion keys per second, was assembled in six Sun 2 cabinets, and had 27 boards and 1800 custom chips. Built for less than $250,000, it found the key in approximately 56 hours of brute-force search.

DES I

The DES I contest was the first attempt to prove that DES is no longer fit for any serious use. It was completed on June 17, 1997, by R. Verser in a collaborative effort, after checking about 14 percent (10,178,478,175,420,416 keys) of the key space. It took 84 days.

DES II

There were, in fact, two DES II challenges. distributed.net participated in the first one, which began on January 13, 1998, and completed it on February 23, 1998. About 63 quadrillion keys were checked. At the end, the participants of distributed.net were checking 28 gigakeys per second. The decrypted text was "The unknown message is: Many hands make light work." The EFF won the second challenge on July 15, 1998, in less than three days, with distributed.net coming in second. This time the plaintext read "It's time for those 128-, 192-, and 256-bit keys."

DES III

The DES III contest, announced by RSA Data Security on December 12, 1998, to start on January 18, 1999, was also a success. In an official press release, RSA said:

> "First adopted by the federal government in 1977, the 56-bit DES algorithm is still widely used by financial services and other industries to protect sensitive on-line applications, despite growing doubts about its vulnerability to hackers. It has been widely known that 56-bit keys, such as those offered by the government's DES standard, offer marginal protection against a committed adversary."

It took 22 hours and 15 minutes for Electronic Frontier Foundation's Deep Crack computer and distributed.net's worldwide distributed computing network to find out the 56-bit DES key, decipher the message, and win the $10,000 contest. The decrypted message read "See you in Rome (Second AES Conference, March 22–23, 1999)" and was found after checking about 30 percent of the key space. This latest exercise finally proved that DES belongs to the past.

AES Timeline

In April 1997, NIST organized a workshop to consider criteria and submission guidelines of candidate algorithms; later in September, an official call for nominations was published in the U.S. Federal Register. By June 1998, 15 algorithms were submitted to the NIST for consideration:

- CAST-256 (Entrust Technologies)
- CRYPTON (Future Systems)
- DEAL (Richard Outerbridge, Lars Knudsen)
- DFC (National Centre for Scientific Research, France)
- E2 (NTT)
- FROG (TecApro Internacional)
- HPC (Rich Schroeppel)
- LOKI97 (Lawrie Brown, Josef Pieprzyk, Jennifer Seberry)
- MAGENTA (Deutsche Telekom)
- Mars (IBM)
- RC6 (RSA)
- Rijndael (Joan Daemen, Vincent Rijmen)
- Safer+ (Cylink)
- Serpent (Ross Anderson, Eli Biham, Lars Knudsen)
- Twofish (Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson)

NIST asked for public comments on these 15 algorithms and set the date for the second AES candidate conference to March 1999, to be held in Rome, Italy. The candidate algorithms were tested from both cryptological and performance viewpoints. One of the original NIST requirements for the algorithm was that it had to be efficient both in software and hardware implementations. (DES was originally practical only in hardware implementations.) Java and C reference implementations were used to do performance analysis of the algorithms. A few months later, a NIST press release announced the selection of 5 out of 15 algorithms that survived rigorous testing and cryptanalysis. This fact is not to say that the algorithms that were not selected were broken or were without merit. Those algorithms either were not as efficient, or were not as practical to implement.

The selected algorithms were Mars, RC6, Rijndael, Serpent, and Twofish. These algorithms were accepted as cryptologically strong and flexible, as well as able to be efficiently implemented in software and hardware. In August 2000, the National Security Agency published the VHDL model for performance testing of algorithms when implemented in hardware. Finally, in October 2000, a NIST press release announced the selection of Rijndael as the proposed Advanced Encryption Standard.

Rijndael

Rijndael[4] (pronounced "Reign Dahl," "Rain Doll," or "Rhine Dahl") was designed by Joan Daemen, PhD (Proton World International, Belgium) and Vincent Rijmen (Catholic University of Leuven, Belgium). Both authors are internationally known cryptographers. Rijndael is an efficient, symmetric block cipher. It supports key and block sizes of 128, 192, and 256 bits. The main design goals for the algorithm were simplicity, performance, and strength (that is, resistance against cryptanalysis). When used in *Cipher Block Chaining Message Authentication Code* (CBC MAC) mode, Rijndael can be used as a MAC algorithm; it also may be used as a hash function and as a pseudo random number generator (both are special mathematical functions widely used in cryptography; an example of a hash function is *Message Digest 5* (MD5)—a popular message digest algorithm by Ron Rivest). In their specification of the algorithm, the authors specifically state the strength of Rijndael against differential, truncated differential, linear, interpolation, and Square attacks. Although Rijndael is not based on Square[8], some ideas from the Square algorithm design are used in Rijndael.

Square is a 128-bit symmetric iterated block cipher designed by Daemen, Rijnmen, and Knudsen. Its primary design goal was strength against both linear and differential cryptanalyses; the high degree of parallelism of the Square algorithm allows efficient implementation on parallel computers.

Of course, the length of the key is also very important, especially because the most efficient known attack against Rijndael is an exhaustive key search. It would take 2^{255} runs of Rijndael to find a key 256 bits long. To the credit of the authors, Rijndael does not use "parts" or tables from other algorithms, making it easy to implement alone.

Table 1: Comparing DES and AES

	DES	AES
Key Length	56 bits	128, 192, or 256 bits
Cipher Type	Symmetric block cipher	Symmetric block cipher
Block Size	64 bits	128, 192, or 256 bits
Developed	1977	2000
Cryptanalysis resistance	Vulnerable to differential and linear cryptanalysis; weak substitution tables	Strong against differential, truncated differential, linear, interpolation and Square attacks
Security	Proven inadequate	Considered secure
Possible Keys	2^{56}	2^{128}, 2^{192}, or 2^{256}
Possible ASCII printable character keys*	95^{7}	95^{16}, 95^{24}, or 95^{32}
Time required to check all possible keys at 50 billion keys per second**	For a 56-bit key: 400 days	For a 128-bit key: 5 x 10^{21} years

* When a text password input by a user is used for encryption (there are 95 printable characters in ASCII).

** In theory, the key may be found after checking 1/2 of the key space. The time shown is 100% of the key space.

Summary

It is expected that AES will be officially published as a *Federal Information Processing Standard* (FIPS) in April–June 2001, and implementations of AES in various security systems probably will surface shortly thereafter. In the meantime, authoritative information on AES developments may be found on NIST's Web site at **`http://csrc.nist.gov/encryption/aes/`**. The full mathematical specification of the algorithm and reference implementations in C and Java are also available from the same Web site.

References

[1] *Applied Cryptography,* 2nd edition, by Bruce Schneier, 1996, John Wiley & Sons.

[2] National Institute of Standards and Technology (NIST), **http://www.nist.gov**

[3] American National Standards Institute (ANSI), **http://www.ansi.org**

[4] The Rijndael Specification, **http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf**

[5] Electronic Frontier Foundation, **http://www.eff.org**

[6] distributed.net, **http://www.distributed.net**

[7] RSA Security, **http://www.rsa.com**

[8] Square Specification, **http://www.esat.kuleuven.ac.be/~rijmen/square**

[9] Kent, S., Atkinson, R., "Security Architecture for the Internet Protocol," RFC 2401, November 1998.

[10] Michael Wiener, "Efficient DES Key Search," Proceedings of the CRYPTO'93 Conference, August 1993.

[11] Madson, C., Doraswamy, "The ESP DES-CBC Cipher Algorithm With Explicit IV," RFC 2405, November 1998.

[A prior version of this article was published in the February 2001 issue of the *;login:* magazine].

EDGAR DANIELYAN is a Cisco Certified Network, Design and Security Professional, as well as member of ACM, USENIX, SAGE, and the IEEE Computer Society. He has worked for a national telco, a bank, the United Nations, and the Ministry of Defense, among others. Currently self-employed, he consults and writes on internetworking, UNIX, and security. E-mail: **edd@danielyan.com**

Opinion: The Middleware Muddle

by Geoff Huston

[This occasional column is an individual soapbox on views of various aspects of the Internet. The views stated here are intended to be mildly provocative, and, if backed to the wall, the author will rapidly disclaim any responsibility for them whatsoever!]

It is not often that an entire class of technology can generate an emotive response. But, somehow, middleware has managed to excite many strong reactions. For some *Internet Service Providers* (ISPs), middleware—in the form of *Web caches*—is not only useful, it's critical to the success of their enterprise. For many corporate networks, middleware—in the form of *firewalls*—is the critical component of their network security measures. For such networks, middleware is an integral part of the network. Other networks use middleware, in the form of *Network Address Translators* (NATs), as a means of stretching a limited number of Internet public addresses to provide connectivity services to a much larger local network. For others, middleware is seen as something akin to network heresy. For them, not only does middleware often break the basic semantics of the Internet Protocol, it is also in direct contravention to the end-to-end architecture of the Internet. Middleware, they claim, breaks the operation of entire classes of useful applications, and this makes the Internet a poorer network as a result.

Emotions have run high in the middleware debate, and middleware has been portrayed as being everything from absolutely essential to the operation of the Internet as we know it, to being immoral and deceptive. Strong stuff indeed from an engineering community, even one as traditionally opinionated as Internet engineers.

So what is middleware all about and why the fuss?

It may be helpful to start with a definition of middleware. One definition of middleware is that of anything in the network that functions at a level in a network reference model above that of end-to-end transport (TCP/IP), and below that of the application environment (the *Application Programming Interface* [API])[1]. Of course, this definition encompasses a very broad class of services that covers everything from *Authentication, Authorization, and Accounting* (AAA) servers and *Domain Name System* (DNS) servers through to various forms of information discovery services and resource management.

Another possible definition of middleware adopts the perspective of the integrity of the end-to-end model of Internet architecture[2]. From this perspective, middleware is a class of network devices that do something other than forward or discard an IP packet onward along the next hop to the destination address of the packet—in other words, anything other than a packet-switching element that sits in the transmission path of the packet.

With such an end-to-end definition of middleware, these middleware units may intercept the packet and alter the header or payload of the packet, redirect the packet to be delivered to somewhere other than its intended destination, or process the packet as if it were addressed to the middleware device itself. From this perspective, AAA, the DNS, and related services from our first definition are simply applications that traverse the network.

There's nothing like confusion over definitions to fuel a debate, and this area is no exception. However, a debate over definitions is too often a dry one. So, in the interest of adding a little more incendiary material to the topic, let's simply use this second definition of middleware to look further at the issues.

Why would a network go to all this bother to trap and process certain packets? Surely it's easier and cheaper to simply forward the packet onward to its intended destination? The answer can be "yes" or "no," depending on how you feel about the role of middleware in TCP/IP.

An Example: Cache Middleware

Let's look at this in a bit more detail, using a specific flavor of middleware to illustrate the middleware dilemma. A common form of middleware is the *Transparent Web Cache*. Such a Web cache is constructed using two parts, an *interceptor* and a *cache system*. The interceptor is placed into the network, either as a software module added to a router or as a device, which is spliced into a point-to-point link. The interceptor takes all incoming TCP traffic addressed to port 80 (a *Hypertext Transfer Protocol* [HTTP] session) and redirects it across to the cache system. All other traffic is treated normally. The cache system accepts all such redirected packets as if they were directly addressed to the cache itself. It responds to the HTTP requestor as if it were the actual intended destination, using a source address that matches the destination address of the original request, assuming the identity of the actual intended content server. If the requested Web object is located in the local cache, it will deliver the object to the requestor immediately. If it is not in the cache, it will set up its own session with the original destination, send it the original request, and feed the response back to the requestor, while also keeping a copy for itself in its cache.

Caching of content works well in the Web world simply because so much Web traffic today is movement of the same Web page to different recipients. It is commonly reported that up to one half of all Web traffic in the Internet is a duplicate transmission of content. If an ISP locally caches all Web content as it is delivered, and checks the cache before passing through a content request, then the ISP's upstream Web traffic volume may be halved. Even a moderately good cache will be able to service about one quarter of the Web content from the cache. That amount of local caching can be translated into a significant cost saving for the ISP.

The cached Web content is traffic that is not purchased as transit traffic from an upstream ISP, representing a potential saving on the cost of upstream transit services. This saving, in turn, can allow the ISP to operate at a lower price point in the retail market. The cache is also located closer to the ISP's customers, and with appropriate tuning, the cache can also deliver cached content to the customer at a consistently much faster rate than a request to the original content server. For very popular Web sites the originating server may be operating more slowly under extreme load, while the local cache continues to operate at a more consistent service level. The combination of the potential for improved performance and lower overall cost is certainly one that looks enticing: the result is the same set of Web transactions delivered to customers, but cheaper and faster.

End-to-End Issues with Cache Middleware

But not everything is perfect in this transparent caching world. What if the Web server used a security model that served content only to certain requestors, and the identity of the requestor was based on their IP address? This is not a very good security model, admittedly, but it's simple, and because of its simplicity this practice enjoys very common usage. With the introduction of a transparent cache, the Web client sees something quite strange. The Web client can ping the Web server, the client can communicate with any other port on the server, and if the client were to query the status of the server, the Web server would be seen to be functioning quite normally. But, mysteriously, the client cannot retrieve any Web content from the server, and the server does not see any such request from the client. The middleware cache is sitting inside a network somewhere on the path between the client and the service, but it may well be the case that neither the end client or the end server are aware of the deployment of the middleware unit. It is not surprising that this is a remarkably challenging operational problem for either the client or the server to correctly diagnose.

A similar case is where a Web server wishes to deliver different content to different requestors, based on some inference gained from the source IP address of the requestor, or the time of day, or some other variable derived from the circumstances of the request. A transparent cache will not detect such variations in the response of the server and will instead deliver the same version of the cached content to all clients whose requests pass through the transparent cache. Variations of this situation of perceived abnormal service behavior abound, all clustered around the same concept that it is unwise in such an environment for a server to assume that it is always communicating with the end client. Indeed the situation is common enough that the Web application has explicit provision for instructing cache servers about whether the content can be cached and replayed in response to similar subsequent requests.

More subtle vulnerabilities also are present in such a middleware environment. A client can confidently assert that packets are being sent to a server, and the server appears to be responding, but the data appears to have been corrupted. Has the server been compromised? It may look like this is the case, but when middleware is around, looks can be deceiving. If the integrity of the cache is compromised, and different pages are substituted in the cache, then to the clients of the cache it appears that the integrity of original server has been compromised. The twist with transparent cache middleware is that the clients of the cache may be unaware that the cache exists, let alone that their requests are being redirected to the cache server. Any abnormalities in the responses they receive are naturally attributed to problems with the security of the server and the integrity of the associated service.

The common theme of these issues is that there are sets of inconsistent assumptions at play here. On the one hand, the assumption of an end-to-end architecture leads an application designer to assume that an IP session opened with a remote peer will indeed be with that remote peer, and not with some intercepting network-level proxy agent attempting to mimic the behavior of that remote peer. On the other hand, is the assumption that transactions adhere to a consistent and predictable protocol, and transactions may be intercepted and manipulated by middleware as long as the resultant interaction behaves according to the defined protocol.

Middleware Architecture

Are transparent caches good or bad? Is the entire concept of middleware good or bad?

There is no doubt that middleware can be very useful. Cache systems can create improved service quality and reduced cost. NATs can reduce the demand for public IP address space. Firewalls can be effective as security policy agents. Middleware can perform load balancing across multiple service points for a particular class of applications, such as a Web server farm. Middleware can dynamically adjust the Internal Protocol parameters of a TCP session to adapt to particular types of networks, or various forms of network service policies. Middleware can provide services within the network that relieve the end user of a set of tasks and responsibilities, and middleware can improve some aspects of the service quality. Middleware can make an Internet service faster, cheaper, more flexible, and more secure, although probably not all at the same time. But middleware comes at a steep long-term price.

The advantage of the Internet lies in its unique approach to network architecture. In a telephone network, the end device—a telephone handset—is a rather basic device consisting of a pair of transducers and a tone generator. All the functionality of the telephone service is embedded within the network itself.

The architecture of the Internet is the complete opposite. The network consists of a collection of packet switches with basic functionality. The service is embedded within the protocol stack and the set of applications that are resident on the connected device. Within this architecture, adding new services to the network is as simple as distributing new applications among those end systems that want to use the application. The network makes no assumptions about the services it supports, and network services can be added, refined, and removed without requiring any change to the network itself. This results in a cheap, flexible, and basic network, and it passes the entire responsibility for service control to the network users. The real strength of the Internet lies in its architectural simplicity and lack of complex interdependencies within the network.

Middleware cuts across this model by inserting directly into the network functionality that alters the behavior of the network. IP or TCP Packet Header fields may be altered on the fly, or, as with a transparent cache, middleware may intercept user traffic, use an application level interpreter to interpret the upper-level service request associated with the traffic, and generate a response, acting as an unauthorized proxy for the intended recipient. With middleware present in an IP network, sending a packet to an addressed destination and receiving a response with a source address of that destination is no guarantee that you have actually communicated with the addressed remote device. You may instead be communicating with a middleware box, or have had the middleware box alter your traffic in various ways that are not directly visible to the sender.

In such an environment, it's not just the end-user applications that define an Internet-deployed service, because middleware is also part of the Internet service architecture. Services may be deployed that are reliant on the existence of middleware to be effective. Streaming video services, for example, become far more viable as a scalable Internet service when the streaming video server content is replicated across a set of middleware streaming systems deployed close to end users of the service. To change the behavior of a service that has supporting middleware deployed requires the network middleware to be changed. A new service may not be deployed until the network middleware is altered to permit its deployment. Any application requiring actual end-to-end communications may have to have additional functionality to detect if there is network middleware deployed along the path, and then explicitly negotiate with this encountered middleware to ensure that its actual communication will not be intercepted and proxied or otherwise altered.

Conclusion

The cumulative outcome is that such a middleware-modified Internet service model is not consistent with an end-to-end architecture. It represents the introduction of a more muddled service architecture where the network may choose to selectively intervene in the interaction between one device and another. Such a network architecture may not have stable scaling properties. Such an architecture may not readily support entire classes of new applications and new services. Such an architecture may not be sufficiently flexible and powerful to underpin a ubiquitous global data communications system. All this middleware overhead makes applications more complex, makes the network more complex, and makes networking more expensive, more limited, and less flexible.

From this perspective, middleware is an unglamorous hack. To adapt a 350-year-old quote from Thomas Hobbes, middleware is nasty, brutish, and short-sighted. It is, hopefully, a temporary imposition on an otherwise elegant, simple, and adequate Internet architecture.[3, 4]

References

[1] Aiken, B. et.al, "Network Policy and Services: A Report of a Workshop on Middleware," RFC 2768, February 2000.

[2] Carpenter, B. ed., "Architectural Principles of the Internet," RFC 1958, June 1996.

[3] Hobbes, Thomas (1588–1679), *Leviathan,* London, 1651. Available from many sources, including, ISBN 0140431950, Penguin Press, 1982.

[4] **http://www.orst.edu/instruct/phl302/texts/hobbes/leviathan-contents.html**

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Scientist in the Internet area for Telstra. He is also a member of the Internet Architecture Board, and is the Secretary of the Internet Society Board of Trustees. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@telstra.net**

Book Review

Internetworking with TCP/IP

Internetworking with TCP/IP (Vol. 1): Principles, Protocols, and Architectures, Douglas E. Comer, ISBN 0-13-018380-6, Prentice Hall, 2000.

Internetworking With TCP/IP (Vol. 1): Principles, Protocols, and Architectures (fourth edition) is the latest update to Comer's landmark work containing *Internetworking With TCP/IP (Vol. 2): Design, Implementation, and Internals* and *Internetworking With TCP/IP (Vol. 3): Client-Server Programming and Applications/BSD Socket Version.* As a recent engineering graduate, I wish I had read this book sooner; it is very concise and would have saved me a lot of time early in my studies.

Comer imparts Volume 1 in four sections. The first section provides a basic introduction to general networking including descriptions of typical network components. This section is most helpful for the entry-level student or casual reader. Advanced readers may want to skip right to the next section of the text, which continues with coverage of the TCP/IP networking environment from the host's point of view. Here, the organization and operation of local host protocols, addressing, and routing are thoroughly discussed. After reading this portion of the book, you will definitely understand how your desktop computer communicates on the network. Next, the global Internet architecture is laid out in a very comprehensible format. The reader is introduced to router-to-router protocols and algorithms that don't seem so complicated after this treatment. Lastly, application-level services and the client-server model of networking are covered in the final portion of the book.

Classic Reference

When reviewing one of the eminent texts in the field, it is of limited use to comment on the work chapter by chapter. However, I am compelled to comment on the quality of Chapter 11, Protocol Layering. This chapter is particularly interesting because Comer directly compares the ISO 7-layer reference model to the TCP/IP 5-layer model. As is par for this book, the comparison is clear and concise. Furthermore, the advantages and disadvantages of protocol layering are discussed in general and a realistic perspective is provided with reference to actual software implementation practices which may result in layer blurring. This is a very cogent presentation of the interaction between theory and reality in engineering. Although covering a specific topic, it could easily serve as an object lesson in a discussion of "real world" engineering techniques. In addition to Chapter 11, the chapters covering Internet routing (14 through 16) really shine as mainstays of this book. The Internet is viewed from the top down and "big network" protocols such as the *Border Gateway Protocol* (BGP) are given good coverage. This is an area where very few people are completely comfortable and Comer once more brings the important material forward in an easily understandable fashion. In the following paragraphs, I will highlight some of the new material included in the fourth edition.

New TCP/IP Concepts

The book's handling of *Classless Inter-Domain Routing* (CIDR) is very informative. In addition to explaining the inner-workings of the address space, Comer points out the requirement for new routing algorithms. This is an associated cost of adopting this new concept that is often overlooked when CIDR is presented.

Two new and important IP topics are also well-presented. Comer begins his treatment of IP Version 6 (IPv6) with a quick history of the protocol and a review of the logic behind this change. The new address space notation and allocation by type are explained very well. New advantages provided under IPv6 protocol structures are then discussed. Additionally, Mobile IP concepts and practicalities are introduced. Comer does a good job of bringing out both good news and bad news of this crucial new networking technology.

Coverage of *Random Early Drop* (RED) was rather brief and really needs more detail before readers can thoroughly grasp the concept. However, this would require greater mathematical sophistication on the part of the reader. Accordingly, depth of coverage is forgone in the interest of readability.

The section on *Network Address Translation* (NAT) does not adequately explain the dynamic nature of IP address assignment across hosts and data flows. An additional detailed example would help here.

Multimedia

In the application-level services section of the book, Comer offers a hasty explanation of how voice and video are sent over IP internets and how IP Telephony operates. The H.323 protocol is briefly mentioned as the low-bandwidth videoconferencing standard. However, it is not presented in its full importance as an umbrella recommendation from the *International Telecommunications Union* (ITU). A chapter explaining the roles of subordinate H.320 protocols in general would be a welcome addition to this section. *Quality of Service* (QoS) concepts such as *Resource Reservation Protocol* (RSVP), *Differentiated Services* (DiffServ), and *Real Time Protocol* (RTP) are likewise given short rift. However, IP Multicast is given significant treatment in one of the book's longest chapters; its concepts, mechanics, and implementation choices are thoroughly addressed.

Security

The book provides clear introductions to *Virtual Private Networks* (VPNs) and the IPSec set of protocols. The actual mechanics of IPSec are detailed thoroughly. Various required algorithms are introduced and pertinent RFC references are pointed out. Finally, firewall basics and implementation issues are covered. Overall, these sections clearly define the pertinent security concepts and make them simple.

Prerequisite Knowledge

This book thoroughly covers the fundamental principles of network design including implementation trade-offs and their associated foibles. However, understanding this text requires little more than a modest understanding of basic computer and networking concepts. An introductory programming course that covers computer organization, the binary number system, and basic data structures should suffice. From this point, the student can use the text for initial network familiarization as well as a future reference to ground the more abstract topics in network design.

A Must-Have Reference

An extensive, concept-based overview of the TCP/IP internetworking protocols makes Comer's Volume 1 the classic introduction to TCP/IP. He makes this an enjoyable read by breaking the topic into short, digestible chapters. Additionally, Comer pauses throughout the text to intersperse review material. Recurrent, italicized summaries provide a significant advantage to the student. These asides concisely summarize key points and provide a coherent set of landmarks for quick review and study.

By itself, Volume 1 is broad enough to be complete as an introduction to IP networking protocols. Comer further extends the work by pointing the reader to very specific resources for in-depth information including web pages and specific RFC numbers for applicable topics at the end of each chapter. One of life's simple treasures is found in the *Guide to RFCs* (Appendix 1). Here, the first 2728 RFCs are organized by major categories and subtopics. At last, a navigable index of RFCs has been incorporated with a superb text from which the beginner can delve the body of networking knowledge.

—*Albert C. Kinney*
kinney@ieee.org

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Fragments

Jonathan B. Postel Service Award for 2001 Presented to Daniel Karrenberg

Internet Society (ISOC) Chairman Brian Carpenter presented the *2001 Jonathan Postel Service Award* to Mr. Daniel Karrenberg, one of the pioneers of the Internet's development in Europe, during the opening ceremony of the 2001 INET Conference. His early work was at the University of Dortmund creating a basic networked e-mail and USENET service. The success of this initiative was the seed on which the first pre-commercial network, EUnet, was built. As the Internet came to Europe in the late 1980s, Mr. Karrenberg was active in organizing the first RIPE meeting and in creating the RIPE NCC to serve as secretariat for the Internet community in Europe. The RIPE NCC became the first *Regional Internet Registry* as we know them, taking on address allocation as one of its core services. Daniel headed the effort from the start, working hard to maximize the benefit for the community.

Mr. Karrenberg humbly accepted the award, thanking the Internet community for this recognition and pledging to continue his work guided by the spirit of Jon Postel.

The Jonathan B. Postel Service Award was established by the Internet Society to honor a person who has made outstanding contributions in service to the data communications community. It is named for Dr. Jonathan B. Postel to recognize and commemorate the extraordinary stewardship exercised by Jon over the course of a thirty year career in networking. The Award consists of an engraved crystal globe and US $20,000.00. The first award was presented posthumously to Jon Postel himself, accepted by his mother, Lois Postel at INET '99. Scott Bradner received the second award during INET 2000. For additional information on Jon Postel's life and contributions, please visit:
http://www.isoc.org/postel/

RFC 1149 Implemented

The Internet Engineering Task Force (IETF) has a long tradition of publishing humorous *Request For Comments* (RFCs) each year on April 1st. One of the more famous such RFCs is "A Standard for the Transmission of IP Datagrams on Avian Carriers," RFC 1149, by David Waitzman, published on April 1, 1990. This "carrier pigeon" RFC was recently implemented by a group in Bergen, Norway. For details see:
http://www.blug.linux.no/rfc1149/

Jon Crowcroft Joins IPJ Editorial Advisory Board

We are pleased to announce that Dr. Jon Crowcroft of University College London has joined the Editorial Advisory Board for the *Internet Protocol Journal* (IPJ). Dr. Crowcroft has been working in the field of internetworking and protocol design since the early days of the ARPANET. For more information, see:
http://www.cs.ucl.ac.uk/staff/J.Crowcroft/

We would also like to thank Edward Kozel, the creator of IPJ, for his support and advice over the last three years. Mr. Kozel has left Cisco to pursue other interests.

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

September 2001 Volume 4, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Multiprotocol Label Switching (MPLS) is a technology that has received a great deal of attention in recent years. The IETF alone has produced over 300 Internet Drafts and numerous RFCs related to MPLS and continues its work on refining the standards. So, what is MPLS all about? We asked Bill Stallings to give us a basic tutorial.

The tragic events of September 11, 2001 have focused attention on the stability and robustness of the Internet. The Internet played an important role in the aftermath of the terrorist attacks. While popular news Web sites initially appeared overloaded, a great deal of private traffic in the form of instant messaging and e-mail took place. Companies directly or indirectly affected by the events in New York and Washington were quick to use the Web as a way to disseminate important information to their clients as well as to their employees. In many cases, the Internet was used in place of an overloaded telephone network. With this in mind, The *Internet Corporation for Assigned Names and Numbers* (ICANN) has decided to re-focus its next meeting to address issues of Internet stability and security, particularly with regard to naming and addressing. (See "Fragments," page 32.) To provide some background information, we bring you the article "A Unique, Authoritative Root for the DNS," by M. Stuart Lynn, the president and CEO of ICANN. Since this article has been posted for public comment, you are encouraged to address your feedback to: **comments@icann.org**

We would like to remind our readers to send us postal address updates. The computer-communications industry is one where people change jobs and locations often. While we do receive some address changes automatically when mail is returned to us, it is much more reliable to send us e-mail with the new information. In the near future, readers will be able to make address changes and select delivery options through a Web interface which will be deployed at **http://www.cisco.com/ipj**. Until then, please send your updates to **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

MPLS

by William Stallings

Multiprotocol Label Switching (MPLS) is a promising effort to provide the kind of traffic management and connection-oriented *Quality of Service* (QoS) support found in *Asynchronous Transfer Mode* (ATM) networks, to speed up the IP packet-forwarding process, and to retain the flexibility of an IP-based networking approach.

Background

The roots of MPLS go back to numerous efforts in the mid-1990s to combine IP and ATM technologies. The first such effort to reach the marketplace was IP switching, developed by Ipsilon. To compete with this offering, numerous other companies announced their own products, notably Cisco Systems (Tag Switching), IBM (aggregate route-based IP switching), and Cascade (IP Navigator). The goal of all these products was to improve the throughput and delay performance of IP, and all took the same basic approach: Use a standard routing protocol such as *Open Shortest Path First* (OSPF) to define paths between endpoints; assign packets to these paths as they enter the network; and use ATM switches to move packets along the paths. When these products came out, ATM switches were much faster than IP routers, and the intent was to improve performance by pushing as much of the traffic as possible down to the ATM level and using ATM switching hardware.

In response to these proprietary initiatives, the *Internet Engineering Task Force* (IETF) set up the MPLS working group in 1997 to develop a common, standardized approach. The working group issued its first set of Proposed Standards in 2001. Meanwhile, however, the market did not stand still. The late 1990s saw the introduction of many routers that are as fast as ATM switches, eliminating the need to provide both ATM and IP technology in the same network.

Nevertheless, MPLS has a strong role to play. MPLS reduces the amount of per-packet processing required at each router in an IP-based network, enhancing router performance even more. More significantly, MPLS provides significant new capabilities in four areas that have ensured its popularity: QoS support, traffic engineering, *Virtual Private Networks* (VPNs), and multiprotocol support. Before turning to the details of MPLS, we briefly examine each of these.

Connection-Oriented QoS Support

Network managers and users require increasingly sophisticated QoS support for numerous reasons. The following are key requirements:

- Guarantee a fixed amount of capacity for specific applications, such as audio/video conference
- Control latency and jitter and ensure capacity for voice
- Provide very specific, guaranteed, and quantifiable service-level agreements, or traffic contracts
- Configure varying degrees of QoS for multiple network customers

A connectionless network, such as in IP-based internetwork, cannot provide truly firm QoS commitments. A *Differentiated Service* (DS) framework works in only a general way and upon aggregates of traffic from numerous sources. An *Integrated Services* (IS) framework, using the *Resource Reservation Protocol* (RSVP), has some of the flavor of a connection-oriented approach, but is nevertheless limited in terms of its flexibility and scalability. For services such as voice and video that require a network with high predictability, the DS and IS approaches, by themselves, may prove inadequate on a heavily loaded network. By contrast, a connection-oriented network has powerful traffic-management and QoS capabilities. MPLS imposes a connection-oriented framework on an IP-based internet and thus provides the foundation for sophisticated and reliable QoS traffic contracts.

Traffic Engineering

MPLS makes it easy to commit network resources in such a way as to balance the load in the face of a given demand and to commit to differential levels of support to meet various user traffic requirements. The ability to dynamically define routes, plan resource commitments on the basis of known demand, and optimize network utilization is referred to as *traffic engineering*.

With the basic IP mechanism, there is a primitive form of automated traffic engineering. Specifically, routing protocols such as OSPF enable routers to dynamically change the route to a given destination on a packet-by-packet basis to try to balance load. But such dynamic routing reacts in a very simple manner to congestion and does not provide a way to support QoS. All traffic between two endpoints follows the same route, which may be changed when congestion occurs. MPLS, on the other hand, is aware of not just individual packets, but flows of packets in which each flow has certain QoS requirements and a predictable traffic demand. With MPLS, it is possible to set up routes on the basis of these individual flows, with two different flows between the same endpoints perhaps following different routers. Further, when congestion threatens, MPLS paths can be rerouted intelligently. That is, instead of simply changing the route on a packet-by-packet basis, with MPLS, the routes are changed on a flow-by-flow basis, taking advantage of the known traffic demands of each flow. Effective use of traffic engineering can substantially increase usable network capacity.

VPN Support

MPLS provides an efficient mechanism for supporting VPNs. With a VPN, the traffic of a given enterprise or group passes transparently through an internet in a way that effectively segregates that traffic from other packets on the internet, proving performance guarantees and security.

Multiprotocol Support

MPLS, which can be used on many networking technologies, is an enhancement to the way a connectionless IP-based internet is operated, requiring an upgrade to IP routers to support the MPLS features. MPLS-enabled routers can coexist with ordinary IP routers, facilitating the introduction of evolution to MPLS schemes. MPLS is also designed to work in ATM and Frame Relay networks. Again, MPLS-enabled ATM switches and MPLS-enabled Frame Relay switches can be configured to coexist with ordinary switches. Furthermore, MPLS can be used in a pure IP-based internet, a pure ATM network, a pure Frame Relay network, or an internet that includes two or even all three technologies. This universal nature of MPLS should appeal to users who currently have mixed network technologies and seek ways to optimize resources and expand QoS support.

For the remainder of this discussion, we focus on the use of MPLS in IP-based internets, with brief comments about formatting issues for ATM and Frame Relay networks.

MPLS Operation

An MPLS network or internet consists of a set of nodes, called *Label Switched Routers* (LSRs), that are capable of switching and routing packets on the basis of a label which has been appended to each packet. Labels define a flow of packets between two endpoints or, in the case of multicast, between a source endpoint and a multicast group of destination endpoints. For each distinct flow, called a *Forwarding Equivalence Class* (FEC), a specific path through the network of LSRs is defined. Thus, MPLS is a connection-oriented technology. Associated with each FEC is a traffic characterization that defines the QoS requirements for that flow. The LSRs do not need to examine or process the IP header, but rather simply forward each packet based on its label value. Therefore, the forwarding process is simpler than with an IP router.

Figure 1: MPLS Operation

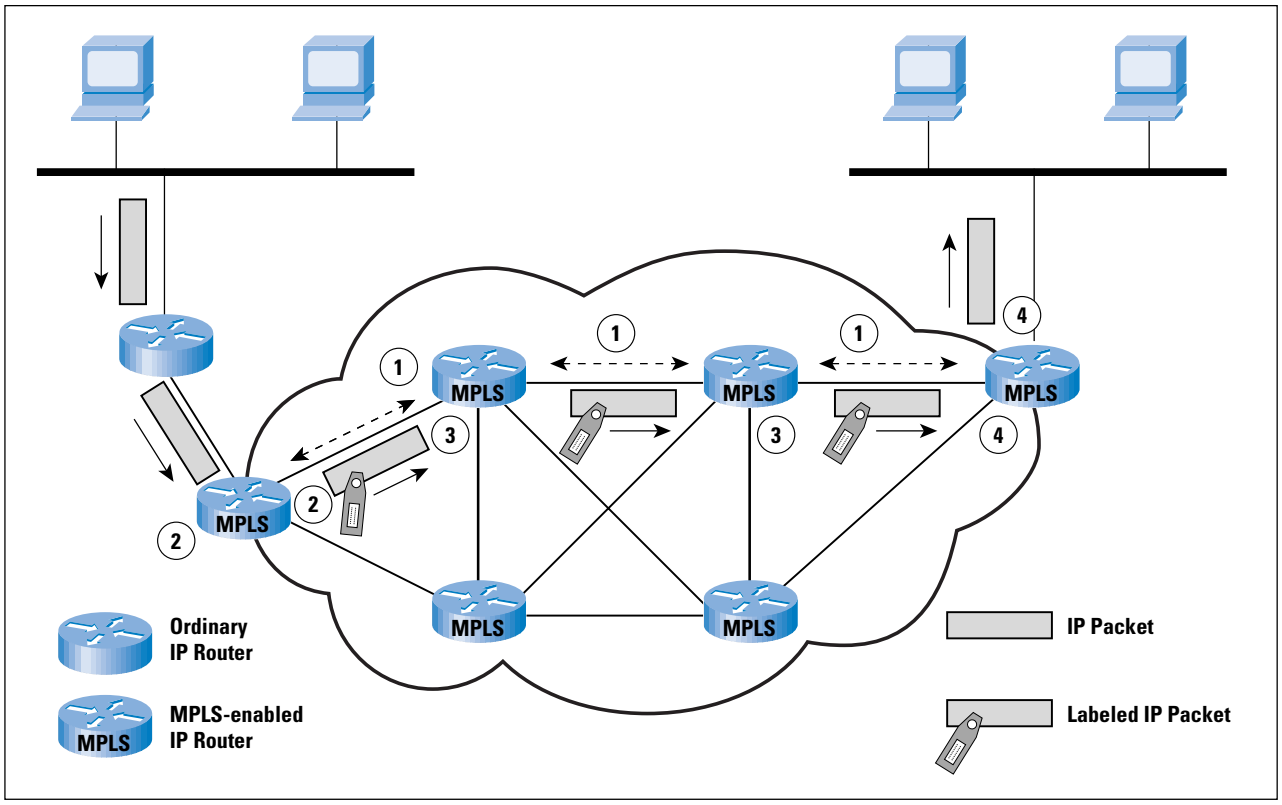


Figure 1, based on one in[4], depicts the operation of MPLS within a domain of MPLS-enabled routers. The following are key elements of the operation.

1. Prior to the routing and delivery of packets in a given FEC, a path through the network, known as a *Label Switched Path* (LSP), must be defined and the QoS parameters along that path must be established. The QoS parameters determine (1) how many resources to commit to the path, and (2) what queuing and discarding policy to establish at each LSR for packets in this FEC. To accomplish these tasks, two protocols are used to exchange the necessary information among routers:
   (a) An interior routing protocol, such as OSPF, is used to exchange reachability and routing information.
   (b) Labels must be assigned to the packets for a particular FEC. Because the use of globally unique labels would impose a management burden and limit the number of usable labels, labels have local significance only, as discussed subsequently. A network operator can specify explicit routes manually and assign the appropriate label values. Alternatively, a protocol is used to determine the route and establish label values between adjacent LSRs. Either of two protocols can be used for this purpose: the *Label Distribution Protocol* (LDP) or an enhanced version of RSVP.

2. A packet enters an MPLS domain through an ingress edge LSR where it is processed to determine which network-layer services it requires, defining its QoS. The LSR assigns this packet to a particular FEC, and therefore a particular LSP, appends the appropriate label to the packet, and forwards the packet. If no LSP yet exists for this FEC, the edge LSR must cooperate with the other LSRs in defining a new LSP.
3. Within the MPLS domain, as each LSR receives a labeled packet, it:
   (a) Removes the incoming label and attaches the appropriate outgoing label to the packet.
   (b) Forwards the packet to the next LSR along the LSP.
4. The egress edge LSR strips the label, reads the IP packet header, and forwards the packet to its final destination.

Several key features of MLSP operation can be noted at this point:

1. An MPLS domain consists of a contiguous, or connected, set of MPLS-enabled routers. Traffic can enter or exit the domain from an endpoint on a directly connected network, as shown in the upper-right corner of Figure 1. Traffic may also arrive from an ordinary router that connects to a portion of the internet not using MPLS, as shown in the upper-left corner of Figure 1.
2. The FEC for a packet can be determined by one or more of a number of parameters, as specified by the network manager. Among the possible parameters:
   - Source or destination IP addresses or IP network addresses
   - Source or destination port numbers
   - IP protocol ID
   - Differentiated services codepoint
   - IPv6 flow label
3. Forwarding is achieved by doing a simple lookup in a predefined table that maps label values to next-hop addresses. There is no need to examine or process the IP header or to make a routing decision based on destination IP address.
4. A particular *Per-Hop Behavior* (PHB) can be defined at an LSR for a given FEC. The PHB defines the queuing priority of the packets for this FEC and the discard policy.
5. Packets sent between the same endpoints may belong to different FECs. Thus, they will be labeled differently, will experience different PHB at each LSR, and may follow different paths through the network.

Figure 2: MPLS Packet Forwarding

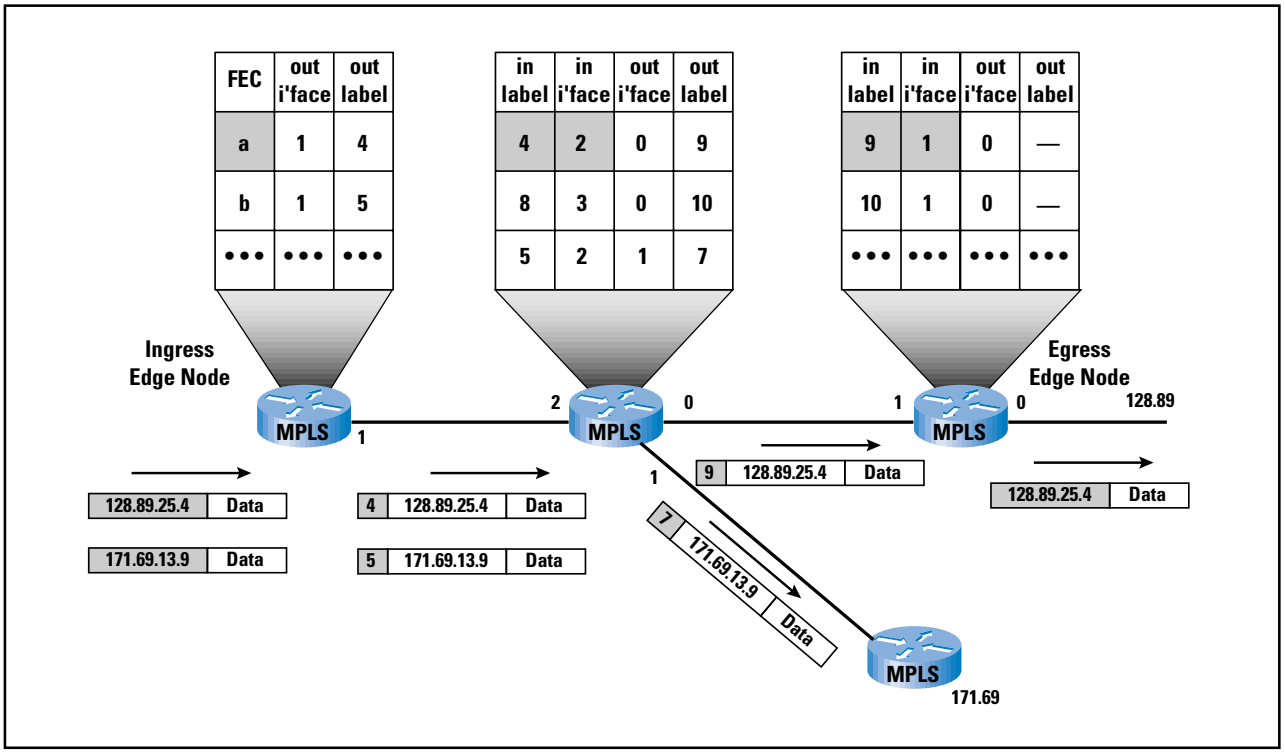


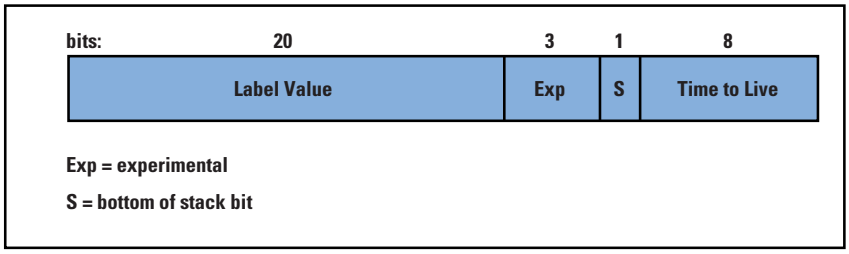
Figure 2 shows the label-handling and label-forwarding operation in more detail. Each LSR maintains a forwarding table for each LSP passing through the LSR. When a labeled packet arrives, the LSR indexes the forwarding table to determine the next hop. For scalability, as was mentioned, labels have local significance only. Thus, the LSR removes the incoming label from the packet and attaches the matching outgoing label before forwarding the packet. The ingress-edge LSR determines the FEC for each incoming unlabeled packet and, on the basis of the FEC, assigns the packet to a particular LSP, attaches the corresponding label, and forwards the packet.

Label Stacking

One of the most powerful features of MPLS is *label stacking*. A labeled packet may carry many labels, organized as a last-in-first-out stack. Processing is always based on the top label. At any LSR, a label may be added to the stack (push operation) or removed from the stack (pop operation). Label stacking allows the aggregation of LSPs into a single LSP for a portion of the route through a network, creating a *tunnel*. At the beginning of the tunnel, an LSR assigns the same label to packets from a number of LSPs by pushing the label onto the stack of each packet. At the end of the tunnel, another LSR pops the top element from the label stack, revealing the inner label. This is similar to ATM, which has one level of stacking (virtual channels inside virtual paths), but MPLS supports unlimited stacking.

Label stacking provides considerable flexibility. An enterprise could establish MPLS-enabled networks at various sites and establish numerous LSPs at each site. The enterprise could then use label stacking to aggregate multiple flows of its own traffic before handing it to an access provider. The access provider could aggregate traffic from multiple enterprises before handing it to a larger service provider. Service providers could aggregate many LSPs into a relatively small number of tunnels between points of presence. Fewer tunnels means smaller tables, making it easier for a provider to scale the network core.

Figure 3: MPLS Label Format



Label Format and Placement

An MPLS label is a 32-bit field consisting of the following elements (Figure 3):

- *Label value*: locally significant 20-bit label
- *Exp:* 3 bits reserved for experimental use; for example, these bits could communicate DS information or PHB guidance
- *S*: set to one for the oldest entry in the stack, and zero for all other entries
- *Time To Live* (TTL): 8 bits used to encode a hop count, or time to live, value

Time-to-Live Processing

A key field in the IP packet header is the TTL field (IPv4), or Hop Limit (IPv6). In an ordinary IP-based internet, this field is decremented at each router and the packet is dropped if the count falls to zero. This is done to avoid looping or having the packet remain too long in the internet because of faulty routing. Because an LSR does not examine the IP header, the TTL field is included in the label so that the TTL function is still supported. The rules for processing the TTL field in the label are as follows:

1. When an IP packet arrives at an ingress edge LSR of an MPLS domain, a single label stack entry is added to the packet. The TTL value of this label stack entry is set to the value of the IP TTL value. If the IP TTL field needs to be decremented, as part of the IP processing, it is assumed that this has already been done.

   When an MPLS packet arrives at an internal LSR of an MPLS domain, the TTL value in the top label stack entry is decremented.

Then:

(a) If this value is zero, the MPLS packet is not forwarded. Depending on the label value in the label stack entry, the packet may be simply discarded, or it may be passed to the appropriate "ordinary" network layer for error processing (for example, for the generation of an *Internet Control Message Protocol* [ICMP] error message).

(b) If this value is positive, it is placed in the TTL field of the top label stack entry for the outgoing MPLS packet, and the packet is forwarded. The outgoing TTL value is a function solely of the incoming TTL value, and is independent of whether any labels are pushed or popped before forwarding. There is no significance to the value of the TTL field in any label stack entry that is not at the top of the stack.

2. When an MPLS packet arrives at an egress edge LSR of an MPLS domain, the TTL value in the single label stack entry is decremented and the label is popped, resulting in an empty label stack. Then:

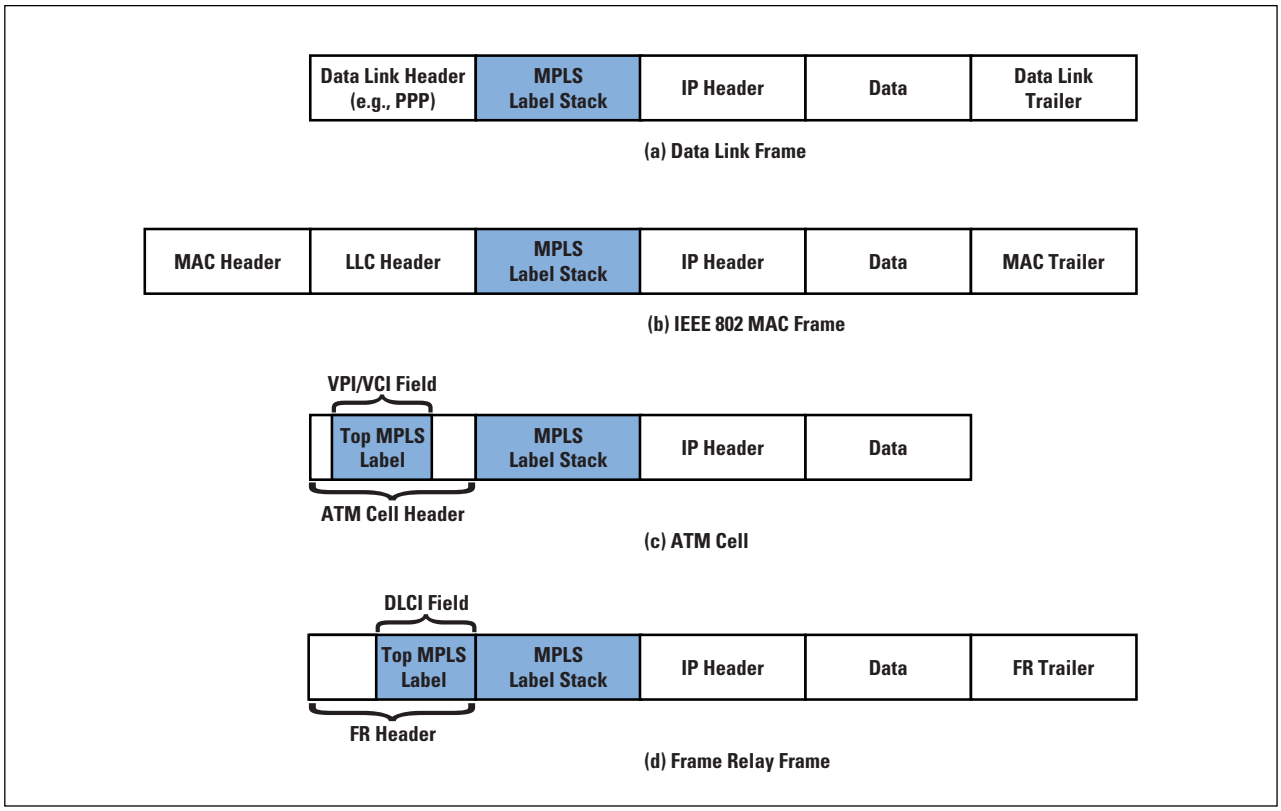
(a) If this value is zero, the IP packet is not forwarded. Depending on the label value in the label stack entry, the packet may be simply discarded, or it may be passed to the appropriate "ordinary" network layer for error processing.

(b) If this value is positive, it is placed in the TTL field of the IP header, and the IP packet is forwarded using ordinary IP routing. Note that the IP header checksum must be modified prior to forwarding.

Label Stack

The label stack entries appear after the data link layer headers, but before any network layer headers. The top of the label stack appears earliest in the packet (closest to the network layer header), and the bottom appears latest (closest to the data link header). The network layer packet immediately follows the label stack entry that has the *S* bit set. In a data link frame, such as for the *Point-to-Point Protocol* (PPP), the label stack appears between the IP header and the data link header (Figure 4a). For an IEEE 802 frame, the label stack appears between the IP header and the *Logical Link Control* (LLC) header (Figure 4b).

Figure 4: Position of MPLS Label



If MPLS is used over a connection-oriented network service, a slightly different approach may be taken, as shown in Figure 4c and d. For ATM cells, the label value in the topmost label is placed in the *Virtual Path/Channel Identifier* (VPI/VCI) field in the ATM cell header. The entire top label remains at the top of the label stack, which is inserted between the cell header and the IP header. Placing the label value in the ATM cell header facilitates switching by an ATM switch, which would, as usual, need to look only at the cell header. Similarly, the topmost label value can be placed in the *Data Link Connection Identifier* (DLCI) field of a Frame Relay header. Note that in both these cases, the TTL field is not visible to the switch and so is not decremented. The reader should consult the MPLS specifications for the details of the way this situation is handled.

FECs, LSPs, and Labels

To understand MPLS, it is necessary to understand the operational relationship among FECs, LSPs, and labels. The specifications covering all the ramifications of this relationship are lengthy. In the remainder of this section, we provide a summary.

The essence of MPLS functionality is that traffic is grouped into FECs. The traffic in an FEC transits an MPLS domain along an LSP. Individual packets in an FEC are uniquely identified as being part of a given FEC by means of a *locally significant label*.

At each LSR, each labeled packet is forwarded on the basis of its label value, with the LSR replacing the incoming label value with an outgoing label value.

The overall scheme described in the previous paragraph imposes numerous requirements. Specifically:

1. Traffic must be assigned to a particular FEC.
2. A routing protocol is needed to determine the topology and current conditions in the domain so that a particular LSP can be assigned to an FEC. The routing protocol must be able to gather and use information to support the QoS requirements of the FEC.
3. Individual LSRs must become aware of the LSP for a given FEC, must assign an incoming label to the LSP, and must communicate that label to any other LSR that may send it packets for this FEC.

The first requirement is outside the scope of the MPLS specifications. The assignment needs to be done either by manual configuration, by means of some signaling protocol, or by an analysis of incoming packets at ingress LSRs. Before looking at the other two requirements, let us consider the topology of LSPs. We can classify these in the following manner:

- *Unique ingress and egress LSR:* In this case a single path through the MPLS domain is needed.
- *Unique egress LSR, multiple ingress LSRs:* If traffic assigned to a single FEC can arise from different sources that enter the network at different ingress LSRs, then this situation occurs. An example is an enterprise intranet at a single location but with access to an MPLS domain through multiple MPLS ingress LSRs. This situation would call for multiple paths through the MPLS domain, probably sharing a final few hops.
- *Multiple egress LSRs for unicast traffic:* RFC 3031 states that most commonly, a packet is assigned to a FEC based (completely or partially) on its network layer destination address. If not, then it is possible that the FEC would require paths to multiple distinct egress LSRs. However, more likely, there would be a cluster of destination networks, all of which are reached via the same MPLS egress LSR.
- *Multicast:* RFC 3031 lists multicast as a subject for further study.

Route Selection

Route selection refers to the selection of an LSP for a particular FEC. The MPLS architecture supports two options: hop-by-hop routing and explicit routing.

With *hop-by-hop routing,* each LSR independently chooses the next hop for each FEC. The RFC implies that this option makes use of an ordinary routing protocol, such as OSPF.

This option provides some of the advantages of MPLS, including rapid switching by labels, the ability to use label stacking, and differential treatment of packets from different FECs following the same route. However, because of the limited use of performance metrics in typical routing protocols, hop-by-hop routing does not readily support traffic engineering or policy routing (defining routes based on some policy related to QoS, security, or some other consideration).

With *explicit routing*, a single LSR, usually the ingress or egress LSR, specifies some or all of the LSRs in the LSP for a given FEC. For strict explicit routing, an LSR specifies all of the LSRs on an LSP. For loose explicit routing, only some of the LSRs are specified. Explicit routing provides all the benefits of MPLS, including the ability to do traffic engineering and policy routing.

Explicit routes can be selected by configuration, that is, set up ahead of time, or dynamically. Dynamic explicit routing would provide the best scope for traffic engineering. For dynamic explicit routing, the LSR setting up the LSP would need information about the topology of the MPLS domain as well as QoS-related information about that domain. An MPLS traffic engineering specification[2] suggests that the QoS-related information falls into two categories:

- A set of attributes associated with an FEC or a collection of similar FECs that collectively specify their behavioral characteristics
- A set of attributes associated with resources (nodes, links) that constrain the placement of LSPs through them

A routing algorithm that accounts for the traffic requirements of various flows and the resources available along various hops and through various nodes is referred to as a *constraint-based routing algorithm*. In essence, a network that uses a constraint-based routing algorithm is aware of current utilization, existing capacity, and committed services at all times. Traditional routing algorithms, such as OSPF and the *Border Gateway Protocol* (BGP), do not employ a sufficient array of cost metrics in their algorithms to qualify as constraint-based.

Furthermore, for any given route calculation, only a single cost metric (for instance, number of hops, delay) can be used. For MPLS, it is necessary either to augment an existing routing protocol or to deploy a new one. For example, an enhanced version of OSPF has been defined[1] that provides at least some of the support required for MPLS. Examples of metrics that would be useful to constraint-based routing include the following:

- Maximum link data rate
- Current capacity reservation
- Packet loss ratio
- Link propagation delay

Label Distribution

Route selection consists of defining an LSP for an FEC. A separate function is the actual setting up of the LSP. For this purpose, each LSR on the LSP must:

1. Assign a label to the LSP to be used to recognize incoming packets that belong to the corresponding FEC.
2. Inform all potential upstream nodes (nodes that will send packets for this FEC to this LSR) of the label assigned by this LSR to this FEC, so that these nodes can properly label packets to be sent to this LSR.
3. Learn the next hop for this LSP and learn the label that the downstream node (LSR that is the next hop) has assigned to this FEC. This process will enable this LSR to map an incoming label to an outgoing label.

The first item in the preceding list is a local function. Items 2 and 3 must be done either by manual configuration or by using some sort of label distribution protocol. Thus, the essence of a label distribution protocol is that it enables one LSR to inform others of the label/FEC bindings it has made. In addition, a label distribution protocol enables two LSRs to learn each other's MPLS capabilities. The MPLS architecture does not assume a single label distribution protocol but allows for multiple such protocols. Specifically, RFC 3031 refers to a new label distribution protocol and to enhancements to existing protocols, such as RSVP and BGP, to serve the purpose.

The relationship between label distribution and route selection is complex. It is best to look at in the context of the two types of route selection.

With hop-by-hop route selection, no specific attention is paid to traffic engineering or policy routing concerns, as we have seen. In such a case, an ordinary routing protocol such as OSPF is used to determine the next hop by each LSR. A relatively straightforward label distribution protocol can operate using the routing protocol to design routes.

With explicit route selection, a more sophisticated routing algorithm must be implemented, one that does not employ a single metric to design a route. In this case, a label distribution protocol could make use of a separate route selection protocol, such as an enhanced OSPF, or incorporate a routing algorithm into a more complex label distribution protocol.

References

The two most important defining documents for MPLS are [5] and [6]. Reference [3] provides a thorough treatment of MPLS; [8] covers not only MPLS but other Internet QoS concepts; it includes an excellent chapter on MPLS traffic engineering. Reference [7] includes a concise overview of the MPLS architecture and describes the various proprietary efforts that preceded MPLS.

[1] Apostolopoulos, G., et al., "QoS Routing Mechanisms and OSPF Extensions," RFC 2676, August 1999.

[2] Awduche, D., et al. "Requirements for Traffic Engineering over MPLS," RFC 2702, September 1999.

[3] Black, U., *MPLS and Label Switching Networks,* ISBN 0130158232, Prentice Hall, 2001.

[4] Redford, R., "Enabling Business IP Services with Multiprotocol Label Switching," Cisco White Paper, July 2000 (**www.cisco.com**).

[5] Rosen, E., et al. "Multiprotocol Label Switching Architecture," RFC 3031, January 2001.

[6] Rosen, E., et al. "MPLS Label Stack Encoding," RFC 3032, January 2001.

[7] Viswanathan, A., et al., "Evolution of Multiprotocol Label Switching," *IEEE Communications Magazine,* May 1998.

[8] Wang, Z., *Internet QoS: Architectures and Mechanisms for Quality of Service,* ISBN 1558606084, Morgan Kaufmann, 2001.

Useful Web Sites

- *MPLS Forum:* An industry forum to promote MPLS: **http://www.mplsforum.org/**
- *MPLS Resource Center:* Clearinghouse for information on MPLS: **http://www.mplsrc.com/**
- *MPLS Working Group:* Chartered by IETF to develop standards related to MPLS. The Web site includes all relevant RFCs and Internet Drafts: **http://www.ietf.org/html.charters/mpls-charter.html**

WILLIAM STALLINGS is a consultant, lecturer, and author of over a dozen books on data communications and computer networking. He also maintains a computer science resource site for CS students and professionals at **WilliamStallings.com/StudentSupport.html.** He has a PhD in computer science from M.I.T. His latest book is *Wireless Communications and Networks* (Prentice Hall, 2001). His home in cyberspace is **WilliamStallings.com** and he can be reached at **ws@shore.net**

A Unique, Authoritative Root for the DNS

by M. Stuart Lynn, ICANN

The following *Internet Coordination Policy* (ICP) is being posted for the information of the Internet community by the *Internet Corporation for Assigned Names and Numbers* (ICANN) and is a statement of policy currently followed in administering the authoritative root of the Domain Name System. Comments on this article are welcome and should be directed to **comments@icann.org**

Abstract

This article reaffirms ICANN's commitment to a single, authoritative public root for the Internet *Domain Name System* (DNS) and to the management of that unique root in the public interest according to policies developed through community processes. This commitment is founded on the technical and other advice of the community and is embodied in existing ICANN policy.

The DNS is intended to provide a convenient means of referring to sites available on the Internet. By offering users an easy-to-use and reliable means of unambiguously referring to Web sites, e-mail servers, and the Internet's many other services, the DNS has helped the Internet achieve its promise as a global communications medium for commerce, research, education, and cultural and other expressive activities.

The DNS is a globally distributed database of domain name (and other) information. One of its core design goals is that it reliably provides the same answers to the same queries from any source on the public Internet, thereby supporting predictable routing of Internet communications. Achievement of that design goal requires a globally unique public name space derived from a single, globally unique DNS *root*.

Although the Internet allows a high degree of decentralized activities, coordination of the assignment function by a single authority is necessary where unique parameter values are technically required. Because of the uniqueness requirement, the content and operation of the DNS root must be coordinated by a central entity.

Where central coordination is necessary, it should be performed by an organization dedicated to serving the public interest and that acts according to policies developed through processes that are developed through the participation of affected stakeholders. Traditionally, the responsibility for performing the central coordinating functions of the global Internet for the public good, including management of the unique public DNS root, has been carried out by the *Internet Assigned Numbers Authority* (IANA)[12]. ICANN's core mission is to continue the work of the IANA in a more formalized and globally representative framework, to ensure the views of all the Internet's stakeholders are taken into account in carrying out this public trust.

Over the past several years, some private organizations have established DNS roots as alternates to the authoritative root. Some uses of these alternate roots do not jeopardize the stability of the DNS. For example, some are purely private roots operating inside institutions and are carefully insulated from the DNS. Others are purely experimental in the best traditions of the Internet and are carefully managed so as not to interfere with the operation of the DNS. These both operate within community-established norms.

Frequently, however, these alternate roots have been established to support top-level or pseudo-top-level domain name registries that are operated for profit. Yet other alternate roots have been established by certain individuals to protest the policies developed by the broader community processes for management of the authoritative root, or to express their disinterest in participating in those processes. These alternate roots have not been launched through any ICANN consensus processes, so they have not been entered into the authoritative root managed by the IANA or ICANN.

These alternate roots typically substitute insular concerns in place of the community-based processes that govern the management of the authoritative root. Their operators decide to include particular top-level domains in these alternate roots that have not been subjected to the tests of community support and conformance with consensus processes—coordinated by ICANN—that would allow their inclusion in the authoritative root. These decisions of the alternate-root operators have been made without any apparent regard for the fundamental public-interest concern of Internet stability. The widespread use of active domain names in these alternate roots could in fact impair the uniqueness of the authoritative name-resolution mechanism and hence the stability of the DNS.

ICANN's mandate to preserve stability of the DNS requires that it avoid encouraging the proliferation of these alternate roots that could cause conflicts and instability. This means that ICANN continues to adhere to community-based processes in its decisions regarding the content of the authoritative root. Within its current policy framework, ICANN can give no preference to those who choose to work outside of these processes and outside of the policies engendered by this public trust.

None of this precludes experimentation done in a manner that does not threaten the stability of name resolution in the authoritative DNS. Responsible experimentation is essential to the vitality of the Internet. Nor does it preclude the ultimate introduction of new architectures that may ultimately obviate the need for a unique, authoritative root. But the translation of experiments into production and the introduction of new architectures require community-based approaches, and are not compatible with individual efforts to gain proprietary advantage.

The Technical Need for a Single Authoritative Root

The DNS was originally deployed in the mid-1980s[13] as an improved means of mapping easy-to-remember names (i.e., **example.com)** to the IP addresses (i.e., **128.9.176.32**) by which packets are routed on the Internet. It is a distributed database that holds this mapping information (as well as various other types of technical information regarding computers on the Internet) in *resource records*. The DNS provides these resource records in response to queries it receives from programs called *resolvers* on individual computers throughout the Internet. The resolvers translate domain names into the corresponding IP addresses.

From the inception of the DNS, its most fundamental design goal has been to provide the same answers to the same queries issued from any place on the Internet. As stated in RFC 1034, the basic specification of the DNS's "Concepts and Facilities,"[16], "The primary (design) goal is a consistent name space which will be used for referring to resources." And as reiterated in RFC 2535, "Domain Name System Security Extensions,"[15] "It is part of the design philosophy of the DNS that the data in it is public and that the DNS gives the same answers to all inquirers."

The DNS is hierarchical. By design, the hierarchy begins with a group of *root nameservers* (often called simply *root servers*), which are specially-designated computers operated under common coordination that provide information about which other computers are authoritative regarding the top-level domains in the DNS naming structure. These set of root servers house the *authoritative root*. Thus, a resolver seeking information concerning a domain name such as **www.example.com** obtains one of the root servers' resource records about **.com**, which tells the resolver which computers have authoritative information about names within the **.com** top-level domain. The resolver then queries one of those authoritative **.com** nameservers about example.com, to locate the nameservers for **example.com**. A query is then made to one of those nameservers obtain the IP address of the computer designated by the name **www.example.com**.

The principal advantage of this hierarchical structure is that it allows different parts of the naming database to be maintained by different entities. According to the DNS's design, each domain was intended to be administered by a single entity.[19]

When the DNS was deployed in the mid-1980s, a set of root nameservers was designated and several top-level domains were established. These root nameservers (there are now 13 of them distributed around the world) are intended to provide authoritative information about which nameservers hold the naming information for each of the top-level domains. Since the authoritative root nameservers operate at the top of the hierarchy, resolvers find them by referring to IP addresses prestored at local computers throughout the Internet.

Over the past several years, some groups have established alternate root nameservers on the public Internet that distribute different information than the information distributed by the authoritative root nameservers. These groups then seek to persuade ISPs and Internet users to replace the pre-stored IP addresses of the authoritative root nameservers with those of their alternate servers. For a variety of reasons, these alternate roots have not to date achieved a significant level of usage on the public Internet.

Fortunately, the rare usage of alternate roots has thus far limited their practical effect on the Internet. If these alternate roots were to become prevalent, however, they would have the potential for seriously disrupting the reliable functioning of the DNS. Some of the consequences include:

- *Providing the Wrong Location:* The presence of alternate public DNS roots can result in different answers being given to the same DNS query issued from different computers on the Internet, depending on whether the inquiring computer is programmed to access the authoritative root or a particular one of the alternate roots (or more precisely a domain-name resolver associated with one or the other of these). The fundamental DNS design goal of providing consistent answers to DNS queries is therefore frustrated.[1]
- *Reaching the Wrong Computer:* The main consequence of such inconsistent data is that the same domain name can identify different computers depending on where the name is used. Put another way, *Uniform Resource Locators* (URLs) are no longer uniform. Thus, typing in a Web site address at two different computers configured to reference different roots can result in reaching different Web sites—a particularly disturbing possibility if, for example, money is to change hands or privacy or security concerns are violated. Similarly, the same piece of e-mail sent to the same address from the two computers can be directed to different recipients. The return of inconsistent DNS data defeats the globally consistent resolution of domain names that is vital to the Internet achieving its promise as a universal communications and applications medium for commerce, research, education, cultural exchange, expressive activities, and other uses.
- *Consequences Unpredictable to Most Users:* The set of DNS answers that will be received (from the authoritative root or one of the several alternate roots) is not predictable by most end users. Most users on the Internet employ a local DNS resolver that is configured by another person. Few users are likely to appreciate the significance of the resolver's DNS configuration; even fewer are likely to have detailed knowledge of that configuration. As the number of users on the Internet has grown, the proportion of users knowledgeable about technical concepts such as DNS resolvers and root servers has diminished. Yet these non-technical users are precisely those for whom the Internet in general—and the DNS in particular—hold the greatest potential benefits.

- *Intermediate Hosts Add to Confusion:* Moreover, some Internet services depend on the actions of DNS resolvers employed by intermediate hosts. Alternate roots introduce the possibility that the DNS answer obtained by the intermediate host alters the character of the service in an unexpected way. A similar phenomenon can occur where one user sends another a reference to a URL, such as an e-mail reply address or a link on a Web site. If the recipient of an e-mail or the visitor to the Web site is using a computer that employs a different DNS root than intended by the sender of the e-mail or the designer of the Web site, unexpected results are likely to occur. For example, the e-mail could end up with the wrong person.
- *Cache Poisoning:* Alternate roots also introduce the possibility of misdirected Internet activities due to the phenomenon known as cache poisoning. For performance reasons, the DNS design calls for resource records to be passed around among the nameservers on the Internet, so that a resolver can obtain quicker access to a local copy of the resource record. Because the DNS assumes a single-root system, resource records are not marked to distinguish them according to the root from which they emanate. Thus, the presence of alternate roots introduces the possibility that Internet activities by those intending to use the authoritative root could be misdirected by a stray resource record emanating from an alternate root. Indeed, some malicious hacking attacks have been based on this principle, prompting the *Internet Engineering Task Force* (IETF) to propose a series of not-yet-fully-implemented improvements known as *DNS-Security* or *DNSSec*.

(It should be noted that the original design of the DNS provided a way to operate alternate roots in a way that does not imperil stability. See "Experimentation" below for details.)

These potentially destructive effects of alternate roots have long been accepted by the vast majority of Internet engineers. Despite this broad-based recognition, some have sought to justify the alternate roots by downplaying these effects. In response, and to document what it referred to as "some of the problems inherent in a family of recurring technically naive proposals," in May 2000 the *Internet Architecture Board* (IAB)[14] issued RFC 2826, entitled "IAB Technical Comment on the Unique DNS Root." The IAB summarized its comments (in relevant part) as follows:

"Summary: To remain a global network, the Internet requires the existence of a globally unique public name space. The DNS name space is a hierarchical name space derived from a single, globally unique root. This is a technical constraint inherent in the design of the DNS. Therefore it is not technically feasible for there to be more than one root in the public DNS. That one root must be supported by a set of coordinated root servers administered by a unique naming authority.

> "Put simply, deploying multiple public DNS roots would raise a very strong possibility that users of different ISPs who click on the same link on a Web page could end up at different destinations, against the will of the Web page designers."

For some concrete examples of potential failures and instabilities that would likely result from alternate roots prevalently used on the public Internet, see the draft "Alt-Roots, Alt-TLDs."[17]

In the face of the destabilizing consequences of alternate roots, as articulated by the IAB and others, ICANN's prime directive of preserving the stability of the Internet and DNS requires an unwavering commitment to promote the continued prevalence of a single authoritative root for the public DNS. Any other course of action by ICANN would be irresponsible.

The Public Trust in Coordinated Assignment Functions

The Internet's proper operation requires assignment of unique values to various identifiers for different computers or services on the Internet. To be effective, these assigned values must be made broadly available and their significance must be respected by the many people responsible for the Internet's operation. For example, every computer on the public Internet is assigned a unique IP address; this address is made known to routers throughout the Internet to cause TCP/IP packets with that destination address to be routed to the intended computer. Without common agreement to respect the assignment, the Internet would not reliably route communications to their intended destinations.

Beginnings to 1998: Central Coordination as a Public Trust

From the very beginnings of the Internet, the technical community has recognized the need for central coordination of the unique assignment of the values of identifiers. The IANA, now operated by ICANN was created to fill this need; it now makes assignments of unique values for approximately 120 different identifier types. This responsibility has always been understood to be a public trust, and the IANA long ago adopted the motto: "Dedicated to preserving the central coordinating functions of the global Internet for the public good."

The most commonly known of the Internet's uniquely assigned identifiers, of course, are domain names. From the time the DNS was deployed, the Internet community made the IANA "responsible for the overall coordination and management of the Domain Name System (DNS), and especially the delegation of portions of the name space called top-level domains."[18] As in its other assignment responsibilities, the IANA's role is to act in the public interest, neutrally, and without proprietary motives.

Competition as a Value Guiding the Internet's Technical Management

In the Internet's early years, with limited exceptions day-to-day registration activities for domain names were done by a single company (first SRI International and later Network Solutions) under the IANA's guidance.

By the mid-1990s, however, the growth and increasing commercialization of the Internet led the U.S. Government's Green[2] and White[3] Papers to note the emergence of "widespread dissatisfaction about the absence of competition in domain name registration." This dissatisfaction prompted the Green and White Papers to include the promotion of competition in registration services as one of the four values (stability; competition; private, bottom-up coordination; and representation) that should guide the Internet's technical management. Both documents made clear that, of these four values, preservation of stability was to be paramount.

Building on the IANA model of a non-profit entity carrying the public trust to perform the vital central coordination functions, the U.S. Government reconciled the need to ensure Internet stability with the desire to introduce competitive domain-name registration services as follows:

> "In keeping with these principles, we divide the name and number functions into two groups, those that can be moved to a competitive system and those that should be coordinated. We then suggest the creation of a representative, not-for-profit corporation to manage the coordinated functions according to widely accepted objective criteria. We then suggest the steps necessary to move to competitive markets in those areas that can be market driven." [4]

This dichotomy recognizes that the Internet is, after all, a network (albeit a network of networks), and networks require coordination among their participants to operate in a stable and efficient manner. It also reflects the phenomenal success of the Internet's tradition of cooperatively developed open and non-proprietary standards. Those standards have provided an environment of highly interoperable systems that has allowed competition and innovation to flourish.

ICANN Assumes the Public Trust

After public comment on the Green Paper, the United States Government issued the White Paper, which laid out the basic charter on which ICANN was founded and continues to operate. The White Paper re-emphasized the prime directive of stability and, to that end, the need to avoid creation of alternate roots:

> "The introduction of a new management system should not disrupt current operations or create competing root systems. During the transition and thereafter, the stability of the Internet should be the first priority of any DNS management system." [5]

The United States Government then invited the Internet community to form a not-for-profit corporation to perform the "coordinated functions" that should be handled as a matter of public trust, rather than according to a competitive regime that would not be conducive to stability. Among the "coordinated functions" were management of the root-server system and decisions to introduce new TLDs:

> "Similarly, coordination of the root server network is necessary if the whole system is to work smoothly. While day-to-day operational tasks, such as the actual operation and maintenance of the Internet root servers, can be dispersed, *overall policy guidance and control of the TLDs and the Internet root server system should be vested in a single organization* that is representative of Internet users around the globe.
>
> "Further, changes made in the administration or the number of gTLDs contained in the authoritative root system will have considerable impact on Internet users throughout the world. In order to promote continuity and reasonable predictability in functions related to the root zone, the *development of policies for the addition, allocation, and management of gTLDs and the establishment of domain name registries and domain name registrars to host gTLDs should be coordinated.*" [6]

In response to this invitation for the formation of a non-profit, Internet-community-based organization, ICANN was established in 1998. ICANN was subsequently selected by the United States Government from among several proposals submitted precisely because it was open, consensus-based, and rooted in the Internet community. The establishment of ICANN had followed extensive dialogs among different constituencies of the Internet community to ensure that ICANN could be responsive to the needs of these various constituencies.

ICANN, among its other responsibilities, now acts as the coordinator for operation of the authoritative root-server system and the policy forum for decisions about the policies governing what TLDs are to be included in the authoritative DNS root. [7]

In linking the formation of ICANN to the global Internet community, the White Paper established a public trust that required that the DNS be administered in the public interest as the unique-rooted,[8] authoritative database for domain names that provides a stable addressing system for use by the global Internet community. This commitment to a unique and authoritative root is a key part of the broader public trust—to carry out the Internet's central coordination functions for the public good—that is ICANN's reason for existence.

The Public Trust and the Introduction of New TLDs

It is essential that the centrally coordinated functions be performed in the public interest, not out of proprietary or otherwise self-interested motives. For this reason, ICANN was founded as a not-for-profit public-benefit organization, accountable to the Internet community. Longstanding Internet principles also require that the policies guiding the coordinated functions be established openly based on community deliberation and input. For these reasons ICANN's structure is representative of the geographic and functional diversity of the Internet, and relies to the extent possible on private-sector, bottom-up methods.

As the White Paper emphasized, the decisions about the introduction of new TLDs are appropriately done within this open, non-proprietary, and broadly representative framework, rather than by individuals or entities not accountable to the community and that ordinarily act for their own proprietary motives:

> "As Internet names increasingly have commercial value, the decision to add new top-level domains cannot be made on an ad hoc basis by entities or individuals that are not formally accountable to the Internet community." [9]

Within the framework of its commitment to a unique root system and to the stability of the Internet, last year ICANN launched a process for carefully introducing several new generic TLDs to the DNS. This introduction was fashioned as a proof of concept of the technical and business feasibility of introducing more TLDs into the DNS. Proceeding with an initial proof of concept was in response to the advice of ICANN's *Protocol Supporting Organization* (PSO) and its *Domain Name Supporting Organizatio*n (DNSO) to proceed cautiously and in an orderly fashion. The PSO and the DNSO represent the consensus views of the technical and the user/business/other institutional communities, respectively. Generic TLDs had not been introduced for many years, and there were and still are serious questions as to what the effect of introducing new TLDs will be on the stability and reliability of the DNS; and many questions about what should be the appropriate contractual and business context.

In response to an issued RFP, forty-seven institutions and groups submitted proposals for the establishment of new TLDs. They chose to work within the community-based ICANN process, even though they knew that only a "limited number" of TLDs would be selected—at least in the first round. In fact, seven were selected, and, following a methodology which allowed for considerable community input, contracts have or will shortly be signed with these initial seven. ICANN looks forward to the successful introduction of these new TLDs and will work with the community to monitor their performance so that a community decision can be made on moving forward with the introduction of more TLDs, should this be the conclusion of the proof of concept.

Outside the Process

Some private organizations have established DNS roots as alternates to the authoritative root. Some uses of these alternate roots do not jeopardize the stability of the DNS. For example, many are purely private roots operating inside institutions and are carefully insulated from the DNS. Others are purely experimental in the best traditions of the Internet and are carefully managed so as not to interfere with the operation of the DNS. These both operate within community-established norms.

Frequently, however, these alternate roots have been established to support top-level or pseudo-top-level domain name registries that are operated for profit. Yet other alternate roots have been established by certain individuals to protest the policies developed by the broader community processes for management of the authoritative root, or to express their disinterest in participating in those processes. These alternate roots have not been launched through any ICANN consensus processes, so they have not been entered into the authoritative root managed by the IANA or ICANN.

These alternate roots typically substitute insular concerns in place of the community-based processes that govern the management of the authoritative root. Their operators decide to include particular top-level domains in these alternate roots that have not been subjected to the tests of community support and conformance with consensus processes—coordinated by ICANN—that would allow their inclusion in the authoritative root. These decisions of the alternate root operators have been made with no apparent regard for the fundamental public-interest concern of Internet stability. The widespread introduction of active domain names into these alternate roots could in fact impair the uniqueness of the authoritative name resolution mechanism and hence the stability of the DNS.

In fact, some of the operators of these alternate roots state that stability is not an important attribute for the DNS. This thesis, for reasons already stated, is at fundamental variance with ICANN policy as embodied in its founding documents. Some of these operators and their supporters assert that their very presence in the marketplace gives them preferential right to TLDs to be authorized in the future by ICANN. They work under the philosophy that if they get there first with something that looks like a TLD and invite many registrants to participate, then ICANN will be required by their very presence and force of numbers to recognize in perpetuity these pseudo TLDs, inhibiting new TLDs with the same top-level name from being launched through the community's processes.

No current policy allows ICANN to grant such preferential rights. To do so would effectively yield ICANN's mandate to introduce new TLDs in an orderly manner in the public interest to those who would simply grab all the TLD names that seem to have any marketplace value, thus

circumventing the community-based processes that ICANN is required to follow. For ICANN to yield its mandate would be a violation of the public trust under which ICANN was created and under which it must operate. Were it to grant such preferential rights, ICANN would abandon this public trust, rooted in the community, to those who only act for their own benefit. Indeed, granting preferential rights could jeopardize the stability of the DNS, violating ICANN's fundamental mandate.

Alternate roots inherently endanger DNS stability—that is, they create the real risk of name resolvers being unable to determine to which numeric address a given name should point. This violates the fundamental design of the DNS and impairs the Internet's utility as a ubiquitous global communications medium. Some of these alternate systems also employ special technologies that—ingenious as they may be—may conflict with future generations of community-established Internet standards. Indeed, can there be any guarantee that these proprietary technologies can or will be adapted to future changes in Internet standards?

Experimentation

Experimentation has always been an essential component of the Internet's vitality. Working within the system does not preclude experimentation, including experimentation with alternate DNS roots. But these activities must be done responsibly, in a manner that does not disrupt the ongoing activities of others and that is managed according to experimental protocols.

DNS experiments should be encouraged. Experiments, however, almost by definition have certain characteristics to avoid harm: (a) they are clearly labeled as experiments, (b) it is well understood that these experiments may end without establishing any prior claims on future directions, (c) they are appropriately coordinated within a community-based framework (such as the IETF), and (d) the experimenters commit to adapt to consensus-based standards when they emerge through the ICANN and other community-based processes. This is very different from launching commercial enterprises that lull users into a sense of permanence without any sense of the foregoing obligations or contingencies.

Moreover, it is essential that experimental operations involving alternate DNS roots be conducted in a controlled manner, so that they do not adversely affect those who have not consented to participate in them. Given the design of the DNS, and particularly the intermediate-host and cache poisoning issues described earlier, special care must be taken to insulate the DNS from the alternate roots' effects. For example, alternate roots are commonly operated by large organizations within their private networks without harmful effects, since care is taken to prevent the flow of the alternate resource records onto the public Internet.

It should be noted that the original design of the DNS provides a facility for future extensions that accommodates the possibility of safely deploying multiple roots on the public Internet for experimental and other purposes. As noted in RFC 1034, the DNS includes a "class" tag on each resource record, which allows resource records of different classes to be distinguished even though they are commingled on the public Internet. For resource records within the authoritative root-server system, this class tag is set to "IN"; other values have been standardized for particular uses, including 255 possible values designated for "private use" that are particularly suited to experimentation.[10]

As described in a recent proposal within the IETF,[11] this "class" facility allows an alternate DNS namespace to be operated from different root servers in a manner that does not interfere with the stable operation of the existing authoritative root-server system. To take advantage of this facility, it should be noted, requires the use of client or applications software developed for the alternate namespace (presumably deployed after responsible testing), rather than the existing software that has been developed to interoperate with the authoritative root. Those who operate alternate roots for global commercial purposes, however, have not followed this course.

In an ever-evolving Internet, ultimately there may be better architectures for getting the job done where the need for a single, authoritative root will not be an issue. But that is not the case today. And the transition to such an architecture, should it emerge, would require community-based approaches. In the interim, responsible experimentation should be encouraged, but it should not be done in a manner that affects those who do not consent after being informed of the character of the experiment.

Conclusion

The success of the Internet and the guarantee of Internet stability rest on the cooperative activities of thousands, even millions, of people and institutions collaborating worldwide towards a common end. This extraordinary—even unprecedented—community effort has served to impel the incredible growth of the Internet. Many of these people and institutions compete intensely among themselves yet agree to do so within a common framework for the overall public good. Their collective efforts provide a policy framework for technical and entrepreneurial innovation, and the advancement of economic, social, and educational goals.

Most members of the global community and most institutions with which they are associated recognize that it is in their best long-term interests to work within these community-based processes, even if that means foregoing short-term advantages to particular individuals or groups. The over-arching principles outlined in this document override exclusive and narrowly focused self-interest.

Community-based policy development is not perfect. It may proceed slower than some would wish. The introduction of new TLDs has proceeded at deliberate speeds. Impatience in the context of Internet timescales is perfectly understandable. The outcome of orderly processes based on the wishes of the community, however, is assurance that the Internet will continue to function in a stable and holistic manner that benefits the global community, and not become captured by the self-interests of the few. That, in the minds of most, is a price worth paying.

ICANN—in deference to its public trust—will continue to collaborate with these citizens of the Internet community to advance the notions of a unique root system as a prerequisite to Internet stability, and to ensure that community-based policies take precedence. ICANN encourages responsible experimentation designed to further advance the Internet as a useful, stable, and accessible medium for the public good.

References

[1] Ironically, to avoid name conflicts in a multi-root system, a single-root system would need to be created-adding a higher level to the hierarchy.

[2] "Improvement of Technical Management of Internet Names and Addresses," (Green Paper), 63 *Federal Register* 8825, 8827 (20 February, 1998).

[3] "Management of Internet Names and Addresses," (White Paper), 63 *Federal Register* 31741, 31742 (10 June, 1998).

[4] Green Paper, 63 *Federal Register* at 8827.

[5] White Paper, 63 *Federal Register* at 31749. The Green and White Papers both made additional references to the need for a single authoritative root system. For example, in response to comments received from the Green Paper, the White Paper notes:

"In the absence of an authoritative root system, the potential for name collisions among competing sources for the same domain name could undermine the smooth functioning and stability of the Internet."

[6] White Paper, 63 *Federal Register* at 31749 (emphasis added).

[7] ICANN's corporate charter emphasizes its role in overseeing operation of the unique DNS root:

"... the Corporation shall ... pursue the charitable and public purposes ... of promoting the global public interest in the operational stability of the Internet by ... (iv) overseeing operation of the authoritative Internet DNS root server system ..."

ICANN Articles of Incorporation, para. 3. The phrase "the authoritative Internet DNS root server system" is decidedly in the *singular*.

See: **`http://www.icann.org/general/articles.htm`**

[8] The Memorandum of Understanding between the United States Government and ICANN that governs the transfer of responsibilities from the U.S. Department of Commerce to ICANN also makes reference to the authoritative root in the singular, not in the plural:

"In the DNS Project, the parties will jointly design, develop, and test the mechanisms, methods, and procedures to carry out the following DNS management functions: ...

"b. Oversight of the operation of the authoritative root server system;

"c. Oversight of the policy for determining the circumstances under which new top level domains would be added to the root system ... "

See also: **www.icann.org/general/icann-mou-25nov98.htm**

[9] White Paper, 63 *Federal Register* at 31742.

[10] Eastlake, D., Brunner-Williams, E., Manning, B., "Domain Name System (DNS) IANA Considerations," section 3.2, RFC 2929, September, 2000.

[11] Klensin, J., "Internationalizing the DNS—A New Class," Internet Draft, work in progress, December, 2000.

[12] Internet Assigned Numbers Authority (IANA). See **www.iana.org**

[13] Postel, J., "Domain Name System Implementation Schedule—Revised," RFC 921, October 1984.

[14] Internet Architecture Board (IAB). See **http://www.iab.org**

[15] Eastlake, D., "Domain Name System Security Extensions," RFC 2535, March 1999.

[16] Mockapetris, P., "Domain Names—Concepts and Facilities," RFC 1034, November 1987.

[17] **http://www.icann.org/stockholm/draft-crispin-alt-roots-tlds-00.txt**

[18] Postel, J., "Domain Name System Structure and Delegation," RFC 1591, March 1994.

[19] Postel, J., and Reynolds, J., "Domain Requirements," RFC 920, October 1984.

Dr. M. STUART LYNN is President & CEO of The Internet Corporation for Assigned Names and Numbers (ICANN). Dr. Lynn has had a distinguished career in computing and information technology that dates back almost four decades. His most recent position until his retirement in 1999 was as Associate Vice President for Information Resources and Communications for the University of California Office of the President where he served as chief information officer for the combined University of California system. Dr. Lynn also served as President and Chairman of the Board of the Corporation for Education Network Initiatives in California (CENIC). Dr. Lynn has also held positions at Cornell University, UC Berkeley, Rice University, Baylor College of Medicine, IBM and Chevron. Over the course of his career, he has been active in several professional organizations including the Association for Computing Machinery (ACM) and the American Federation of Information Processing Societies. In 1994, he was elected a Fellow of the ACM. In addition, he has served on numerous boards of directors, advisory committees and as a consultant to academia, government and industry. Dr. Lynn holds a M.A. and Ph.D. in Mathematics from the University of California at Los Angeles and a B.A. and M.A. in Mathematics from Oxford University.
E-mail: **lynn@icann.org**

Book Review

Web Protocols and Practice

Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement, by Balachander Krishnamurty and Jennifer Rexford, ISBN 0-201-71088-9, Addison-Wesley, 2001.

If you want to know something about the underlying workings of the Web, you can find it somewhere out there on the Web itself. But, as we all know, it is not always easy to find the page you want, and particularly not if you are in a hurry and don't want to have to wade through documentation hierarchies or download PDF files. In these cases a real book is unbeatable, if one is available. Sadly, for information about the lower reaches of Web protocols there has been no single useful printed reference source available.

Organisation

This book fills that gap. It provides a detailed look at all the low level protocol issues as well as many other things; the book's subtitle sums it up admirably. The first section provides a brief history of the Web and its development which introduces all the important terminology and, most importantly, also says what the book is *not* about: nothing on XML (hurrah!), HTML, scripting languages, administration of Web servers, or specific products.

Section two moves on to more technical matters looking at Web clients, proxies and servers. The client chapter has a particularly useful section on spiders with an excellent table showing the names and calling hosts of the commonest spider programs. The information about proxies and servers is also of high quality and provide a solid grounding in how they interact with each other and the potential problems that can arise.

The third section looks at the protocols involved when using the Web. Starting with a concise run through TCP and the use of the DNS, the authors then glance at FTP, SMTP and NNTP, before going to a detailed examination of HTTP/1.1. In my personal experience, information on HTTP/1.1 has always been particularly inaccessible, both from the point of view of discoverability and readability, and this chapter explained several things that I had been puzzled about, especially about cache control which is rather a black art. (Also featured is a comprehensive table of HTTP return codes to which I shall turn quite often.) To finish this section of the book, there is a chapter on how HTTP interacts with TCP—a whole area that I had never really thought about before and which is much more complex that I would have thought it to be.

Next is a short section devoted to measuring and characterizing Web traffic. This a hugely contentious area and the discussion is well balanced and sensible. Following this the authors look in more detail at caching and at multimedia streaming, and manage to cover the latter topic without going into much unnecessary details about the actual bits that get sent whilst still giving a good coverage of the important material.

To round off the book, there are three chapters devoted to research topics, looking again at caching, measurement and protocol issues. Much of the material here is not directly of relevance to someone who is dealing with Web protocols on a daily basis, but there is still much here that will be of interest as the authors draw attention to places where improvements can be expected and how these might be realised.

Excellent Book

As you might expect, there is also a comprehensive bibliography and index. All in all an excellent book that is well researched, well written, and clearly set out without the excess of white space that is so common in computing books today. The price is perhaps rather high (I certainly could not recommend this as a textbook to my students—they simply could not afford it), but for people working in the industry it would be a worthwhile purchase and I think that they would soon find it an indispensable source of reference.

—Lindsay Marshall, University of Newcastle upon Tyne
Lindsay.Marshall@ncl.ac.uk

Summary of Acronyms

DNS: *Domain Name System*

FTP: *File Transfer Protocol*

HTTP: *HyperText Transfer Protocol*

NNTP: *Network News Transfer Protocol*

PDF: *Portable Document Format*

SMTP: *Simple Mail Transfer Protocol*

TCP: *Transmission Control Protocol*

XML: *Extensible Markup Language*

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

Next ICANN Meeting, Marina del Rey, November 13–15, 2001

Many members of the *Internet Corporation for Assigned Names and Numbers* (ICANN) community wrote in response to a call for input as to whether the events of September 11 would affect their plans to travel to Los Angeles in November to attend the scheduled ICANN meetings. Almost without exception the respondents emphatically encouraged ICANN to hold its meetings and stated unequivocally that they planned to attend unless the international situation deteriorated to where travel was not practical.

Given this response and given the need to address emerging priorities, ICANN is planning to proceed with its November meeting, subject to any further serious change in the international situation that would affect travel conditions. However, as discussed below, the format of the meeting will differ significantly from what had previously been announced.

The events of September 11 have caused institutions worldwide to rethink their priorities and plans. As an international institution, ICANN is not immune. Although those events raise logistical and other concerns for holding meetings, they also underscore the need to address Internet stability issues, and security as a key component of stability. ICANN is not responsible for the overall security of the Internet. However, given ICANN's global responsibilities for the stability of the Internet's naming and addressing systems and under the new circumstances facing the international community, it would be irresponsible for ICANN not to conduct an in depth assessment of the robustness and security of these systems, and to take steps, if necessary, to strengthen the Internet in these regards. These are urgent matters and of worldwide importance.

The Internet is global in reach, as are the threats of terrorism. The events of September 11 offered a stark and tragic reminder of the incalculable importance of a reliable and secure naming and addressing system to support emergency response, personal and other communications, and information sharing. E-mail, instant messaging, and the Web, for example, all played essential roles.

Accordingly, the November ICANN meetings will focus on stability and security of the Internet's naming and addressing systems and of their operational implementation globally. This will be the overriding imperative for the meeting. As such, this will be a very different kind of meeting than previous ICANN meetings and will not follow the usual format.

At this meeting, ICANN will be seeking to promote discussion throughout the community on how to reassess areas of potential threats that could affect services within the scope of ICANN's responsibilities, how to improve readiness to meet these threats, and what additional policies or other actions should be considered and implemented to facilitate such improvements.

Clearly not all these questions will be answered in one meeting, but ICANN must now devote its energies as members of the global Internet community towards obtaining answers. Every constituency and supporting organization will be asked to report on its efforts to ensure the stability of the Internet's naming and addressing systems and what additional steps it proposes to take to improve that stability and security among its member organizations. Agenda items will be assessed for inclusion by what they contribute to the overall focus of the meeting.

Although a precise schedule has not yet been mapped out, these meetings will last three days from November 13 through 15, inclusive. Constituencies and supporting organizations will be asked to meet during this time to focus on the topic of the meeting. There will be a Board meeting at the end of the meeting to address essential business. The Board agenda will concentrate on topics where time is of the essence.

The focus of the meetings may well delay progress on some of the worthy and important initiatives that are currently underway. The effects of such delays have to be measured against the importance of ensuring the stability and security of the Internet itself. This will require patience on the part of those who may experience delays in matters of importance to them so that the ICANN community can bear down on the issue at hand.

This is only a preliminary announcement to enable attendees to firm up their travel plans. Details of the meeting will be announced as soon as possible. Please visit the ICANN Web site (**`http://www.icann.org`**) for further updates.

Van Jacobson Receives 2001 ACM SIGCOMM Award

Van Jacobson, the man widely credited with saving the Internet from an otherwise inevitable congestion collapse in the late 1980s, has been named the 2001 recipient of the ACM SIGCOMM Award. Jacobson is chief scientist at networking startup Packet Design, LLC.

The award is given annually by the *Association for Computing Machinery's Special Interest Group in Data Communications* (ACM SIGCOMM) to a recipient with a long and distinguished history of contributing to the field of data communications. Jacobson began his career in data communications developing control systems for the Department of Energy in the 1970s. He is best known for redesigning the TCP/IP protocol's flow-control algorithms to better handle congestion, preventing the Internet's collapse from traffic congestion in 1988–89. He is also widely recognized for his work on network synchronization effects, scalable multimedia protocols and applications, IP operations tools (for example *traceroute* and *pathchar*) and high-performance TCP implementations.

Prior to joining Packet Design as a member of the founding team, Jacobson was chief scientist at Cisco Systems, and before that had been group leader for Lawrence Berkeley Laboratory's Network Research Group.

The SIGCOMM Award has been presented every year since 1989. Prior recipients include Paul Baran, Vinton G. Cerf, David Farber and Leonard Kleinrock. ACM SIGCOMM is the world's largest professional society devoted to data communications. For more information, see: **http://www.acm.org/sigcomm/**

Useful Links

The following is a list of Web addresses that we hope you will find relevant to the material typically published in *The Internet Protocol Journal*. In the near future we will make these and other links available on our Web site: **http://www.cisco.com/ipj**

If you have suggestions for other pointers to include, please drop us a line at **ipj@cisco.com**

- The *Internet Engineering Task Force* (IETF). The primary standards-setting body for Internet technologies. **http://www.ietf.org**
- *Internet-Drafts* are working documents of the IETF, its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are not an archival document series. These documents should not be cited or quoted in any formal document. Unrevised documents placed in the Internet-Drafts directories have a maximum life of six months. After that time, they must be updated, or they will be deleted. Some Internet-Drafts become RFCs (see below). **http://www.ietf.org/ID.html**
- The *Request For Comments* (RFC) document series. The RFCs form a series of notes, started in 1969, about the Internet (originally the ARPANET). The notes discuss many aspects of computer communication, focusing on networking protocols, procedures, programs, and concepts but also including meeting notes, opinion, and sometimes humor. The specification documents of the Internet protocol suite, as defined by IETF and its steering group the IESG, are published as RFCs. Thus, the RFC publication process plays in important role in the Internet standards process. **http://www.rfc-editor.org/**
- The *Internet Society* (ISOC) is a non-profit, non-governmental, international, professional membership organization. **http://www.isoc.org**
- The *Internet Corporation for Assigned Names and Numbers* (ICANN) "... is the non-profit corporation that was formed to assume responsibility for the IP address space allocation, protocol parameter assignment, domain name system management, and root server system management functions previously performed under U.S. Government contract by IANA and other entities." **http://www.icann.org**

- The *North American Network Operators' Group* (NANOG) "...provides a forum for the exchange of technical information, and promotes discussion of implementation issues that require community cooperation. Coordination among network service providers helps ensure the stability of overall service to network users."
  http://www.nanog.org
- The *Regional Internet Registries* (RIRs) provide IP address block assignments for Internet Service Providers and others. Currently, there are three active RIRs:
  - The *Asia Pacific Network Information Centre* (APNIC):
    http://www.apnic.net
  - *RIPE Network Coordination Centre*—the RIR responsible for Europe and Northern Africa: **http://www.ripe.net**
  - *American Registry for Internet Numbers* (ARIN)—the RIR responsible for the Americas and Sub-Saharan Africa:
    http://www.arin.net

  Two more RIRs are in the process of formation: *AfriNIC* for Africa and *LACNIC* for Central- and Latin America.
- The *World Wide Web Consortium* (W3C) " ... develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential as a forum for information, commerce, communication, and collective understanding."
  http://www.w3.org/
- The *International Telecommunication Union* (ITU) "... is an international organization within which governments and the private sector coordinate global telecom networks and services."
  http://www.itu.int
- The *International Organization for Standardization* (ISO) " ... is a worldwide federation of national standards bodies from some 140 countries, one from each country. The mission of ISO is to promote the development of standardization and related activities in the world with a view to facilitating the international exchange of goods and services, and to developing cooperation in the spheres of intellectual, scientific, technological and economic activity. ISO's work results in international agreements which are published as International Standards." **http://iso.org**

This is by no means intended to be a complete list of organizations that are related to Internet development in one way or another, but this list should give you a good starting point.

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

December 2001 Volume 4, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

In a previous article entitled "Analyzing the Internet BGP Routing Table," Geoff Huston examined many issues relating to the operation of today's Internet. In this issue he goes a step further and suggests ways in which the fundamental routing architecture could be changed to solve problems related to routing-table growth. The article is called "Scaling Inter-Domain Routing—A View Forward."

The IP address space is administered by three entities, namely APNIC, ARIN and RIPE NCC. Collectively referred to as the *Regional Internet Registries* (RIRs), these organizations are responsible for address allocation to their member organizations (typically national registries or large Internet Service Providers). Since the IPv4 address space is a limited resource, this allocation has to be done with care, while accounting for the needs of the adress space consumers. We asked the RIRs for an overview of the work they perform. What we received was a joint effort that not only describes the RIR structure, but also gives some historical background on the evolution of IP addressing and routing.

We were pleased to receive a couple of Letters to the Editor recently, both in response to articles in our previous issue. This kind of feedback is most welcome and we encourage you to send your comments and suggestions to **ipj@cisco.com**

We'd like to remind you that all back issues of *The Internet Protocol Journal* can be downloaded from **www.cisco.com/ipj**. Click on "IPJ Issues" and you will be taken to the appropriate section.

By the time you read this, our online subscription system should be operational. You will find it at our Web site: **www.cisco.com/ipj**. Please let us know if you encounter any difficulties by sending e-mail to **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Scaling Inter-Domain Routing—A View Forward

by Geoff Huston,Telstra

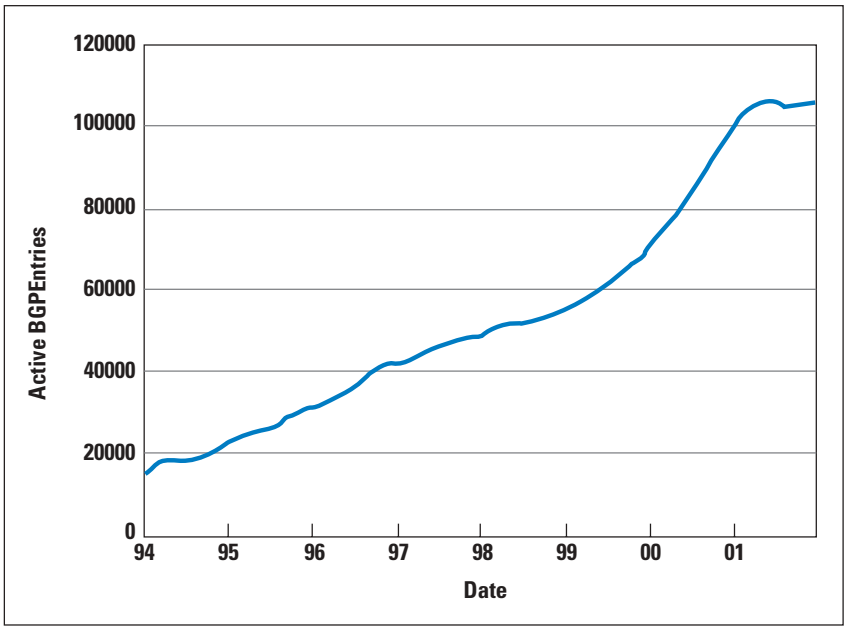
In the previous IPJ article, "Analyzing the Internet BGP Routing Table," (Vol. 4, No. 1, March 2001) we looked at the characteristics of the growth of the routing table in recent years. The motivation for this work is to observe aspects of the Internet routing table in order to understand the evolving structure of the Internet and thereby attempt to predict some future requirements for routing technology for the Internet.

The conclusions drawn in the previous article included the observation that multihomed small networks appeared to be a major contributor to growth of the Internet routing system. It also observed that there was a trend toward a denser mesh of inter-Autonomous System connectivity within the Internet. At the same time there has been an increase of various forms of policy-based constraints imposed upon this connectivity mesh, probably associated with a desire to undertake various forms of inter-domain traffic engineering through manipulation of the flow of routing information.

Taken together, these observations indicate that numerous strong growth pressures are being exerted simultaneously on the inter-domain routing space. Not only is the network itself growing in size, but also the internal interconnectivity of the network is becoming more densely meshed. The routing systems that are used to maintain a description of the network connectivity are being confronted with having to manipulate smaller route objects that describe finer levels of network detail. This is coupled with lengthening lists of qualifying attributes that are associated with each route object. The question naturally arises as to whether the *Border Gateway Protocol* (BGP) and the platforms used to support BGP in the Internet today can continue to scale at a pace that matches the growth in demands that are being placed upon it.

The encouraging news is that there appears to be no immediate cause for concern regarding the capability of BGP to continue to support the load of routing the Internet. The processor and memory capacity in current router platforms is easily capable of supporting the load associated with various forms of operational deployment models, and the protocol itself is not in imminent danger of causing network failure through any internal limitation within the protocol itself. Also, numerous network operators have exercised a higher level of care as to how advertisements are passed into the Internet domain space and, as a result, the growth rates for the routing table over 2001 shows a significant slowdown over the rates of the previous two years (Figure 1).

Figure 1: BGP Table Size 1994–2001



However, the observed trends in inter-domain routing of an increasingly detailed and highly qualified view of a more densely interconnected and still-growing network provide adequate grounds to examine the longer-term routing requirements. It is useful, therefore, to pose the question as to whether we can continue to make incremental changes to the BGP protocol and routing platforms, or whether the pace of growth will, at some point in time, mandate the adoption of a routing architecture that is better attuned to the evolving requirements of the Internet.

This article does not describe the operation of an existing protocol, nor does it describe any current operational practice. Instead it examines those aspects of inter-domain routing that are essential to today's Internet, and the approaches that may be of value when considering the evolution of the Internet inter-domain routing architecture. With this approach, the article illustrates one of the initial phases in any technology development effort—that of an examination of various requirements that could or should be addressed by the technology.

Attributes of an Inter-Domain Routing Architecture

Let's start by looking at those aspects of the inter-domain routing environment that could be considered a base set of attributes for any inter-domain routing protocol.

Accuracy

For a routing system to be of any value, it should accurately reflect the forwarding state of the network. Every routing point is required to have a consistent view of the routing system in order to avoid forwarding loops and black holes (points where there is no relevant forwarding information and the packet must be discarded). Local changes in underlying physical network, or changes in the policy configuration of the network at any point, should cause the routing system to compute a new distributed routing state that accurately reflects the changes.

This requirement for accuracy and consistency is not, strictly speaking, a requirement that every node in a routing system has global knowledge, nor a requirement that all nodes have precisely the same scope of information. In other words, a routing system that detects and avoids routing loops and inconsistent black holes does not necessarily need to use routing systems that rely on uniform distribution of global knowledge frameworks.

Scalability

Scalability can be expressed in many ways, including the number of routing entries, or prefixes, carried within the protocol, the number of discrete routing entities within the inter-domain routing space, the number of discrete connectivity policies associated with these routing entries, and the number of protocols supported by the protocol. Scalability also needs to encompass the dynamic nature of the network, including the number of routing updates per unit of time, time to converge to a coherent view of the connectivity of the network following changes, and the time taken for updates to routing information to be incorporated into the network forwarding state. In expressing this ongoing requirement for scalability in the routing architecture, there is an assumption that we will continue to see an Internet that is composed of a large number of providers, and that these providers will continue to increase the density of their interconnection.

The growth trends in the inter-domain routing space do not appear to have well-defined upper limits, so placing bounds on various aspects of the routing environment is impractical. The only practical way to describe this attribute is that it is essential to use a routing architecture that is scalable to a level well beyond the metrics of today's Internet.

In the absence of specific upper bounds to quantify this family of requirements, the best we conclude here is that at present we are working in an inter-domain environment that manipulates some 10^5 distinct routing entries, and at any single point of interconnection there may be of the order of 10^6 routing protocol elements being passed between routing domains. Experience in scaling transmission systems for the Internet indicates that an improvement of a single order of magnitude in the capacity of a technology has a relatively short useful lifetime. It would, therefore, be reasonable to consider that a useful attribute is to be able to operate in an environment that is between two to three orders of magnitude larger than today's system.

Policy Expressiveness

Routing protocols perform two basic tasks: first, determining if there is at least one viable path between one point in the network and another, and secondly, where there is more than one such path, determining the "best" such path to use. In the case of interior routing protocols, "best" is determined by the use of administratively assigned per-link metrics, and a "best" path is one that minimizes the sum of these link metrics.

In the case of the inter-domain routing protocols, no such uniformly interpreted metric exists, and "best" is expressed as a preference using network paths that yield an optimal price and performance outcome for each domain.

The underlying issue here is that the inter-domain routing system must straddle a collection of heterogeneous networks, and each network has a unique set of objectives and constraints that reflect the ingress, egress, and transit routing policies of a network. Ingress routing policies reflect how a network learns information, and which learned routes have precedence when selecting a routing entry from a set of equivalent routes. In a unicast environment, exercising control over how routes are learned by a domain has a direct influence over which paths are taken by traffic leaving the domain. Egress policies reflect how a domain announces routes to its adjacent neighbors. A domain may, for example, wish to announce a preferential route to a particular neighbor, or indicate a preference that the route not be forwarded beyond the adjacent neighbor. In a unicast environment, egress routing policies have a bearing on which paths are used for traffic to reach the domain. Transit routing policies control how the routes learned from an adjacent domain are advertised to other adjacent domains. If a domain is a transit provider for another domain, then a typical scenario for the transit provider would be to announce all learned routes to all other connected domains. For a multi-homed transit customer, routes learned from one transit provider would normally not be announced to any other transit provider.

This requirement for policy expressiveness implies that the inter-domain routing protocol should be able to attach various attributes to protocol objects, allowing a domain to communicate its preferences relating to handling of the route object to remote domains.

Robust Predictable Operational Characteristics

A routing system should operate in such a way that it achieves predictable outcomes. The inference here is that under identical initial conditions a routing system should always converge to the same routing state, and that with knowledge of the rules of operation of the protocol and the characteristics of the initial environment, an observer can predict what this state will be. Predictability also implies stability of the routing environment, such that a routing state should remain constant for as long as the environment itself remains constant.

The routing protocol should operate in a way that tends to damp propagation of dynamic changes to the routing system rather than amplify such changes. This implies that minor variations in the state of the network should not cause large-scale instability across the entire network while a new stable routing state is reached. Instead, routing changes should be propagated only as far as necessary to reach a new stable state, so that the global requirement for stability implies some degree of locality in the behavior of the system.

The routing system should have robust convergence properties. A change in the physical configuration or policy environment in any part of the network causes a distributed computation of the routing state. Convergence implies that this distributed computation reaches a conclusion at some point. The requirement for a robust convergence property implies that the distributed computation should always halt, that the halting point be reached quickly, and the system should avoid generating transitory incorrect intermediate routing states. The interpretation of "quickly" in this context is variable. Currently, this value for BGP convergence time is of the order of tens to hundreds of seconds. In order to support increasingly time-critical applications, there appears to be an emerging requirement to reduce the median convergence time for the inter-domain routing protocol to a small number of seconds.

Efficiency

The routing system should be efficient, in that the amount of network resources, in terms of bandwidth and processing capacity of the network switching elements, should not be disproportionately large. This is an area of trade-off in that the greater the amount of information passed within the routing system and the greater the frequency of such information exchanges, the greater the level of expectation that the routing system can continuously maintain an accurate view of the connectivity of the network, but at a cost of higher overhead. It is necessary to pass enough information across the system to allow each routing element to have a sufficiently accurate view of the network, yet ensure that the total routing overhead is low.

Evolving Requirements of Inter-Domain Routing

Layered on top of the base set of routing requirements listed above are a second set of requirements that can be seen as reflecting current directions in the deployed Internet, and are not necessarily well integrated into the existing routing architecture.

Multi-Homing of Edge Networks

Multi-homing refers to the practice of using more than one upstream transit provider. The common motivation for such a configuration is that if service from one transit provider fails, the customer can use the other provider as a means of service restoration. It may also allow some form of traffic balancing across multiple services. With careful use of route policies, the customer can direct traffic to each provider to minimize delay and loss, achieving some improved application performance.

The issue presented by multi-homing is that the multi-homed network is now not wholly contained within a service hierarchy of any particular provider. This implies that routing information describing reachability to the multi-homed customer cannot readily be aggregated into any single provider's routing advertisements, and the usual outcome is that the multi-homed customer must independently announce its reachability to each transit provider, who in turn must propagate this information across the routing system.

The evolving requirement here is one that must be able to integrate the demands of an increasing use of multi-homing into the overall network design. Two basic forms of approach can be used here—one is to use a single address block across the customer network and announce this block to all transit providers as an unaggregatable routing advertisement into the inter-domain routing system, and the other is to use multiple address blocks drawn from each provider's address block, and use either host-based software or some form of dynamic address translation within the network in order to use a source address drawn from a particular provider's block for each network transaction (Figure 2). The second approach is not widely used, and for the immediate future the requirement for multi-homing is normally addressed by using unique address blocks for the multi-homed network that are not part of any provider's aggregated address blocks. The consequence of this is that widespread use of multi-homing as a means of service resiliency will continue to have an impact on the inter-domain routing system.

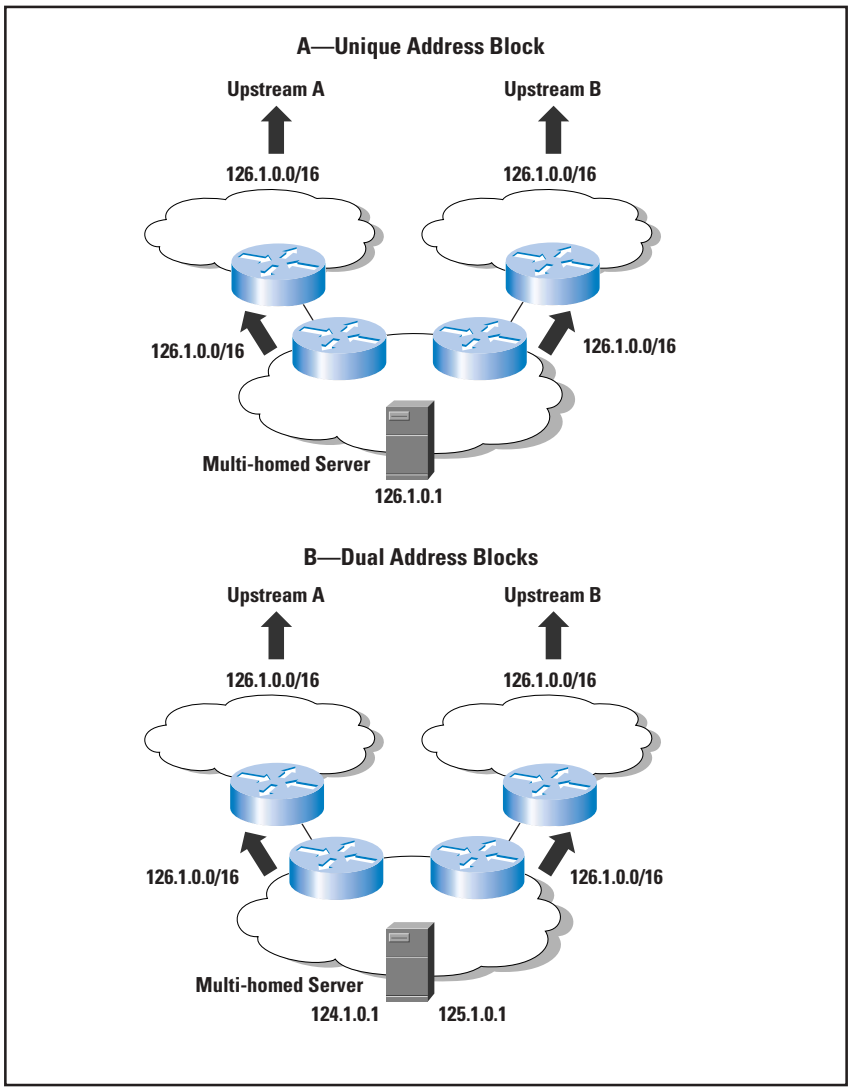


Figure 2: Routing Approaches to Multi-Homing

Inter-Domain Traffic Engineering

In an increasingly densely interconnected network, selecting and using just one path between two points is not an optimal outcome of a routing architecture. Of more importance is the ability to identify a larger set of viable paths between these points and distribute the associated traffic flows in such a way that each individual transaction uses a single path, but the total set of flows is distributed across the set of paths.

To achieve this outcome, more information must be placed into the routing system, allowing a route originator to describe the policy-based preferences of which sets of paths should be preferred for traffic destined to the route originator, allowing a transit service operator to add information regarding current preferences associated with using particular transit paths, and allowing the traffic originator the ability to use local traffic egress policies to reach the destination. These traffic engineering-related preferences are not necessarily represented by static values of routing attributes. One of the requirements of traffic engineering is to allow the network to dynamically respond to shifting traffic load patterns, and this implies that there is a component of dynamic information update that is associated with such traffic engineering-related aspects of the routing system.

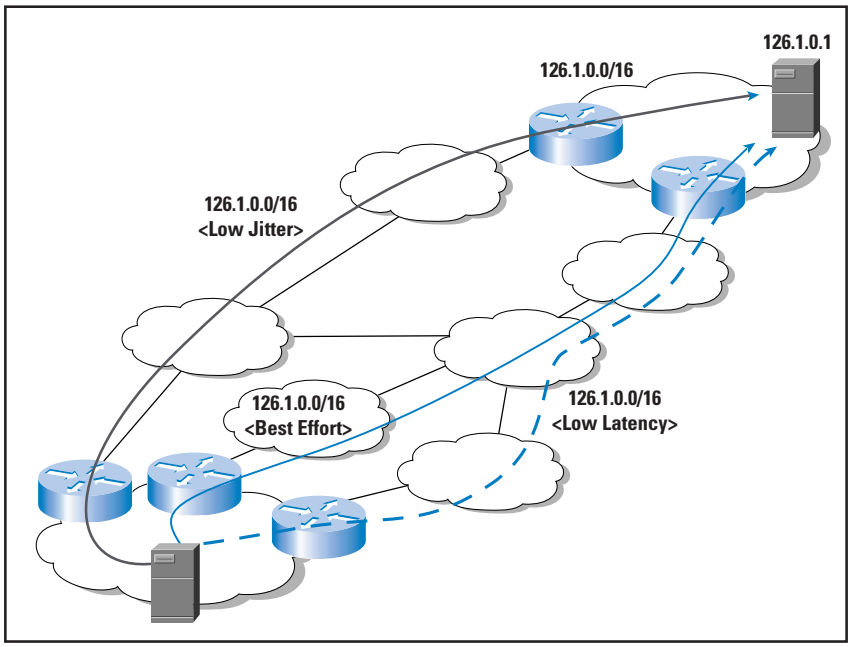
At an abstract level, this greater volume of routing information is needed in order to address the dual role of the routing system as both an inter-domain connectivity maintenance protocol and as a traffic-engineering tool.

Inter-Domain Quality of Service

Quality of Service (QoS) is a term that encompasses a wide variety of mechanisms. In the case of routing, the term is used to describe the process of modifying the normal routing response of associating a single forwarding action with a destination address prefix in such a way that there may be numerous forwarding decisions for a particular address prefix. Each forwarding decision is associated with a particular service response, so that a "best-effort" path to a particular destination address may differ from a "low-latency" path, which in turn may differ from a "high-bandwidth" path, and so on.

As with inter-domain traffic engineering, this requirement is one which would be expected to place greater volumes of information into the routing domain. At an abstract level this requirement can be seen as the association of a service quality attribute with an address prefix, and passing the paired entity into the routing domain as a single routing object. The inference is that multiple quality attributes associated with a path to a particular prefix would require the routing system to independently manipulate multiple route objects, because it would be reasonable to anticipate that the routing system would select different paths to reach the same address prefix if different QoS service attributes were used as a path qualifier (Figure 3).

Figure 3: Inter-Domain Routing with QoS



Approaches to Inter-Domain Routing

Let's now take this set of requirements and attempt to match them to various approaches to routing protocols.

Routing is a distributed computation wherein each element of the computation set must reach an outcome that is consistent with all other computations undertaken by other members of the set. There are two major approaches to this form of distributed computation, namely *serial* or *parallel* computation. Serial computation involves each element of the set undertaking a local computation and then passing the outcomes of this computation to its adjacent elements. This approach is used in various forms of distance-vector routing protocols where each routing node computes a local set of selected paths, and then propagates the set of reachable prefixes and the associated path metric to its neighbors. Parallel computation involves rapid flooding of the current state of connectivity within the set to all elements, and all set elements simultaneously compute forwarding decisions using the same base connectivity data. This approach is used in various forms of link-state routing protocols, where the protocol uses a flooding technique to rapidly propagate updated link-status information and then relies on each routing node to perform a local path selection computation for each reachable address prefix. Is one of these approaches substantially better suited than the other to the inter-domain routing environment?

Open or Closed Routing Policies

One of the key issues behind consideration of this topic is that of the role of *local policy*. Using a distance-vector protocol, a routing domain gathers selected path information from its neighbors, applies local policy to this information, and then distributes this updated information in the form of selected paths to its neighbor domains.

In this model the nature of the local policy applied to the routing information is not necessarily visible to the domain neighbors, and the process of converting received route advertisements into advertised route advertisements uses a local policy process whose policy rules are not visible externally. This scenario can be described as *policy opaque*. The side effect of such an environment is that a third party cannot remotely compute which routes a network may accept and which may be readvertised to each neighbor.

In link-state protocols, a routing domain effectively broadcasts its local domain adjacencies, and the policies it has with respect to these adjacencies, to all nodes within the link-state domain. Every node can perform an identical computation upon this set of adjacencies and associated policies in order to compute the local inter-domain forwarding table. The essential attribute of this environment is that the routing node has to announce its routing policies in order to allow a remote node to compute which routes will be accepted from which neighbor, and which routes will be advertised to each neighbor and what, if any, attributes are placed on the advertisement. Within an interior routing domain the local policies are in effect metrics of each link, and these polices can be announced within the routing domain without any consequent impact.

In the exterior routing domain it is not the case that interconnection policies between networks are always fully transparent. Various permutations of supplier/customer relationships and peering relationships have associated policy qualifications that are not publicly announced for business competitive reasons. The current diversity of interconnection arrangements appears to be predicated on policy opaqueness, and to mandate a change to a model of open interconnection policies may be contrary to operational business imperatives. An inter-domain routing tool should be able to support models of interconnection where the policy associated with the interconnection is not visible to any third party. If the architectural choice is a constrained one between distance vector and link state, then this consideration would appear to favor the continued use of a distance-vector approach to inter-domain routing. This choice, in turn, has implications on the convergence properties and stability of the inter-domain routing environment. If there is a broader spectrum of choice, the considerations of policy opaqueness would still apply.

Separation of Functions

The inter-domain routing function undertakes many roles simultaneously. First, it maintains the current view of inter-domain connectivity. Any changes in the adjacency of a domain are reflected in a distributed update computation that determines if the adjacency change implies a change in path selection and in address reachability. Secondly, it maintains the set of currently reachable address prefixes. And finally, the protocol binds the first two functions together by associating each prefix with a path through the inter-domain space.

This association uses a policy framework to allow each domain to select a path that optimizes local policy constraints within the bounds of existing constraints applied by other domains. This policy may be related to traffic-engineering objectives, QoS requirements, local cost optimization, or related operational or business objectives.

An alternative approach to inter-domain routing is to separate the functions of connectivity maintenance, address reachability, and policy negotiation. As an example of this approach, a connectivity protocol can be used to identify all viable paths between a source and a destination domain. A policy negotiation protocol can be used to ensure that there are a consistent sequence of per-domain forwarding decisions that will pass traffic from the source domain to the destination domain. An address reachability protocol can be used to associate a collection of address prefixes with each destination domains. This framework is illustrated in Figure 4.

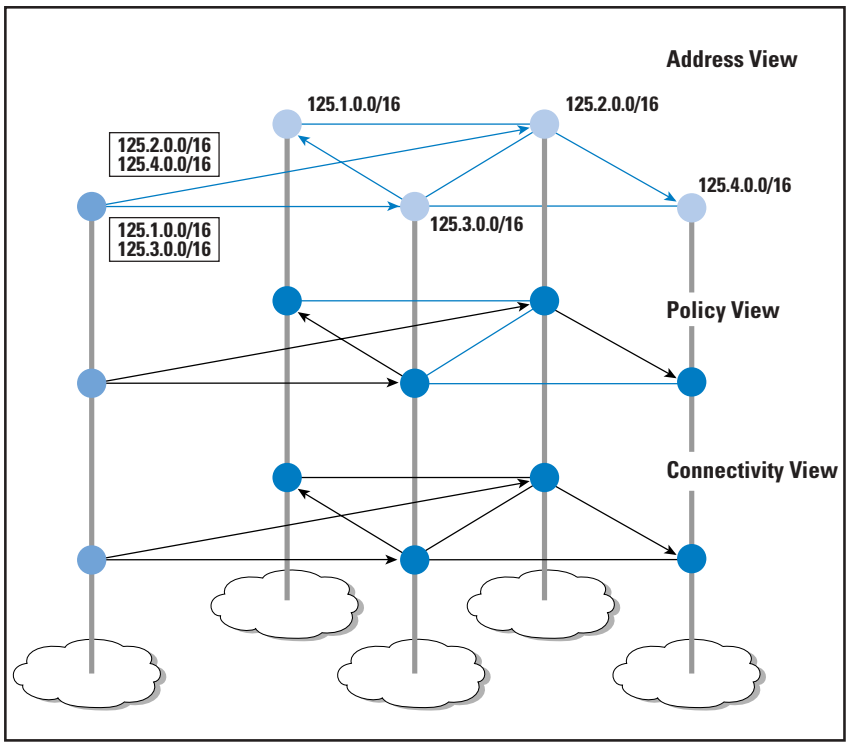


Figure 4: A Multi-Tiered Approach to Inter-Domain Routing

Address Prefixes and Autonomous System Numbers

One observation about the current inter-domain routing system is that it uses a view of the network based on computing the optimal path to each address prefix. This view is translated into an inter-domain routing protocol that uses the address prefix as the basic protocol element and attaches various attributes to each address prefix as they are passed through the network

As of late 2001, the routing system had some 100,000 distinct address prefixes and 11,500 origin domains. This implies that each origin domain is responsible for an average of 8 to 9 address prefixes. If each domain advertised its prefixes with a consistent policy, then each address prefix would be advertised with identical attributes. If the routing protocol were to be inverted such that the routing domain identifier, or *Autonomous System* number, were the basic routing object and the set of prefixes and associated common set of route attributes were attributes of the Autonomous System object, then the number of routing objects would be reduced by the same factor of between 8 and 9.

The motivation in this form of approach is that seeking clear hierarchical structure in the address space as deployed is no longer feasible, and that no further scaling advantage can be obtained by various forms of address aggregation within the routing system. This approach replaces this address-based hierarchy with a two-level hierarchy of routing domains. Within a routing domain, routing is undertaken using the address prefix. Between routing domains, routing is undertaken using domain identifiers and associated sets of domain attributes.

Although this approach appears to offer some advantage in creating a routing domain, one-tenth of the size of the address prefix-based routing domain, it is interesting to note that since late 1996 the average number of address prefixes per Autonomous System has fallen from 25 to the current value of 9. In other words, the number of distinct routing domains is growing at a faster rate than the number of routed address prefixes. While the adoption of a domain-based routing protocol offers some short-term advantages in scaling, the longer-term prospects are not so attractive, given these relative growth rates.

Routing Hierarchies of Information

The scaling properties of an inter-domain routing protocol are related on the ability of the protocol to remove certain specific items of information from the routing domain at the point where it ceases to have any differentiating impact. For example, it is important for a routing protocol to carry information that a particular domain has multiple adjacencies and that there are a number of policies associated with each adjacency, and propagate this information to all local domains. At a suitably distant point in the network, the forwarding decision remains the same regardless of the set of local adjacencies, and propagation of the detail of the local environment to points where the information ceases to have any distinguishing outcome is unproductive.

From this perspective, scaling the routing system is not a case of determining what information can be added into the routing domain, but instead it's a case of determining how much information can be removed from the routing domain, and how quickly.

One way of removing information is through the use of *hierarchies*. Within a hierarchical structure, a set of objects with similar properties are aggregated into a single object with a set of common properties. One way to perform such aggregation is by increasing the amount of information contained in each aggregate route object. For example, if single route objects are to be used that encompass a set of address prefixes and a collection of Autonomous Systems, then it would be necessary to define additional attributes within the route object to further qualify the policies associated with the object in terms of specific prefixes, specific Autonomous Systems, and specific policy semantics that may be considered as policy exceptions to the overall aggregate. This approach would allow aggregation of routing information to occur at any point in the network, allowing the aggregator to create a compound object with a common set of attributes, and a set of additional attributes that apply to a particular subset of the aggregate.

Another approach to using hierarchies to reduce the number of route objects is to reduce the scope of advertisement of each routing object, allowing the object to be removed and proxy aggregated into some larger object when the logical scope of the object is reached. This approach would entail the addition of route attributes that could be used to define the circumstances where a specific route object would be subsumed by an aggregate route object without impacting the policy objectives associated with the original set of advertisements. This approach places control of aggregation with the route object originator, allowing the originator to specify the extent to which a specific route object should be propagated before being subsumed into an aggregate object.

It is not entirely clear that the approach of exploiting hierarchies in an address space is the most appropriate response to scaling pressures. Viewed from a more general perspective, scaling of the routing system requires the systematic removal of information from the routing domain. The way this is achieved is by attempting to align the structure of deployment with some structural property of the syntax of the protocol elements that are being used as routing objects. Information can then be eliminated through systematic aggregation of the routing objects at locations within the routing space that correspond to those points in the topology of the network where topology aggregation is occurring. The maintenance of this tight coupling of the structure of the deployed network to the structure of the identifier space is the highest cost of this approach. Alterations to the topology of the network through the relocation or reconfiguration of networks requires renumbering of the protocol element if hierarchical aggregation is to be maintained. If the address space is the basis of routing, as at present, then this becomes a large-scale exercise of renumbering networks that in turn implies an often prohibitively disruptive and expensive exercise of renumbering collections of host systems and associated services.

One view of this is that the connectivity properties of the Internet are already sufficiently meshed that there is no readily identifiable hierarchical structure, and that this trend is becoming more pronounced, not less. In that case, the most appropriate course of action may be to reexamine the routing domain and select some other attribute as the basis of the routing computation that does not have the same population, complexity, and growth characteristics as address prefixes, and base the routing computation on this attribute. One such alternative approach is to consider Autonomous System numbers as routing "atoms" where the routing system converges to select an Autonomous System path to a destination Autonomous System, and then uses this information to add the associated set of prefixes originated by this Autonomous System, and next-hop forwarding decision to reach this Autonomous System into the local forwarding table.

Extend or Replace BGP

A final consideration is to consider whether these requirements can best be met by an approach of a set of upward-compatible extensions to BGP, or by a replacement to BGP.

The rationale for extending BGP would be to increase the number of commonly supported transitive route attributes, and, potentially, allow a richer syntax for attribute definition which in turn would allow the protocol to use a richer set of semantic definitions in order to express more complex routing policies.

This direction may sound like a step backward, in that it proposes an increase in the complexity of the route objects carried by the protocol and potentially increases the amount of local processing capability required to generate and receive routing updates. However, this can be offset by potential benefits that are realizable through the greater expressive capability for the policy attributes associated with route objects. It can allow a route originator an ability to specify the scope of propagation of the route object, rather than assuming that propagation will be global. The attributes can also describe intended service outcomes in terms of policy and traffic engineering. It may also be necessary to allow BGP sessions to negotiate additional functionality intended to improve the convergence behavior of the protocol. Whether such changes can produce a scalable and useful outcome in terms of inter-domain routing remains, at this stage, an open question.

An alternative approach is that of a replacement protocol. Use of a parallel-processing approach to the distributed computation of routing, such as that used in the link-state protocols, can offer the benefits of faster convergence times and avoidance of unstable transient routing states. On the other hand, link-state protocols present issues relating to policy opaqueness, as described above. Another major issue with such an approach is the need to address the efficiency of inter-domain link-state flooding.

The inter-domain space would need some further levels of imposed structure similar to intra-domain areas in order to ensure that individual link updates are rapidly propagated across the relevant subset of the network. The use of such an area structure may well imply the need for an additional set of operator relationships, such as mutual transit. Such inter-domain relationships may prove challenging to adapt to existing operator practices.

Another approach could be based on the adoption of a multi-layer approach of separate protocols for separate functions, as described above. A base inter-domain connectivity protocol could potentially be based on a variant of a link-state protocol, using the rapid convergence properties of such protocols to maintain a coherent view of the current state of connectivity within the network. The overlay of a policy protocol would be intended as a signaling mechanism to allow each domain to make local forwarding decisions that are consistent with those adopted by adjacent domains, thereby maintaining a collection of coherent inter-domain paths from source to destination. Traffic engineering can also be envisaged as an overlay mechanism, allowing a source to make a forwarding decision that selects a path to the destination where the characteristics of the path optimize the desired service outcomes.

Directions for Further Activity

Although short-term actions based on providing various incentives for network operators to remove redundant or inefficiently grouped entries from the BGP routing table may exist, such actions are short-term palliative measures, and will not provide long-term answers to the need for a scalable inter-domain routing protocol. One approach to the longer-term requirements may be to preserve many of the attributes of the current BGP protocol, while refining other aspects of the protocol to improve its scaling and convergence properties. A minimal set of alterations could retain the Autonomous System concept to allow for administrative boundaries of information summarization, as well as retaining the approach of associating each prefix advertisement with an originating Autonomous System. The concept of policy opaqueness would also be retained in such an approach, implying that each Autonomous System accepts a set of route advertisements, applies local policy constraints, and readvertises those advertisements permitted by the local policy constraints. It could be feasible to consider alterations to the distance-vector path-selection algorithm, particularly as it relates to intermediate states during processing of a route withdrawal. It is also feasible to consider the use of compound route attributes, allowing a route object to include an aggregate route, and numerous specifics of the aggregate route, and attach attributes that may apply to the aggregate or a specific address prefix. Such route attributes could be used to support multi-homing and inter-domain traffic-engineering mechanisms. The overall intent of this approach is to address the major requirements in the inter-domain routing space without using an increasing set of globally propagated specific route objects.

Another approach is to consider the feasibility of decoupling the requirements of inter-domain connectivity management with the applications of policy constraints and the issues of sender- and receiver-managed traffic-engineering requirements. Such an approach may use a link-state protocol as a means of maintaining a consistent view of the topology of inter-domain network, and then use some form of overlay protocol to negotiate policy requirements of each Autonomous System, and use a further overlay to support inter-domain traffic-engineering requirements. The underlying assumption of such an approach is that if the functional role of inter-domain routing is divided into distinct components, each component will have superior scaling and convergence properties which in turn will result in superior properties for the entire routing system. Obviously, this assumption requires some testing.

Research topics with potential longer-term application include the approach of drawing a distinction between the identity of a network, its location relative to other networks, and maintenance of a feasible path set between a source and destination network that satisfies various policy and traffic-engineering constraints. Again the intent of such an approach would be to divide the current routing function into numerous distinct scalable components rather than using a single monolithic routing protocol.

Further Reading

[0] Huston, G., "Analyzing the Internet BGP Routing Table," *The Internet Protocol Journal,* Vol. 4, No. 1, March 2001.
www.cisco.com/warp/public/759/ipj_4-1/ipj_4-1_bgp.html

[1] Huitema, C., *Routing in the Internet, 2nd Edition,* ISBN 0130226475, Prentice Hall, January 2000. *A good introduction to the general topic of IP routing.*

[2] Rekhter, Y., and Li T., "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, March 1995. *The base specification of BGP 4. This document is currently being updated by the IETF. The state of this work in progress as of November 2001 is documented as an Internet Draft,*
draft-ietf-idr-bgp4-15.txt

[3] Elwyn Davies et al., "Future Domain Routing Requirements," work in progress, July 2001. *This work is currently documented as an Internet Draft,* **draft-davies-fdr-reqs-01.txt**. *It contains a review of an earlier effort in enumerating routing requirements ("Goals and Functional Requirements for Inter-Autonomous System Routing," RFC 1126, October 1989), as well as a commentary on a proposed set of current routing requirements.*

GEOFF HUSTON holds a B.Sc. and M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is currently the Chief Scientist in the Internet area for Telstra, a member of the Internet Architecture Board, and is the Secretary of the APNIC Executive Committee. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@telstra.net**

Development of the Regional Internet Registry System

by Daniel Karrenberg, RIPE-NCC; Gerard Ross, APNIC; Paul Wilson, APNIC; Leslie Nobile, ARIN

The current system of managing Internet address space involves *Regional Internet Registries* (RIRs), which together share a global responsibility delegated to them by the *Internet Assigned Numbers Authority* (IANA). This regime is now well established, but it has evolved over ten years from a much simpler, centralized system. Internet number spaces were originally managed by a single individual "authority," namely the late Jon Postel, co-inventor of some of the most important technical features of today's Internet.

It is important to understand that the evolution of the RIR system was not simply the result of Internet growth and the natural need to refine and decentralize a growing administrative task. On the contrary, it arose from, and closely tracked, the technical evolution of the Internet Protocol, in particular the development of today's IP addressing and routing architecture.

In a relatively short time, the Regional Internet Registry system has evolved into a stable, robust environment for Internet address management. It is maintained today through self-regulatory practices that are well established elsewhere in the Internet and other industries, and it maintains its legitimacy and relevance by firmly adhering to open, transparent, participatory decision-making processes.

Before the RIRs:

IP Address Architecture

An important feature of the Internet Protocol (IP) is the ability to transparently use a wide variety of underlying network architectures to transport IP packets. This is achieved by encapsulating IP packets in whatever packet or frame structure the underlying network uses. Routers connecting different networks forward IP traffic by decapsulating incoming IP packets and then re-encapsulating them as appropriate for the next network to carry them.

To achieve this task with full transparency, the IP needed an addressing structure, which developed as a two-level hierarchy in both addressing and routing. One part of the address, the *network* part, identifies the particular network a host is connected to, while the other part, the *local* part, identifies the particular end system on that network.

Internet routing, then, has to deal only with the network part of the address, routing the packet to a router directly connected to the destination network. The local part is not used at all in Internet routing itself; rather it is used to determine the intended address within the addressing structure of the destination network.

The method by which the local part of an IP address is translated to a local network address depends on the architecture of the destination network—static tables, simple conversions, or special-purpose protocols are used as appropriate.

The original Internet addresses comprised 32 bits, the first 8 bits providing the network part and the remaining 24 bits the local part. These addresses were used for many years. However, in June 1978, in Internet Engineering Note (IEN) 46 "A proposal for addressing and routing in the internet," Clark and Cohen observed:

> "The current internet header has space to name 256 networks. The assumption, at least for the time being, is that any network entering the internet will be assigned one of these numbers. While it is not likely that a great number of large nets, such as the ARPANET, will join the internet, the trend toward local area networking suggests that a very large number of small networks can be expected in the internet in the not too distant future. We should thus begin to prepare for the day when there are more than 256 networks participating in the internet."

Classful Addressing

As predicted, it soon became necessary to adapt the address architecture to allow more networks to be connected. By the time the Internet Protocol itself was comprehensively specified (in RFC 790, published in 1981, edited by Jon Postel), the IP address could be segmented in numerous ways to provide three classes of network address.

In Class A, the high-order bit is zero, the next 7 bits are the network, and the last 24 bits are the local address. In Class B, the high-order 2 bits are one-zero, the next 14 bits are the network, and the last 16 bits are the local address. In Class C, the high-order 3 bits are one-one-zero, the next 21 bits are the network, and the last 8 bits are the local address.

This so-called "classful" architecture served the Internet for the next 12 years, during which time it grew from a small U.S.-based research network to a global academic network showing the first signs of commercial development.

Early Registration Models

In the 1980s, the American *National Science Foundation's* (NSF's) high-speed network, NSFNET, was connected to the ARPANET, a U.S. *Defense Advanced Research Projects Agency* (ARPA, now DARPA) wide-area network, which essentially formed the infrastructure that we now know as the Internet.

From these early days of the Internet, the task of assigning addresses was a necessary administrative duty, to ensure simply that no two networks would attempt to use the same network address in the Internet.

At first, the elementary task of maintaining a list of assigned network addresses was carried out voluntarily by Jon Postel, using (according to legend) a paper notebook.

As the Internet grew, and particularly as classful addressing was established, the administrative task grew accordingly. The IANA was established, and within it the Internet Registry (IR). But as the task of the IR outgrew Postel's notebook, it was passed to SRI International in Menlo Park, California, under a NSF contract, and was called the *Defense Data Network* (DDN) *Network Information Center* (NIC).

During this time, under the classful address architecture, networks were allocated liberally and to any organization that fulfilled the simple request requirements. However, with the accelerating growth of the Internet during the late 1980s, two problems loomed: the rapid depletion of address space, due to the crude classful divisions; and the uncontrolled growth of the Internet routing table, due to unaggregated routing information.

Conservation vs. Aggregation

The problems of "three sizes fit all" highlight the basic dilemma of address space assignment: conservation versus aggregation. On the one hand, one wants to conserve the address space by assigning as little as possible; on the other hand, one wants to ease routing-table pressures by aggregating as many addresses as possible in one routing-table entry.

This can be illustrated by looking at a typical networking setup of the time. Within organizations having a single Internet connection, buildings, departments, or campuses would have their own local networks. Often the use of multiple networks was dictated by distance limitations inherent in the emerging local-area networking technologies, such as Ethernet.

These networks typically had to accommodate more than the 254 hosts addressable by a Class C address, but would rarely exceed 1000 hosts. Using pure classful addressing, one could either subdivide networks artificially to remain below the 254 host limit, or use a Class B address for each local network, possibly wasting more than 60,000 addresses in each. Whereas the latter solution is obviously wasteful in terms of address space, the former is obviously cumbersome. Less obviously, the former also puts an additional burden on the Internet routing system, because each of these networks would require a separate route propagated throughout the whole Internet.

This basic dilemma persists to this day. Assigning address space generously tends to reduce the routing-table size, but wastes address space. Assigning conservatively will waste less, but cause more stress for the routing system.

Subnetting

In order to address some of the problems of classful addressing, the technique of *subnetting* was invented. Described in RFC 791 in 1984, subnetting provided another level of addressing hierarchy by inserting a *subnet* part into the IP address between the network and local parts. Global routing remained the same using the *network* part of the address (Class A, B, or C) until traffic reached a router on the network identified by the network part of the address. This router, configured for subnetting, would interpret a statically configured number of bits from the local part of the address (the subnet part) to route the packet further among a set of similarly configured routers. When the packet reached a router connected to the destination subnet, the remaining bits of the local part would be used to determine the local address of the destination as usual. So, in the previous example, the organization could have used a Class B address with 6-bit subnetting, a setup that would allow for 62 networks of 1022 hosts each.

Subnetting nicely solved the routing-table problem, because now only one global routing-table entry was needed for the organization. It also helped address space conservation somewhat because it provided an obvious alternative to using many sparsely populated Class B networks.

Because the boundary between the subnet part and the local part of an address could not be determined from the address itself, this local knowledge needed to be configured into the routers. At first this was done by static configuration. Later, interior routing protocols carried that information. Refer to RFC 791 for numerous historically interesting case studies.

Supernetting

Within seven years, however, it was becoming clear that subnetting was no longer sufficient to keep up with Internet growth. RFC 1338 stated the problem:

"As the Internet has evolved and grown ... in recent years, it has become painfully evident that it is soon to face several serious scaling problems. These include:

1. Exhaustion of the Class-B network address space. One fundamental cause of this problem is the lack of a network class of a size that is appropriate for a midsized organization; Class C, with a maximum of 254 host addresses, is too small while Class B, which allows up to 65534 addresses, is too large to be widely allocated.
2. Growth of routing tables in Internet routers beyond the ability of current software (and people) to effectively manage.
3. Eventual exhaustion of the 32-bit IP address space.

It has become clear that the first two of these problems are likely to become critical within the next one to three years."

The solution proposed was to extend the subnetting technique beyond the local organization, into the Internet itself. In other words, RFC 1338 proposed abolishing classful addressing, and replacing it with *supernetting*. The proposal was summarized as follows:

"The proposed solution is to hierarchically allocate future IP address assignment, by delegating control of segments of the IP address space to the various network service providers."

CIDR

In 1993, the supernetting technique was published as a standards track RFC under the name *Classless Inter-Domain Routing* (CIDR), by which it is known and used today. Two main ingredients were necessary to make CIDR work: routing system changes and new address allocation and assignment procedures.

Under CIDR, routers could no longer determine the network part of an address from the address itself. This information now needed to be conveyed by Internet routing protocols. Fortunately, there was only one such protocol in widespread use at the time, and it was quickly extended by the major router vendor of the time. According to legend, the necessary extensions of the *Border Gateway Protocol* (BGP)-3 to BGP-4 were designed on a napkin, with all implementors of significant routing software present. The changes were implemented in a matter of days, but only much later described by the Internet standards track RFC 1654.

CIDR also required that forwarding decisions of routers be changed slightly. The network part of an address, now more generally called the *prefix,* can be of any length. This means that a router can have multiple valid routes covering a specific 32-bit destination address. Routers need to use the most specific of these routes—*the longest prefix*—when forwarding packets.

In additional to technical changes, the success of CIDR also relied on the development of administrative procedures to allocate and assign address space in such a way that routes could be aggregated as much as possible. Because the Internet was evolving toward the current state of arbitrarily interconnected networks of *Internet Service Providers* (ISPs), it was obvious that ISPs should play a role in address space distribution. In the new technique, ISPs would now, as much as possible, assign address space to their customers in contiguous blocks, which could be aggregated into single routes to the rest of the Internet.

Emergence of the RIRs:

Internationalization

While the engineering-driven need for topological address space assignment was becoming clear, there was also an emerging recognition that the administrative mechanisms of address space distribution needed further development. A central system just would not scale for numerous reasons, including:

- Sheer volume
- Distance from the address space consumers
- Lack of an appropriate global funding structure
- Lack of local community support

The need to change administrative procedures was formally recognized by August 1990, when the Internet Activities Board published a message it had sent to the U.S. Federal Networking Council, stating "it is timely to consider further delegation of assignment and registration authority on an international basis" (RFC 1174).

The increasing cultural diversity of the Internet also posed administrative challenges for the central IR. In October 1992, the *Internet Engineering Task Force* (IETF) published RFC 1366, which described the "growth of the Internet and its increasing globalization" and set out the basis for an evolution of the registry process, based on a regionally distributed registry model. This document stressed the need for a single registry to exist in each geographical region of the world (which would be of "continental dimensions"). Registries would be "unbiased and widely recognized by network providers and subscribers" within their region. Each registry would be charged with allocating remaining address space in a manner "compatible with potential address aggregation techniques" (or CIDR).

RIPE NCC

While in the United States the Government continued to support and fund registry functions, this was not the case in other parts of the world. In Europe, IP network operators cooperating in *Réseaux IP Européens* (RIPE) realized the need for professional coordination and registration functions. Establishment of the RIPE *Network Coordination Centre* (NCC) was proposed in the same month that RFC 1174 was published. The RIPE NCC was to "function as a 'Delegated Registry' for IP numbers in Europe, as anticipated and defined in RFC 1174" (RIPE-19).

Although consensus among IP network operators was quickly established, it took almost two years of organizing and fund-raising before the first RIR was fully operational in May 1992. The RIPE NCC was organized as a highly independent part of RARE, the organization of European research networks. It was to be funded by contributions from those networks, as well as a small number of emerging commercial networks. The RIPE NCC published its first regional address distribution policy in July 1992 (RIPE-65).

During the following months, European regional policies were refined and, for the first time, global guidelines were published as RFCs (RFC 1366, RFC 1466).

The RIPE NCC is presently organized as a membership association, performing the essential coordination and administration activities required by the RIPE community. Located in Amsterdam, Netherlands, the RIPE NCC service region incorporates 109 countries covering Europe, the Middle East, Central Asia, and African countries located north of the equator. The RIPE NCC currently consists of more than 2700 members. At the time of publication, RIPE NCC is performing the secretariat function for the *Address Supporting Organization* (ASO) of The *Internet Corporation for Assigned Names and Numbers* (ICANN). More information about RIPE NCC is available at **`http://www.ripe.net`**

APNIC

Asia Pacific Network Information Centre (APNIC), the second RIR, was established in Tokyo in 1993, as a pilot project of APCCIRN (Asia Pacific Coordination Committee for Intercontinental Research Networks, now *Asia Pacific Networking Group* [APNG]).

The project was an intended as a trial model for servicing the Internet addressing needs of national *Network Information Centres* (NICs) and other networks throughout the region.

After a successful ten-month trial period, APNIC was established as a permanent organization to serve the Asia Pacific region (which includes 62 economies from Central and South Asia to the Islands of Oceania and the Western Pacific).

Originally, APNIC relied on the support of networking organizations and national NICs. However, in 1996, APNIC implemented a tiered membership structure.

APNIC relocated to Brisbane, Australia, in mid-1998. It currently services approximately 700 member organizations, across 39 economies of the region. Within the APNIC membership, there are also five *National Internet Registries* (NIRs), in Japan, China, Taiwan, Korea, and Indonesia. The NIRs perform analogous functions to APNIC at a national level and together represent the interests of more than 500 additional organizations.

In 2000, APNIC hosted the secretariat functions of the ASO in its inaugural year. More information about APNIC is available at:
`http://www.apnic.net`

ARIN

In 1991, the contract to perform the IR function was awarded to Network Solutions, Inc. in Herndon, Virginia. This included the transition of services including IP address registration, domain name registration and support, *Autonomous System Number* (AS) registration, user registration, online information services, help-desk operations, and RFC and Internet-Draft archive and distribution services (RFC 1261).

With explosive Internet growth in the early 1990s, the U.S. Government and the NSF decided that network support for the commercial Internet should be separated from the U.S. Department of Defense. The NSF originated a project named InterNIC under a cooperative agreement with *Network Solutions, Inc*. (NSI) in 1993 to provide registration and allocation of domain names and IP address numbers for Internet users.

Over time, after lengthy consultation with the IANA, the IETF, RIPE NCC, APNIC, the NSF, and the *Federal Networking Council* (FNC), a further consensus was reached in the general Internet community to separate the management of domain names from the management of IP numbers. This consensus was based on the recognition that the stability of the Internet relies on the careful management of IP address space.

Following the examples of RIPE NCC and APNIC, it was recommended that management of IP address space then administered by the InterNIC should be under the control of, and administered by, those that use it, including ISPs, end-user organizations, corporate entities, universities, and individuals.

As a result, ARIN (*American Registry for Internet Numbers*) was established in December 1997, as an independent, nonprofit corporation, with a membership structure open to all interested entities or individuals.

ARIN is located in Chantilly, Virginia, United States. Its service region incorporates 70 countries, covering North America, South America, the Caribbean, and African countries located south of the equator. ARIN currently consists of more than 1500 members. Within the ARIN region, there are two national delegated registries, located in Mexico and Brazil.

Until now, ARIN has carried the responsibility for maintaining registration of resources allocated before the inception of the RIRs. However, a major project is now under way to transfer these legacy records to the relevant RIRs. More information about ARIN is available at:
`http://www.arin.net`

Emerging RIRs

The existing RIRs currently serve countries outside their core regions to provide global coverage; however, new RIRs are expected to emerge, necessitating changes to the existing service regions. Because the regions are defined on continental dimensions, the number of new RIRs will be low.

Currently, two groups have made significant progress in seeking to establish new RIRs. *AfriNIC* (for the Africa region) and *LACNIC* (for Latin America and the Caribbean) have each conducted public meetings, published documentation, and participated in the activities of the

existing RIRs. In recognition of the regional support they have so far obtained, each organization has been granted observer status at ICANN ASO meetings. The existing RIRs have also sought to provide as much assistance and support as possible to these emerging organizations.

More information about AfriNIC is available at;
http://www.afrinic.org/

More information about LACNIC is available at:
http://lacnic.org/

The RIR System:

Goals of the RIRs

RFC 2050, published in November 1996, represented a collaboration of the global Internet addressing community to describe a set of goals and guidelines for the RIRs. Although IANA was to retain ultimate responsibility for the entire address pool, RFC 2050 recognizes that RIRs operate under the consensus of their respective regional Internet community. This document, along with a history of RIR coordination, has helped to form the basis for a set of consistent global policies.

The three primary goals of the RIR system follow:

- *Conservation:* to ensure efficient use of a finite resource and to avoid service instabilities due to market distortions (such as stockpiling or other forms of manipulation);
- *Aggregation (routability):* to assist in maintenance of Internet routing tables at a manageable size, by supporting CIDR techniques to ensure continued operational stability of the Internet;
- *Registration:* to provide a public registry documenting address space allocations and assignments, necessary to ensure uniqueness and provide information for Internet troubleshooting at all levels.

The Open Policy Framework

It was always recognized that these goals would often be in conflict with each other and with the interests of individuals and organizations. It was also recognized that legitimate regional interests could justify varying approaches in balancing these conflicts. Therefore, within the global framework, each regional community has always developed its own specific policies and procedures.

However, whereas the specific approaches may differ across the RIRs, all operate on a basic principle of open, transparent, consensus-based decision-making, following self-regulatory practices that exist elsewhere in the Internet and other industries. Furthermore, the RIRs all maintain not-for-profit cost-recovery systems and organizational structures that seek to be inclusive of all interested stakeholders.

The activities and services of each of the RIRs are defined, performed, discussed, and evaluated in open forums, whose participants are ultimately responsible for decision-making.

To facilitate broad participation, open policy meetings are hosted by RIRs regularly in each of the regions. Ongoing discussions are carried out on the public mailing lists of each RIR, which are open to both the RIR constituents and the broader community. The RIRs also participate actively in other Internet conferences and organizations and, importantly, each RIR has a strong tradition of participating in the public activities of the others.

A current example of the coordinated efforts of the RIRs is the Provisional IPv6 Assignment and Allocation Policy Document, a joint effort of the RIRs with the assistance of the IETF, The *Internet Architecture Board* (IAB), and the *Internet Engineering Steering Group* (IESG) to describe the allocation and assignment policies for the first release of IPv6 address numbers.

Also, the RIRs recently published the RIR Comparative Policy Overview, which is available at: **`http://www.ripe.net/ripencc/mem-services/registration/rir-comp-matrix-rev.html`**

These documents help illustrate that the well-established combination of bottom-up decision-making and global cooperation of the RIRs has created a stable, robust environment for Internet address management.

RIR Functions

The primary function of each RIR is to ensure the fair distribution and responsible management of IP addresses and the related numeric resources that are required for the stable and reliable operation of the Internet. In particular, the resources allocated, assigned, and registered by RIRs are Internet address numbers (IPv4 and IPv6) and AS numbers. RIRs are also responsible for maintaining the reverse delegation registrations of the parent blocks within their respective ranges.

Complementing their registry function, the RIRs have an important role in educating and informing their communities. The activities carried out by the individual RIRs vary, but include open policy meetings, training courses, seminars, outreach activities, statistical reporting, and research.

Additionally, a crucial role for the RIRs is to represent the interests of their communities by participating in global forums and providing support to other organizations involved in Internet addressing issues.

RIRs and The Global Internet Community:

Formation of ICANN and the ASO

The global Internet governance landscape began to undergo radical changes in mid-1998, with the publication of a U.S. Government white paper outlining the formation of a "not-for-profit corporation formed by private sector Internet stakeholders to administer policy for the Internet name and address system." ICANN was formed later that year.

At the heart of the ICANN structure are "supporting organizations" that are formed to "assist, review and develop recommendations on Internet policy and structure" within specialized areas. In October 1999, the existing RIRs and ICANN jointly signed a *Memorandum of Understanding* (MoU) to establish the principles for forming and operating the *Address Supporting Organization* (ASO). It is intended that new RIRs will sign the MoU as they emerge.

Under the ASO MoU, the policy forums within each of the RIR regions continue to be responsible for development of regional IP address policy. In addition, each signatory RIR is responsible for electing three members to the ICANN *Address Council*.

The purpose of the Address Council, as described in the MoU, is to review and develop recommendations on issues related to IP address space, using the open processes that exist in the three regions; and to advise the ICANN Board on these matters. In addition, the Address Council is responsible for the appointment of three ICANN Directors to the ICANN Board.

RIR–ASO Coordination

Since the formation of the ASO, the RIRs have played an integral part in facilitating its activities. By joint agreement, the RIRs will share the ASO secretariat duties, including the hosting of the ASO Web site, on a revolving basis. APNIC provided these services in the ASO's first year of operation, and RIPE NCC is currently performing this role.

The ASO Address Council holds monthly telephone conferences, which are attended by representatives of the RIRs (and emerging RIRs on a listener basis). In accordance with the MoU, the ASO also holds regular open meetings in conjunction with the open policy meetings of the RIRs.

RIRs and Industry Development

As noted previously, the RIRs maintain high levels of participation in the conferences and activities of other organizations. Similarly, they invite the participation of interested parties in their own activities.

The RIRs are active in many areas of new technology implementation (such as *General Packet Radio Service* [GPRS] and *Universal Telecommunications System* [UMTS] mobile telephony, IPv6, and cable and *Digital Subscriber Line* [xDSL]-based Internet services).

The established regional processes have proved both flexible and open enough to incorporate such new developments into policy formation. Industry representatives frequently join policy discussions, present at plenary sessions, and participate in working groups.

The RIRs pursue relationships with industry bodies, particularly those with representative and developmental functions, to facilitate industry convergence on open standards and policy processes.

Many diverse parties have legitimate interests in the allocation and registration of IP addresses, and the RIRs remain committed to participating with these parties to achieve a consensus among the Internet community on IP address allocation issues.

The Future of RIRs

In Internet time it can be easy to forget that eight years is actually not long. Since it was first proposed in 1990, the RIR system has evolved rapidly, enjoyed strong community support, and has been relatively free of the political wrangling that has characterized the registration systems of other Internet resources. Without doubt, this position is largely due to the early determination to provide accessible, open forums for the interested stakeholders in the various regions.

New technologies, such as GPRS, broadband services, and IPv6 may raise operational and policy challenges to the RIRs, yet at the same time they bring opportunities for increased global cooperation, in a context where distinct regional concerns are represented more effectively than ever before.

It is hoped that the emergence of new RIRs will only serve to expand and enhance the inclusive nature of RIR activities.

References

[1] Clark, D., and Cohen, D., "A Proposal for Addressing and Routing in the Internet," IEN 46, June 1978.

[2] Postel, J., "Assigned Numbers," RFC 790, September 1981.

[3] Information Sciences Institute, "Internet Protocol, DARPA Internet Program, Protocol Specification," RFC 791, September 1981.

[4] Cerf, V., "IAB Recommended Policy on Distributing Internet Identifier Assignment and IAB Recommended Policy Change to Internet 'Connected' Status," RFC 1174, August 1990.

[5] Williamson, S., and Nobile, L., "Transition of NIC Services," RFC 1261, September 1991.

[6] Fuller, V., Li, T., Yu, J., and Varadhan, K., "Supernetting: An Address Assignment and Aggregation Strategy," RFC 1338, June 1992.

[7] Gerich, E., "Guidelines for Management of IP Address Space," RFC 1366, October 1992.

[8] Gerich, E., "Guidelines for Management of IP Address Space," RFC 1466, May 1993.

[9] Rekhter, Y., and Li, T., "A Border Gateway Protocol 4 (BGP-4)," RFC 1654, July 1994.

[10] Hubbard, K., Kosters, M., Conrad, D., Karrenberg, D., and Postel, J., "Internet Registry IP Guidelines," RFC 2050, November 1996.

[11] Blokzijl, R., Devillers, Y., Karrenberg, D., and Volk, R., "RIPE Network Coordination Center," RIPE-19, September 1990.

[12] Terpstra, M., "RIPE NCC Internet Numbers Registration Procedures," RIPE-65, July 1992.

DANIEL KARRENBERG has helped to build the European Internet since the early 1980s. As one of the founding members of the German UNIX Users Group, he has been involved in the setting up of EUnet, a pan-European coperative network providing electronic mail and news to businesses and academic institutions all over Europe. While at CWI in Amsterdam, Karrenberg helped to expand this network and convert it to a fully IP-based service. During this time he created a whois database of operational contacts, which was the nucleus of the current RIPE database. Karrenberg is one of the founders of RIPE, the IP coordination body for Europe and surrounding areas. In 1992 he was asked to set up the RIPE NCC, the first regional Internet registry providing IP numbers to thousands of Internet service providers in more than 90 countries. Karrrenberg led the RIPE NCC until 1999, when it had an international staff of 59 with more than 20 nationalities; he currently helps to develop new RIPE NCC services. Recently his contributions have been recognized by the Internet Society with its *Jon Postel Service Award*. Karrenberg's current interests include measurements of Internet performance and routing as well as security within the Internet infrastructure. In general he likes building new and interesting things. Mr. Karrenberg holds an MSc in computer science from Dortmund University. E-mail: **Daniel.Karrenberg@ripe.net**

GERARD ROSS holds a BA and LLB from University of Queensland and a Grad.Dip. (Communication) from Queensland Institute of Technology. He was employed as the technical writer at APNIC in 1998 and has been involved in the development and drafting of several major policy documents both in the APNIC region and as part of coordinated global RIR activities. He was the ASO webmaster in its inaugral year. He is currently the APNIC Documentation Manager. E-mail: **gerard@apnic.net**

PAUL WILSON has been Director-General of APNIC since August 1998. Previously, he was a founding staff member and subsequently Chief Executive Officer at Pegasus Networks, the first private ISP in Australia. Over an eight-year period he worked as a consultant to the United Nations and other international agencies on Internet projects in many countries. Since 1994, he has worked with the International Development Research Centre (IDRC) on its Pan-Asia Networking (PAN) Programme, supporting projects in Mongolia, Vietnam, Cambodia, Maldives, Nepal, Bhutan, PNG, and China. He continues to serve as a member of the PAN Research and Development Grants Committee. E-mail: **pwilson@apnic.net**

LESLIE NOBILE received her B.A. from the American University in Washington, D.C. She has over 15 years of experience in the Internet field, and has been involved with the Internet Registry system since 1991. Prior to that, she held various technical management positions while working under a U.S. Government contract that supported the engineering and implementation of the Defense Data Network, a high-speed data network that evolved from the ARPANET. Her experience with the Registry system began in 1991 working as one of the Operations managers who transitioned the Internet Network Information Center (NIC) from SRI to Network Solutions, Inc. She remained a registration services manager with the DDN/DoD NIC until August 2000, when she became Director of Registration Services at the American Registry for Internet Numbers (ARIN). She has been a contributing author to RFCs, Internet Society (ISOC) articles, and various other industry publications and has been actively involved in the global coordination of Internet addressing policy. Her e-mail address is **leslie@arin.net**

Book Reviews

Web Caching

Web Caching by Duane Wessels, ISBN 1-56592-536-X, O'Reilly, June 2001.

It's always a pleasure to read a technical book written by someone who has not just studied the topic, but has been so involved that he has spent years living and breathing the subject. Such books do more than just describe the technology, because they are invariably able to add a dimension of deeper insight and interest, and in so doing, bring the topic to life for the reader. Duane Wessel's experiences in the Harvest project, and then as self-confessed "Chief Procrastinator" in the *Squid* Web cache project, certainly place him in the category of an author who has lived the topic. The outcome is a well-researched and very readable book on the topic of Web caching.

Web Caching

Web caching has been an integral part of the architecture of the World Wide Web since its inception, and is now a broad topic encompassing a range of approaches, a range of technologies, and a range of deployment issues for the end consumer, the content publisher, and the service provider intermediaries. The book starts with a clear introduction that outlines the elements of the architecture of the Web, and describes the terminology used within the book. This section also provides a basic introduction to the operation of the *Hypertext Transfer Protocol* (HTTP). This section also describes the various forms of Web caches that are in use today.

The way in which a cache interprets the directives at the header of a delivered Web object is described in some detail. I learned something unexpected here, in that a Web object that includes a directive of the form "Cache-control: no-cache" is defined in RFC 2616 as allowing a cache to store a copy of the object and use it, subject to revalidation, for subsequent requests. It seems that if you really want the object not to be stored in a cache, then "no-store" is what you are after, because "no-cache" allows the object to be cached! As well as describing the definition of the cache control directives, this section provides a clear explanation of how document ageing is defined, and when a cache server determines that a cached object should be checked against the original to ensure that the cached copy remains a faithful reproduction.

Caching has its champions and its detractors, and the book attempts to present both perspectives in a balanced fashion. On the positive side, caching is seen as an effective way to improve the performance of the delivery of Web-based services, and to relieve network and server load. The claim is made here that a large busy cache can achieve a hit ratio of some 70 percent. Don't get too enthusiastic, however, because a more common achieved ratio is somewhere between 30 and 40 percent.

On the negative side is the ever-present issue of accuracy of the cache, the inability for a content provider to track contact access, and the issue of integrity of the cache in the face of service attacks that are directed to the cached copy of the content.

The Politics of Caching

This section of the book intrigued me, because it is certainly rare to see a technical book address the various social implications of the technology. The study includes the issues of privacy, request blocking, copyright control, content integrity, cache busting, and the modifications to the trust model in the presence of cache intermediaries. The book exposes the tension between the content provider, the user, and the service provider. The content provider would generally like to exercise some control over tracking who is accessing the content and how each client uses the content and how they navigate through the Web site. The user is interested in efficiency of content delivery, and also has to place a high level of trust in the integrity of the content-delivery system. The service provider is also interested in rapid delivery of content, as well as managing network load. Third parties, such as regulatory or law-enforcement bodies, may be interested in ensuring that the content originator is unambiguously traceable, and that various regulations with respect to content are enforced by content originators and service providers.

Practical Advice

From this overview, the book moves onto more practical topics, and first describes how to configure browsers to take advantage of caches. It also covers how various proxy auto-configurators work. The topic that has generated some attention is that of *interception caching,* where a user's Web-browser commands are intercepted by a provider cache without the direct knowledge of the user of the user's browser. The techniques of implementing such interception caches are described, including a description of the operation of the *Web Cache Coordination Protocol* (WCCP), policy routing, and firewall interception. Interception caching, or transparent caching, is a topic that has generated its fair share of controversy in the past, and the book does take the time to clearly describe the issues associated with this caching approach.

The other topic covered under the general topic of practical advice is advice to server operators and content providers on how to make servers and content work in a predictable fashion with caches, describing which HTTP reply headers affect cacheability. This section provides advice on how to build a cache-friendly Web site, and motivates this with reasons why a content provider would want to ensure that content is readily cacheable. This includes some practical advice on how a content provider can still receive hit counts and site navigation information while still allowing the content of a site to be cached.

Fun with Caches—Cache Hierarchies and Clusters

Although caches can operate in a standalone configuration, it is possible to interconnect caches so that a cache will refer to another cache in the event of a cache miss, rather than directly refer to the origin server. I gather that the author is not overly keen on such an approach, given that the arguments against such configurations consume five times as much space as the arguments in favor! The alternative to a strict hierarchy is a set of cooperating peer caches, together with an intercache protocol to allow a cache to efficiently query its peers for an object. The book describes the *Internet Cache Protocol* (ICP), the *Cache Array Routing Protocol* (CARP), which is pointed out to be an algorithm, not a protocol, despite its name, the *Hypertext Caching Protocol* (HTCP), and *Cache Digests*. The scenarios where each approach would be preferred is a helpful addition to this section. Cache clusters are also described; if I have a criticism of the book, it is that this section is too terse—I was looking for more details of cache-balancing and content-distribution techniques.

Cache Operation

The final section of the book looks at the tasks associated with designing, benchmarking, and operating cache servers. How much disk space is enough for a cache? How much memory? Where should the caches be placed in the network? What aspects of the cache operation should you monitor? And if you are considering purchasing caches, what aspects of the cache should you carefully examine?

Conclusion

This is not a book about how to build a cache, although if you are considering doing that it's a good place to start your research. Nor is it a book about every detail on how to operate a cache. But if you are operating a cache, it will be useful. Although it's not a book about how to operate a Web server, if you are operating a Web server, then caches will attempt to store your content, and this book will help you configure your server to interoperate predictably with caches.

The Web is a large part of today's Internet, and Web caches can make the Web faster, more efficient, and more resilient. If you want to understand how caches work and understand how you can use caches to improve the user's experience rather than making things worse, then this book is essential reading.

—Geoff Huston
gih@telstra.net

IPSec *IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks,* by Naganand Doraswamy and Dan Harkins, ISBN 0-13-011898-2, 1999, Prentice Hall PTR Web Infrastructure series. **http://www.phptr.com**

We all know that Internet security is a major concern. Evolving technologies such as *Virtual Private Networks* (VPNs) are making it easier to deploy secure networks at low costs. VPN technology is based upon encryption techniques that make use of different algorithms. Most of these algorithms are specified in the form of *Requests for Comments* (RFCs). Though RFCs provide the minute details, they are not exactly lively reading. This is where the *IP Security* (IPSec) book comes in handy. The authors have done their best to explain IPSec technology in layman's language, although one encounters a lot of technical jargon in this book.

Organization

The book is divided into three parts. Part I gives a history of cryptography and techniques and cryptographic tools, and overviews of TCP/IP and IPSec. Authentication methods such as *Public Key Infrastructure* (PSI), RSA, and DSA are discussed. Key exchange methods such as Diffie-Hellman and RSA Key Exchange are discussed, along with their advantages and disadvantages. IPSec architecture is explored in the IP Security Overiew section, which describes the security services provided by IPSec, how packets are constructed and processed, and the interaction of IPSec processing with policy. IPSec protocols—*Authentication Header* (AH) and *Encapsulation Security Payload* (ESP)—are the basic ingredients of the IPSec stack to provide security. Both AH and ESP can be operated in either the transport mode or tunnel mode. Part II offers a detailed analysis of IPSec, the different modes, IPSec implentation, the ESP, AH, and the *Internet Key Exchange* (IKE). The authors do a good job of describing the IPSec road map, which defines how various components within IPSec interact with each other. Detailed packet formats of different IPSec formats are discussed in Chapter 4. ESP, AH, and IKE are discussed in depth in Chapters 5 through 7. Part III deals with most of the deployment issues concerned with IPSec, as well as policy definition, policy management, implementation architecture, and end-to-end security are discussed in this section. Chapter 11 discusses the future of IPSec and what it means to the world of security. Though IPSec may be thought of as a totally secure method of communication, it has its conflicts when it comes to *Network Address Translation* (NAT), multicasting, and key management in a multicast environment.

Prerequisites

Although the authors have done a good job delivering the IPSec concept, understanding this text requires more than basic computer and communication concepts. One should understand hacking and different types of Internet attacks. OSI layer details and packet-level understanding of every layer within the OSI model is a must.

—Manohar Chandrashekar, WorldCom Inc
mchandra@wcom.com

Letters to the Editor

ICANN

Mr. Jacobsen,

I very much enjoy the *Internet Protocol Journal* and put it at the top of my reading stack as soon as it is received. In particular, I enjoy the standards and high technical detail and view it as a safe place from overt commercial advertisement and politics.

That is why I was disappointed by the article from Mr. Lynn. My opinion of ICANN is that it is undemocratic in any tradition, uninterested in experimentation, and uninterested in outside views. I took offense at his continued use of the phrase "public trust" and interpreted the article as propaganda. Further, I found the technical content of the article to be zero.

On the other hand, William Stallings article on MPLS was exactly the kind of article I've come to enjoy. I wasn't familiar with MPLS and the article helped me understand the concepts, vocabulary, and high-level issues. I hope that "MPLS" serves as a model of the articles in future IPJ issues.

I keep back issues of IPJ in a binder and continue to hope you uncover more articles like "The Social Life of Routers." My copy of Mr. Krebs article has notes in all the margins—I was excited—but it was a twist on something that I thought I knew and he exposed a different design vocabulary by making an unexpected comparison.

I apologize for complaining about something that is a gift from Cisco; I do understand how crass that is. I hope that you will interpret my note in a complementary manner: I've come to respect the journal and found that it fits an unfilled niche in my reading.

—Brent D. Stewart, Global Knowledge
<brent@stewart.hickory.nc.us>

Brent,

I appreciate your feedback, as I am sure Mr. Lynn will if you send it to him. The article was, after all, published for public comment.

ICANN has unfortunately tended to polarize people and has become a forum in which a certain amount of politics is played out. I don't think this is entirely ICANN (the board)'s fault. What was set up as an organization to take over the work of one man—the late Jon Postel, is seen by some as an opportunity for "Internet Governance" and "world-wide electronic democracy."

Having watched the ICANN process since its beginnings in 1998, I would say that Mr. Lynn's version of history is pretty much on target. When the IANA was in the hands of Jon Postel, it most certainly was a "public trust" (a limited resource to say the least), and if ICANN does not take that responsibility seriously, it certainly will have failed.

However, I do not think this is the case. Yes, ICANN is now a fairly large and slow moving machinery, and I would have liked to see more new domains deployed sooner, but to some extent the slowness is caused by the structure of Supporting Organizations as much as it is by the board itself. There is a lot to sort out, a lot to comment on, and *many* divergent views are indeed being expressed in all kinds of ICANN forums, including the public meetings. So, I cannot agree that ICANN is "uninterested in outside views." A perfect democracy it is not, nor was it ever intended to be, and yes, some of the topics on the agenda such as the *Uniform Dispute Resolution Process* (UDRP) are indeed non-technical. But it is not as if ICANN had much choice in that particular matter. (Although some would argue that it could be moved outside the ICANN process.)

Being part of the ICANN process, through e-mail discussion, public meetings or through the Supporting Organizations is not difficult. Nor do I think that ICANN ignores any of the feedback it gets.

Back to the article. No, it was not particularly technical, but if you read IPJ's Call for Papers you will see that it mentions "Legal, policy and regulatory topics..." Also, in the wake of September 11, I though it was important to provide some background on the thinking of ICANN, and why they chose to refocus the most recent meeting on security etc. IPJ, by the way, also encourages the occasional "Opinion Piece," although the article by Mr. Lynn was not intended as such. The issue of alternate roots is indeed a matter of debate, and while the the IAB has already expressed its view, I appreciate that there might be other (valid) ones.

In any case, thank you for taking the time to write. I certainly don't intend to steer IPJ away from topics such as MPLS and I hope that the occasional policy or even opinion piece won't steer you away from IPJ.

—*Ole Jacobsen, Editor and Publisher* `<ole@cisco.com>`

MPLS Ole,

William Stallings otherwise-excellent article on MPLS in the *Internet Protocol Journal* Vol. 4, No. 3 had a serious error in it with respect to Virtual Private Networks (VPNs). He said that MPLS is an efficient mechanism for supporting VPNs and that MPLS provides security; neither is true.

As the rest of the article shows, MPLS provides a transport tunnel for IP packets, meaning that it helps create virtual networks. However, there is no privacy on those virtual networks, so it is inappropriate and probably dangerous to call MPLS tunnels virtual private networks.

To most Internet users, security means preventing snooping of sensitive traffic, preventing malicious changes to content, or both. MPLS does not provide either service. Instead of relying on insecure MPLS, users who want secure tunnels use systems that employ the IPsec protocol.

Many dozens of vendors supply IPsec systems appropriate for everything from tiny home offices to gigantic telco central switches, all with the same high security. Although the article showed that MPLS has many valuable features, IPJ readers should not fall into the trap of thinking that VPN support or security are MPLS features.

—Paul Hoffman, Director, VPN Consortium
<paul.hoffman@vpnc.org>

Ed: We presented this letter to a panel of experts, and here are some samples of the responses we received:

The term "VPN" has been used in many different contexts. I saw a group once call a VLAN a "VPN" as well. I honestly couldn't say that they were incorrect. It may be appropriate to say that there are IPsec VPNs and that there are MPLS VPNs, but I have a problem calling one "right" and another "wrong" simply because of some perceived, implied definition of the security level that should be provided by a "VPN." Most people support the notion that an MPLS VPN provides about as much "security" as a Frame Relay link. This amount of "security" in a VPN is acceptable to many people.

—Chris Lonvick, Cisco Systems **<clonvick@cisco.com>**

We have different views on security, I'm sure. One view is that a secure private network: a) ensures that a third party cannot impose a condition on the network such that a customer's traffic is directed to another customer b) ensures that a third party cannot inject traffic into a customer's private network, c) a third party cannot alter customer traffic and d) a third party cannot discern that communications is taking place between two parts of a private network.

MPLS uses the same mechanisms as X.25, ATM and Frame, and has similar properties—the objectives above can be met with adequate confidence as long as the network is carefully configured and managed.

Edge to edge IPSec has a different set of security principles—the basic mode of operation is that such networks may be subject to attacks that redirect customer's traffic to third party sites, and allow third parties to inject traffic into the VPN, and allow a third party to discern that communications is taking place within a private context. The essential attribute of edge to edge IPSec is that the encryption is intended to ensure that leakage can be identified: foreign injected traffic or altered traffic can be identified and rejected and leaking traffic cannot be decoded.

Both approaches have vulnerabilities and weaknesses. The first approach places trust in the integrity of the host platform. The second approach is prone to various forms of DOS attacks and traffic profiling.

But I would not concur with a view that labels the MPLS approach as inefficient or insecure, nor would I label X.25 networks, ATM or Frame as *intrinsically* inefficient and insecure. There are insecure operating practices and there are cautious operating practices.

IPSec networks have similar issues—relating particularly to the vulnerabilities of third party disruption and profiling eavesdropping.

So it's not that I believe that all MPLS networks are well designed and well operated—on the contrary! But as an architectural approach I am not able to agree with a comment that appears to condemn MPLS as intrinsically a poor choice for a VPN host technology.

So if the comment is that the article provides the impression that MPLS is such a robust technology that it creates secure private network applications such as VPNs, and appears to make this assertion so strongly that it gives the impression that this outcome occurs irrespective of MPLS network design and operating practices, and that this impression is ill-founded, then I would agree entirely with Mr. Hoffman. Secure networks, or at least robust networks, are a result of careful choice of technologies coupled with careful design and careful operation.

—*Geoff Huston, Telstra* **<gih@telstra.net>**

Ed.: We forwarded these comments to Mr. Hoffman, and he responded:

Geoff believes that it a network that does not prevent an active attacker from seeing or modifying traffic, and does not prevent a passive attacker from seeing packets, is secure and private; I do not. The fact that MPLS restricts the flow of traffic to a particular defined network is sufficient for him; it is not for me, given the fact that an attacker breaking into any node on that defined network can compromise the privacy and integrity of the traffic.

It is typical for ISPs to not want to do the work of actually securing the traffic they say they have put in a VPN by using IPsec. That work is not cheap, and takes more management than vanilla MPLS, but it is the only way to really secure the data. I am absolutely not saying that the IPsec community is without blame here: we have a tendency to ignore the valuable features of MPLS and have done almost nothing to make it easier to intelligently tunnel IPsec in MPLS (we also pretty much stonewalled the IPsec under L2TP work that is now finally standardized). But our lack of openness doesn't make MPLS a VPN technology.

—*Paul Hoffman, Director, VPN Consortium*
<paul.hoffman@vpnc.org>

Ed.: We would love to hear from you. Please send your letters to: **ipj@cisco.com**

Fragments

ACM Assembles Security and Privacy Panel

Prompted by increased public concerns about personal privacy and the security of networked information systems, the *Association for Computing Machinery* (ACM) has announced the formation of a new *Advisory Committee on Security and Privacy* (ACSP). Led by Peter Neumann and Eugene H. Spafford, the ACSP brings together a dozen leaders and innovators in the field of privacy and information assurance to serve as a powerful resource for the ACM community and the public at large.

Comprising experts from research, industry, academia, and government, the diverse group represents a wide range of viewpoints. Commenting on the formation of the ACSP, Co-Chair Peter Neumann noted, "The ACSP will provide timely and accurate assessments of situations relating to information security that are otherwise clouded by confusion, uncertainty, and often, misinformation."

Added ACSP Co-Chair Gene Spafford, "Until recently, computing professionals have been primarily concerned with making computers work consistently, cheaply, and effectively. Now it is critical that we also bring expertise to bear on how computers can be made to operate safely, keep information resources secure from attack, and protect privacy."

The ACSP consists of 12 distinguished members with expertise in information security and assurance, privacy, cybercrime, and allied fields. The group will coordinate with other ACM Committees, including the *U.S. ACM Committee on Public Policy* (USACM) and ACM Law Committee, to provide objective advice to the computing community, the public at large, and to policy-makers. ACSP is expected to provide statements and testimony on information security and privacy issues, as well as undertaking studies of related topics. For more information about the ACSP, see the web site at:
`http://www.acm.org/usacm/ACSP/homepage.htm`

Members of the ACSP (affiliations provided for identification purposes only) are:

Steve Bellovin (AT&T Labs Research)
Matthew Blaze (AT&T Labs Research)
David Clark (MIT)
Dorothy Denning (Georgetown University)
Ed Felten (Princeton University)
David Farber (University of Pennsylvania)
Susan Landau (Sun Microsystems)
Robert Morris (Dartmouth College)
Peter Neumann (SRI International)
Fred Schneider (Cornell University)
Eugene H. Spafford (Purdue University CERIAS)
Willis Ware (RAND Corporation)

For more information, see ACM's Web site at: **`http://www.acm.org`**

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

March 2002 Volume 5, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

Major Internet events such as the IETF meetings, the Regional Internet Registry meetings, APRICOT, SIGCOMM, and NetWorld+Interop to name a few, all provide Internet access for attendees. Commonly referred to as the "Terminal Room," these facilities have evolved into complex high-speed networks with redundant paths, IPv6 routing, multicast, and more. In the last five years or so, these networks have also been providing wireless access using various flavors of the IEEE 802.11 standard. As I write this, I am sitting in the lobby of the Minneapolis Hilton Hotel, where the 53rd IETF meeting is being held. The lobby area and two floors of meeting rooms have IEEE 802.11 coverage, and a directional high-gain antenna provides access in the pub across the street. Wireless Internet computing is a reality, at least when you have a large gathering of engineers such as an IETF meeting. In our first article, Edgar Danielyan takes a closer look at this technology, its applications and evolution.

More and more software is being distributed via the Internet rather than through the use of conventional media such as CD ROMs or floppy disks. Downloading software via the Internet is very convenient, especially if you have reasonably high bandwidth. However, with this convenience comes a certain risk that you may be receiving a modified copy of the software, perhaps one that contains a virus. Code signing is a method wherein software is cryptographically signed and later verified. Eric Fleischman explains the details of code signing.

I should have known better than to announce the imminent availability of our online subscription system in the previous issue. We are working on it, but it isn't ready yet, so please continue to send your subscription requests and updates to: **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

IEEE 802.11

by Edgar Danielyan

Introduced in 1997, the IEEE Standard 802.11 for wireless local-area networks has seen modifications and improvements in the past years and is promising a brighter wireless future, so yearned for by many of us. However, during its lifetime, the standard also has had a few setbacks, which are reminders that nothing is perfect in this world, much less in networking. This article provides a brief but comprehensive introduction to IEEE 802.11 wireless networking, its present and future, and highlights some of its security, performance, and safety aspects.

IEEE 802.11

The initial IEEE Standard 802.11 was published by the *Institute of Electrical and Electronics Engineers* (IEEE) in 1997. That standard is known as IEEE 802.11-1997 and is now updated by the current standard, IEEE 802.11-1999. The current standard has also been accepted as an American national standard by the *American National Standards Institute* (ANSI) and has been adopted by the *International Organization for Standardization* (ISO) as ISO/IEC 8802-11:1999. The completion of IEEE 802.11 in 1997 set in motion the development of standards-based wireless LAN networking. The 1997 standard specified a bandwidth of 2 Mbps, with fallback to 1 Mbps in hostile (noisy) environments with *Direct Sequence Spread Spectrum* (DSSS) modulation, and bandwidth of 1 Mbps with *Frequency Hopping Spread Spectrum* (FHSS) modulation, with possible 2-Mbps operation in friendly (noiseless) environments. Both methods operate in the unlicensed 2.4-GHz band. What is less known about IEEE 802.11 is that it also defines a baseband infrared medium, in addition to the DSSS and FHSS radio specifications, although its usefulness seems somewhat limited. There are also several task groups inside the 802.11 working group itself that work on substandards of 802.11:

- 802.11D: Additional Regulatory Domains
- 802.11E: Quality of Service (QoS)
- 802.11F: Inter-Access Point Protocol (IAPP)
- 802.11G: Higher data rates at 2.4 GHz
- 802.11H: Dynamic Channel Selection and Transmission Power Control
- 802.11i: Authentication and Security

The IEEE 802 group has an official Web site at **`www.ieee802.org`**, and IEEE 802.11 has an official Web site at **`www.ieee802.org/11/`**.

DSSS

Direct Sequence Spread Spectrum (DSSS) is one of the modulation techniques provided for by the IEEE 802.11 and the one chosen by the 802.11 Working Group for the widely used IEEE 802.11b devices. DSSS modulation is governed in the United States by FCC Regulation 15.247 and in Europe by ETSI Regulations 300-328. DSSS in IEEE 802.11 uses *Differential Binary Phase Shift Keying* (DBPSK) for 1 Mbps, and *Differential Quadrature Phase Shift Keying* (DQPSK) for 2 Mbps. The *Higher-Rate DSSS* (DSSS/HR) defined in IEEE 802.11b uses *Complementary Code Keying* (CCK) as its modulation scheme and provides 5.5- and 11-Mbps data rates. Because of their compatibility, all three modulation schemes can coexist using the rate-switching procedures defined in the IEEE 802.11. The *Orthogonal Frequency Division Multiplexing* (OFDM) used by the IEEE 802.11a is regulated in the United States by Title 47 Section 15.407 of the U.S. *Code of Federal Regulation* (CFR). IEEE 802.11a uses a system of 52 subcarriers modulated by BPSK or QPSK and 16-quadrature amplitude modulation. It also uses *forward error correction* (FEC) coding, also used by the Digital Video Broadcasting (DVB) standard with coding rates of 1/2, 2/3, and 3/4.

FHSS

Although specified by the original IEEE 802.11, *Frequency Hopping Spread Spectrum* (FHSS) modulation is not favored by vendors and, it seems, the 802.11 working group itself. DSSS has won the battle—very few vendors support 802.11/FHSS, and further developments with 802.11 use DSSS. Some have expressed ideas that frequency hopping in FHSS may contribute to the security of 802.11, but these are invalid expectations—the hopping codes used by FHSS are specified by the standard and are available to anyone, thus making the expectation of security through FHSS unreasonable.

Two supplements to the IEEE 802.11-1999, known as IEEE 802.11a and IEEE 802.11b, brought considerable changes and improvements to the IEEE 802.11-1999 standard.

IEEE 802.11a

IEEE 802.11a specifies a high-speed physical layer operating in the 5-GHz unlicensed band utilizing a complex coding technique known as OFDM. The data rates specified by IEEE 802.11a are 6, 9, 12, 18, 24, 36, 48, and 54 Mbps, with support for 6, 12, and 24 Mbps as a mandatory requirement. IEEE 802.11a is seen by some in the industry as the future of IEEE 802.11. Some products already implement the IEEE 802.11a, such as the chip from Atheros (**www.atheros.com**) and a PCMCIA/CardBus adapter from Card Access Inc (**www.cardaccess-inc.com**) based on it. However, 802.11a is not without disadvantages. The increased bandwidth of IEEE 802.11a results in a shorter operation range.

Additionally, because of the protocol overhead and interference/error correction, the real bandwidth may be considerably less than the nominal. New surveys and installation will also be required in many cases; the underlying infrastructure will also be more expensive because of the shorter operation range (about 1/3 of 802.11b) and higher density of *base stations* (also known as *access points*).

IEEE 802.11b

Probably the most widely implemented and used wireless LAN technology today, IEEE 802.11b specifies 5.5- and 11-Mbps data rates (in addition to the already specified 1 and 2 Mbps), but operates in the original 2.4-GHz band also using DSSS modulation. Most currently selling IEEE 802.11 products implement IEEE 802.1b. IEEE 802.11b-compliant devices can operate at 1, 2, 5.5, and 11 Mbps.

It is important to note that both incarnations of IEEE 802.11 use the same *Media Access Control* (MAC) protocol, *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA); therefore, these modifications affect only the physical layer (PHY layer in IEEE parlance) of the standard. The 1/2- and 5.5/11-Mbps DSSS (IEEE 802.11b) networks can coexist, enabling a painless transition to IEEE 802.11b (High Rate) at 11 Mbps. Eleven to fourteen radio channels are available for use with IEEE 802.11b in the 2.4-GHz band, depending on the local legal and administrative restrictions.

Distance, Power, and Speed Issues

It is obvious that all three of these parameters of wireless systems are interconnected. However, as with other radio-based technologies, the external conditions (such as the line of sight in case of outdoor use) greatly affect the operation of IEEE 802.11 devices.

Antennae

Antennae used with IEEE 802.11b devices may be grouped into two categories: *omnidirectional* and *point-to-point*. Obviously, omnidirectional antennae are the easiest to use, because they do not require positioning. Omnidirectional antennae are used in most base stations, as well as in most access cards. However, because of their nature, omnidirectional antennae do not work well over longer distances, unless used with external amplifiers; and these are not always legal or appropriate to use. Directional, or point-to-point antennae, on the other hand, require careful positioning and are used outdoors. Although the typical range for an omnidirectional antenna system is 150 ft (45m), configurations with high-gain directional antennae can work on distances up to 25 miles (about 40 km). In localities where amplifiers are allowed, the maximum distance may be considerably increased and is limited only by the line of sight.

Among other factors affecting the operational range of IEEE 802.11b devices are the base-station placement (when used in the infrastructure mode) and radio interference. As mentioned earlier, IEEE 802.11b devices will auto-configure for the highest possible speed and fall back to lower speeds when circumstances so require.

Performance Issues

Aside from obvious factors that affect performance (such as antennae, distance, radio interference) there are numerous other, more subtle issues. In the infrastructure mode, when all devices have to register with the base station(s), the load on the base station(s) increases with the number of clients and may reach a point when the performance reaches unacceptable lows. For example, Apple's AirPort Base Station (Version 2) can support up to 50 simultaneous clients. However, the actual performance of the whole system also depends on the kind of traffic. In particular, isochronous traffic (time-sensitive traffic, such as some types of video, audio, and telemetry), as well as multicast traffic, are particularly taxing for IEEE 802.11 networks and are better kept off the wireless LAN. However, several groups are currently working on extensions to 802.11 to provide for such kinds of traffic in a future version of the standard.

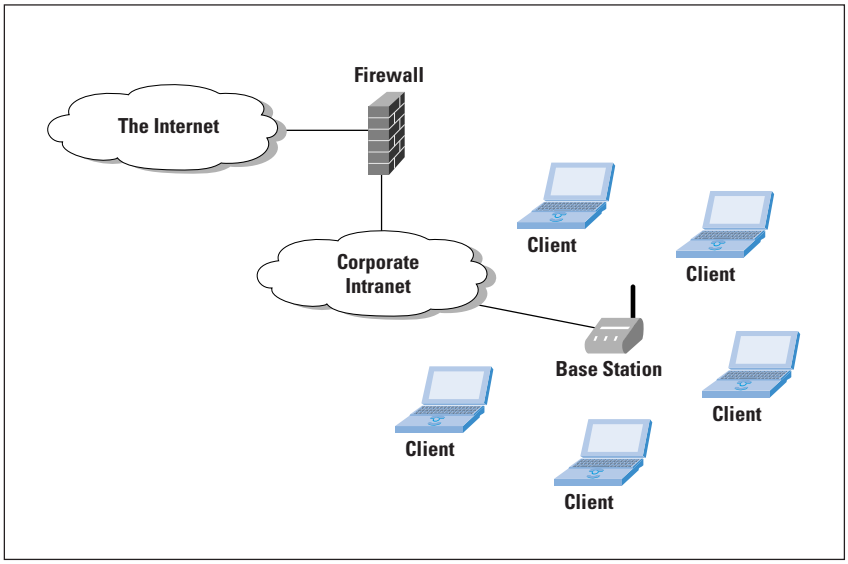


Figure 1: Typical IEEE 802.11 Configuration in Infrastructure Mode

IEEE 802.11 Base Stations and Clients

All IEEE 802.11 devices can be grouped into one of two groups: base stations or clients. Base stations can function as clients; however, not all clients can function as base stations. The reason for this is that base stations are required to provide certain network services to clients (association, distribution, integration, reassociation, and so on) that not all client hardware, firmware, or software can or intended to provide.

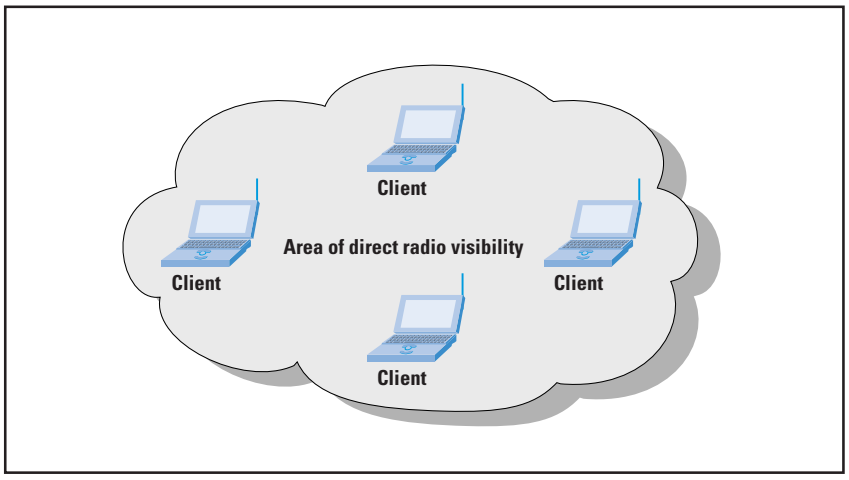
These considerations apply when the infrastructure mode of IEEE 802.11 is deployed. In *ad hoc* networks, where there are no base stations, all clients communicate directly with each other, reminiscent of a traditional shared Ethernet network, with all nodes sharing equal rights and responsibilities. As noted earlier, 11 to 14 radio channels are available, but separate networks may coexist on the same frequency (using different network IDs (*Service Set Identifiers* [SSIDs]), albeit with performance penalties.

The workings of 802.11 devices also differ in the infrastructure and ad hoc modes. In the infrastructure mode (Figure 1), clients associate (and optionally authenticate) themselves with a base station, and the presence of the base station is necessary for the operation of the network.

Complex 802.11 networks may be built using the infrastructure mode, with numerous base stations providing coverage over relatively large physical areas, and clients may roam within this roaming domain, which theoretically may extend from a single building to the entire campus or town. The *Spanning-Tree Protocol* (STP) is usually used in these cases to provide loop-free bridging in this wireless LAN.

In the ad hoc mode (Figure 2), base stations are not used and are not necessary, because all nodes of the wireless LAN have direct reachability (that is, they "see" each other). This mode is usually used in circumstances where all devices are in close proximity to each other (such as a floor or office) and when omnidirectional antennae are used.

Figure 2: IEEE 802.11 ad hoc Network



IEEE 802.11 Roaming and Mobility

IEEE 802.11 provides for roaming and mobility of 802.11 client devices and allows clients to roam among multiple 802.11 base stations that may be operating on the same or different frequencies (channels). This is achieved through the use of *beacon frames,* which are used to synchronize 802.11 devices and, in the infrastructure mode, to associate with a base station.

There are two ways to scan for existing 802.11 networks: active and passive scanning. In active scanning mode, the 802.11 device sends out "probe" frames, soliciting "I am here" responses from existing 802.11 devices. In the passive mode, the devices just listen for beacon frames, which are periodically transmitted by the active devices. In addition, the IEEE 802.11 Task Group F is working on the IAPP, which is to provide better and interoperable mobility and roaming mechanisms.

Security of IEEE 802.11

Up to this point IEEE 802.11 could be considered an absolute success; however, security of IEEE 802.11 is not quite on par with other aspects of the standard. Although an entire chapter (Chapter 8) of the standard is dedicated to authentication and privacy, it is now the common consensus that designers of IEEE 802.11 did not excel in this area. Two reports widely covered in the media, "Your 802.11 Wireless Network Has No Clothes"[7], and "Intercepting Mobile Communications: The Insecurity of 802.11"[6], shed light on the apparent shortcomings of the standard, or to be more exact, on its "vulnerability by design." They demonstrated that although the designers were well aware of the need to plan for authentication and privacy, the actual implementation was not an excellent one. The WEP algorithm, used to provide authentication and privacy in 802.11 wireless networks, is the problem.

WEP

Before discussing the security weaknesses discovered in IEEE 802.11, we quote the aim of the *Wired Equivalent Privacy* (WEP) algorithm as specified in the IEEE 802.11 standard document:

> "Eavesdropping is a familiar problem to users of other types of wireless technology. IEEE 802.11 specifies a wired LAN equivalent data confidentiality algorithm. Wired equivalent privacy is defined as protecting authorized users of a wireless LAN from casual eavesdropping. This service is intended to provide functionality for the wireless LAN equivalent to that provided by the physical security attributes inherent to a wired medium."

As you see, the aim of WEP is to provide a level of privacy equivalent to that of a wired LAN. The wording of standard is very important here: the developers of the standard did not intend to provide a level of security superior to or higher than that of a regular wired LAN, such as Ethernet. The very name of the algorithm, "Wireless Equivalent Privacy," signifies the actual intention of the developers. However, as the practice has shown, the level of security roughly equivalent to the level of security provided by wired LANs is not sufficient—and it is the assumption that "it is OK if wireless LANs are as secure as wired LANs" that is wrong. Other problems, such as the choice of *Cyclic Redundancy Check 32* (CRC-32) instead of *Message Digest Algorithm 5* (MD5) or some other secure hash algorithm, just worsen the problem.

How WEP Works

Let's now look at the workings of WEP. WEP uses a secret key shared between 802.11 nodes to encrypt 802.11 frames (Layer 2). It also uses a checksum (CRC-32) to provide data integrity. The checksum itself is also encrypted using the shared secret key. The decryption is the reverse of the encryption process: the frame is decrypted using the key and the CRC-32 checksum is computed and checked. The cipher used in WEP is RC4, a stream cipher designed by Ron Rivest, and believed to be cryptographically strong. The key is 40 or more bits long (up to 128 bits in some implementations). However, the *Initialization Vector* that is used during the encryption process is only 24 bits long. It is difficult to understand why the designers chose such a small number—more about this later. WEP does not provide any key management—the standard itself does not specify how the shared secret key should be managed and distributed. This leaves one of the most vulnerable parts of any cryptographic system—*key distribution*—open for misuse.

The Borisov Goldberg Wagner Attacks (February 2001)

In their paper entitled "Intercepting Mobile Communications: The Insecurity of 802.11," Nikita Borisov, Ian Goldberg, and David Wagner describe the vulnerabilities present in WEP and attacks against it. In the introduction to their paper, they state:

> "Unfortunately, WEP falls short of accomplishing its security goals. Despite employing the well-known and believed-secure RC4 cipher, WEP contains several major security flaws. The flaws give rise to a number of attacks, both passive and active, that allow eavesdropping on, and tampering with, wireless transmissions."

They go on to say that WEP fails to achieve all three of its security goals, namely confidentiality, access control, and data integrity.

As has been noted earlier, WEP uses the RC4 stream cipher with a 24-bit Initialization Vector for encryption. Borisov, Goldberg, and Wagner show that the poor design of WEP makes the system vulnerable in many areas, and one of the weakest parts of WEP is the 24-bit Initialization Vector, which may result in keystream reuse. Keystream reuse in turn permits successful cryptanalysis attacks against the ciphertext. However, what is surprising is that:

> "The WEP protocol contains vulnerabilities despite the designers' apparent knowledge of the dangers of keystream reuse attacks."

Another not less important but equally poorly designed aspect of WEP is the use of CRC-32. It is known that CRCs are not cryptographically strong and are not intended to be used in place of message digest or hash functions such as MD5 or the *Secure Hash Algorithm* (SHA). Because of the nature of CRC, it fails to provide the required integrity protection.

Some in the industry suggest that MD5 or SHA would introduce performance penalties if used—and indeed they would—one cannot disagree. But let's not forget that CRC-32 was intended as a security measure—which it isn't—yes, it is fast, but it is also insecure. Presumably, a slower but really secure solution is better than an inadequate though fast solution.

The Arbaugh Shankar Wau Attack (April 2001)

In the paper "Your 802.11 Wireless Network Has No Clothes,"[7] authors present their research of the authentication flaws in the IEEE 802.11 and demonstrate a simple eavesdropping attack against IEEE 802.11 authentication. This work is partially based on the knowledge obtained by Borisov, Goldberg, and Wagner in the paper described previously. The attack described in this work is possible even with WEP enabled; however, in that case it will also require application of attack(s) against WEP presented by Borisov et al. The authors also note that a good key management architecture would increase the security of the system; however, in their opinion only a comprehensive redesign of the standard would provide a good long-term solution to these issues.

The Fluhrer Mantin Shamir Attack (August 2001)

Scott Fluhrer, Itsik Mantin, and Adi Shamir describe a passive ciphertext-only attack against the key scheduling algorithm of RC4 as used in WEP[11]. They identify a large number of weak keys, in which knowledge of a small number of key bits suffices to determine many state and output bits with nonnegligible probability. They also show that the first byte generated by the RC4 leaks information about individual key bytes. This paper in particular shows how to reconstruct the secret key in WEP by analyzing enough WEP-encrypted packets. The authors have not tried to do this in practice—others did that.

The Stubblefield Ioannidis Rubin Implementation of Fluhrer Mantin Shamir Attack (August 2001)

In an AT&T Laboratories report published on August 21, 2001[14], Adam Stubblefield, John Ioannidis, and Aviel Rubin describe a real-world successful implementation of the Fluhrer Mantin Shamir attack using a $100 Linksys card on a Linux machine. They report that it took less than a week from ordering the card to recovering the WEP key on a production network. This practical work has shown that no expensive hardware or software is necessary in order to break WEP. They summarize that it is the poor implementation of reasonable secure technologies (such as RC4) that is responsible for WEP weaknesses.

WECA's Response

The *Wireless Ethernet Compatibility Alliance* (WECA) is the organization responsible for certifying compliance with the IEEE 802.11 standards. It also awards the WiFi (*Wireless Fidelity*) industry mark to the products that have passed IEEE 802.11 compliance testing.

In response to the Berkeley paper, WECA has published an official statement, clarifying its understanding of the situation. The main line of this statement is that poor security is better than no security, as well as that WEP was not intended to be a panacea for all security needs. The statement correctly notes that the biggest security threat is the failure to use available protection methods, including WEP.

IEEE 802.11 Chair's Response

In response to the research made at UC Berkeley and the University of Maryland, the Chair of the IEEE 802.11 Working Group, Stuart Kerry, has published a Chair's response intended to clarify some of the issues around the security of IEEE 802.11. He denied allegations made in the media that the security weaknesses of WEP are due to the closed standardization process. In fact, because WEP is a part of IEEE 802.11, it was developed through an open process, like other IEEE standards. The IEEE 802.11 Working Group itself is open to all interested parties to participate. He also rejects the viewpoint that frequency-hopping wireless networks would be less vulnerable to security attacks. It is evident that this is not true because both hopping codes and timing are unencrypted and are available to the attacker. Reminding us that the goal of WEP was to provide a level of security comparable to wired LANs, he states that the IEEE 802.11 Working Group is currently working on improvements to WEP to incorporate better security into the next version of the standard.

IEEE 802.1X

Security in 802.11 networks can be broken down into three components: authentication framework, authentication algorithm/protocol, and encryption. IEEE 802.1X is trying to address the authentication framework part of the puzzle. Although still in development, 802.1X provides a scalable, centralized framework for authentication. 802.1X may deploy a variety of authentication protocols (currently Cisco's *Lightweight Extensible Authentication Protocol* [LEAP] and Microsoft's *Extensible Authentication Protocol – Transport Layer Security* [EAP-TLS] are available), and it works with both wired and wireless LANs. The widely used *Remote Access Dial-In User Service* (RADIUS) protocol is also used in the 802.1X framework. 802.1X/LEAP is available with the Cisco Aironet 350 Series of wireless LAN devices; EAP-TLS is supported in Windows XP. Although it is still a draft, 802.1X may one day become the solution to the authentication issues of 802.11.

IEEE 802.11i

Task Group I of the IEEE Working Group 802.11 is currently defining MAC enhancements to provide enhanced security for 802.11. This is a work in progress, and no IEEE 802.11i draft exists at the time of writing.

Cisco's Solution

Cisco Systems has responded to both papers on the security of the WEP[10]. Cisco agrees that the WEP has serious shortcomings, and states that its Aironet series of wireless networking products offers many solutions to these problems: dynamic WEP keys, secure key derivation, and mutual authentication using LEAP[13]. However, Cisco agrees that improvements are needed in the standard itself.

RC4 Fast Packet Keying for WEP

In a Document Nr 550r2, "Temporal Key Hash," submitted by Russ Housley of RSA Security and Doug Whiting of Hifn to the IEEE 802.11 Working Group, they describe a solution to the WEP problem that uses a hashing technique that rapidly generates a unique RC4 key for each packet of data sent over the wireless network. This technique addresses the performance aspect of the security solution as well—the hash algorithm used in *Fast Packet Keying* (FPK) is much faster than traditional hash algorithms such as MD5 and SHA1 because of the special caching approach. The IEEE 802.11 Working Group has decided to include this technique in the IEEE 802.11i as an informative document. In most cases, FPK may be implemented as a firmware upgrade for the existing hardware. It is possible that when released, IEEE 802.11i may use FPK as the solution—but this decision is yet to be made. No definite plans are announced at the time of writing. For more information, see:
`http://www.rsasecurity.com/rsalabs/technotes/wep-fix.html`.

Health and IEEE 802.11

Concerns about safety and health effects of various wireless solutions such as mobile phones and wireless network devices periodically surface in the media. In particular, the question of whether mobile phones are linked to brain cancer and other diseases is still open. However, in response to these concerns regarding wireless networking equipment health effects, Cisco Systems has published a white paper entitled "Cisco Systems Spread Spectrum Radios and RF Safety," which explains why these devices do not present a threat to human health when correctly used. The bottom line is that devices certified as compliant with U.S. Federal Communications Commission or Industry Canada's regulations are safe to use because of their low emitted power.

Practical Uses

Many companies, such as MobileStar, Wayport, Surf&Sip, and Airwave, have begun providing IEEE 802.11b Internet access at numerous locations throughout the United States. Several international airports also provide 802.11b service free of charge to travelers. No doubt more such services will continue to appear all over the world, maybe making a dream—Internet anywhere—a reality.

Summary

IEEE Standard 802.11 brought the long-awaited standardization to wireless LAN networking. Unfortunately, it also brought various security problems. Despite that, IEEE 802.11 is widely used, and with the coming of IEEE 802.11a, it can only gain in popularity. What now remains to be done is more effective and truly secure privacy and authentication for 802.11 wireless networks.

The IEEE 802.11 Working Group is actively working to improve what has been done to date. The most improvements are obviously needed in the area of security, where Working Groups 802.1X and 802.11i are working to define better security mechanisms. In particular, 802.11 WG is working on a new release of 802.11, which will include improvements over 802.11-1999. In the meantime, consider your wireless LAN as an external, insecure network—just like the Internet—and employ additional security measures, such as Virtual Private Networks, Transport Layer Security, SSH, and IP Security Architecture—in addition to WEP.

References

[1] IEEE Standard 802-1990: "IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture," ISBN 1-55937-052-1.

[2] IEEE Standard 802.11-1999: "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications."

[3] IEEE Standard 802.11a-1999: "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (5 GHz)."

[4] IEEE Standard 802.11b-1999: "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specificiations (2.4 GHz)."

[5] "IEEE 802.11b Wireless Equivalent Privacy (WEP) Security," February 19, 2001, Wireless Ethernet Compatibility Alliance (WECA).

[6] Nikita Borisov, Ian Goldberg, and David Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11."
http://www.isaac.cs.berkeley.edu/isaac/wep-draft.pdf

[7] William A. Arbaugh, Narendar Shankar, and Y.C. Justin Wan, "Your 802.11 Wireless Network Has No Clothes,"
http://www.cs.umd.edu/~waa/wireless.pdf

[8] William A. Arbaugh, "An Inductive Chosen Plaintext Attack Against WEP/WEP2," IEEE Document 802.11-01/230.

[9] J. R. Walker, "Unsafe at Any Key Size; An Analysis of the WEP Encapsulation," IEEE Document 802.11-00/362.

[10] "Cisco Comments on Recent WLAN Security Paper from University of Maryland," Cisco Systems, Product Bulletin 1327.

[11] Fluhrer S., Mantin L., and Shamir A., "Weaknesses in the Key Scheduling Algorithm of RC4," Eighth Annual Workshop on Selected Areas in Cryptography, August 2001.

[12] Stuart J. Kerry et al, "Response from the IEEE 802.11 Chair on WEP Security," IEEE 802.11 Working Group.
http://www.ieee802.org/11/

[13] "Cisco Aironet Security Solution Provides Dynamic WEP to Address Researchers' Concerns," Cisco Systems, Product Bulletin 1281.

[14] Adam Stubblefield, John Ioannidis, and Aviel Rubin, "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP, Revision 2," AT&T Laboratories Technical Report TD-4ZCPZZ, August 21, 2001.

EDGAR DANIELYAN is a Cisco Certified Network, Design, and Security Professional, as well as member of IEEE, ACM, USENIX, SAGE, and the IEEE Computer Society. Currently self-employed, he consults and writes on internetworking, UNIX, and security. His book, *Solaris 8 Security*, was published by New Riders Publishing in October 2001. The author is not affiliated with any of the organizations (except the IEEE) mentioned in this article. E-mail: **edd@danielyan.com**

Code Signing

by Eric Fleischman, The Boeing Company

Code signing is a common mechanism that authors of executable code use to assert their authorship of that code and to provide integrity assurance to the users of the code that an unauthorized third party has not subsequently modified the code in any way. Code signing is widely used to protect software that is distributed over the Internet. It is also widely used for mobile code security, being a core element of the mobile code security systems of both Microsoft's ActiveX and JavaSoft's Java applet systems. Despite this widespread use, common misunderstandings have arisen concerning the actual security benefits provided by code signing. This article addresses this issue. It explains how code signing works, including its dependence upon underlying *Public Key Infrastructure* (PKI) technologies.

Motivation for Code Signing

Code signing, which is also known as *object signing* in certain programming environments, is a subset of electronic document signing. In many ways code signing is a simplification of the more generic technology in that generally only a single signature is permitted and that signature pertains to the entire file. That is, code signing usually does not support multiple signatures, encryption of (data) content, dynamic data placement, or sectional signing, which are commonly available in many document-signing systems. As a result, code signing provides only authenticity and integrity for *electronic executable files*—it does not provide privacy, authentication, or authorization, which are supported by several electronic document-signing approaches.

A signature provides authenticity by assuring users as to where the code came from—who really signed it. If the certificate originated from a trusted third-party *Certificate Authority* (CA), then the certificate embedded in the digital signature as part of the code-signing process provides the assurance that the CA has certified that the code signer is who he or she claims to be. Integrity occurs by using a signed hash function as evidence that the resulting code has not been tampered with since it was signed.

In the pre-Internet era, software was distributed in a packaged manner via branding or trusted sales outlets. It frequently came in a shrink-wrapped form directly from the vendor or a trusted distributor. In the Internet era, software is often distributed via the Web, by e-mail, or by file transfer. Code signing provides users with a similar level of assurance as to software authenticity in this comparatively anonymous—and comparatively insecure—new distribution paradigm as was previously offered by packaged software in the pre-Internet era.

In all cases, what is assured is the authorship of the software, including the verification that third parties have not subsequently modified the code. In no case does the user receive any assurance that the code itself is safe to run or actually does what it claims. Thus, the actual value of code signing remains a function of the reliability and integrity of its author. Code signing, therefore, is solely a mechanism for software creators to assert their authorship of the product and validate that it has not been modified. In no case does it provide the end user with any claim as to the quality, intent, or safety of the code.

How Code Signing Works

Code signing appends a digital signature to the executable code itself. This digital signature provides enough information to authenticate the signer as well as to ensure that the code has not been subsequently modified.

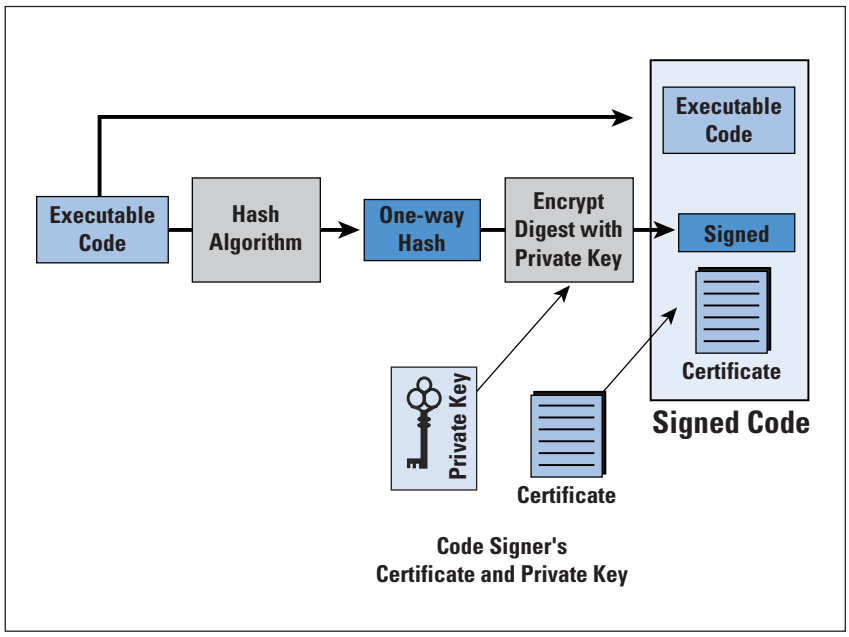
Code signing is an application within a PKI system. A PKI is a distributed infrastructure that supports the distribution and management of public keys and digital certificates. A digital certificate is a signed assertion (via a digital signature) by a trusted third party, known as the Certificate Authority (CA), which correlates a public key to some other piece of information, such as the name of the legitimate holder of the private key associated with that public key. The binding of this information then is used to establish the identity of that individual. All system participants can verify the name-key binding coupling of any presented certificate by merely applying the public key of the CA to verify the CA digital signature. This verification process occurs without involving the CA.

A *public key* refers to the fact that the cryptographic underpinnings of PKI systems rely upon asymmetric ciphers that use two related but different keys, a public key, which is generally known, and a *private key,* which should be known only by the legitimate holder of the public key. This approach is known as *public-key cryptography* and directly contrasts to symmetric ciphers, which contrastingly require the two entities to share an identical secret key in order to encrypt or decrypt information.

The certificates used to sign code can be obtained in two ways: They are either created by the code signers themselves by using one of the code-signing toolkits or obtained from a CA. The signed code itself reveals the certificate origin, clearly indicating which alternative was used. The preference of code-signing systems (and of the users of signed code) is that the certificates come from a CA, and CAs, to earn the fee they charge for issuing certificates, are expected to perform "due diligence" to establish and verify the identity of the individual or institution identified by the certificate. As such, the CA stands behind (validates) the digital certificate, certifying that it was indeed issued only to the individual (or group) identified by the certificate and that the identity of

that individual (or group) has been verified as stated. The CA then digitally signs the certificate in order to formally bind this verified identity with a given private and public key pair, which is logically contained within the certificate itself. This key pair will subsequently be used in the code-signing process. Self-created certificates, by contrast, are unconstrained as to the identities they may impersonate.

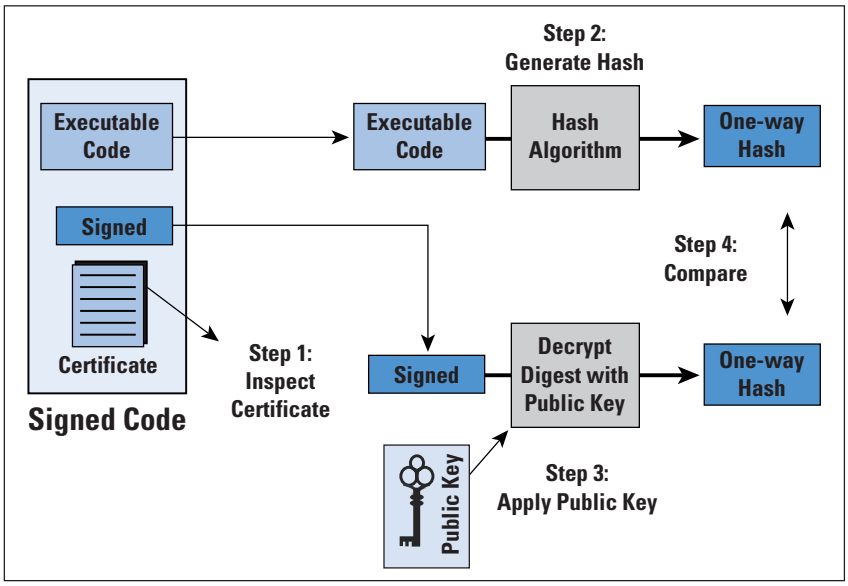
Figure 1: Code-Signing Process



Code signing itself is accomplished as follows: Developers use a *hash function* on their code to compute a *digest*, which is also known as a *one-way hash*. The hash function securely compresses code of arbitrary length into a fixed-length digest result. The most common hash function algorithms used in code signing are the *Secure Hash Algorithm* (SHA), *Message Digest Algorithm 4* (MD4), or MD5. The resulting length of the digest is a function of the hash function algorithm, but a common digest length is 128 bits. The digest is then encrypted using the developer's private key, which is part of the developer's certificate. A package containing the encrypted digest and the developer's Digital Certificate is encapsulated into a special structure called the *signature block*. The signature block is then appended to the executable code to form the signed code.

In a Java context, the signed Java byte code is called a JAR file. First introduced in the *Java Developer's Kit* (JDK) version 1.1, this capability was greatly expanded with Java 2.

Figure 2: Code Verification Process



At some subsequent time, this signed code will be presented to a recipient, usually through the agency of a code-signing verification tool on the recipient's computer. This tool will inspect the signature block to verify the authenticity and integrity of the received code. This inspection is done in the following manner, as shown in Figure 2:

1. The certificate is inspected from the signature block to verify that it is recognizable to the code-signing verification system as a correctly formatted certificate.
2. If it is, the certificate identifies the hash function algorithm that was used to create the signed digest within the received signature block. With this information, the same hash algorithm code that was used to create the original digest is then applied to the received executable code, creating a digest value, which then is temporarily stored. If it is not a correctly formatted certificate, then the code-signing verification process fails.
3. The signed digest value is then taken from the signature block and decrypted with the code signer's public key, revealing the digest value, which was originally computed by the code signer. Failure to successfully decrypt this signed digest value indicates that the code signer's private key was not used to create the received signature. If this is the case, then that signature is a fraud and the code-signing verification process fails.
4. The recomputed digest of Step 2 is then compared to the received digest that was decrypted in Step 3. If these two values are not identical, then the code has subsequently been modified in some way and the code-signing verification process fails. If any such anomaly occurs, then the verification system alerts the recipient concerning the nature of the failure, indicating that the resulting code is suspect and should not be trusted. However, if the digests are identical, then the identity of the code signer is established.

5. If establishment occurs, then the code signer's certificate is copied from the signature block and presented to the recipient. The recipient then has the option to indicate whether or not he or she trusts the code signer. If so, then the code is executed. If not, then it is not executed.

Types of Code Signing

Code signing is a mechanism to sign executable content. The term executable content refers to presenting executable programs in a manner so that they could be run locally—regardless of whether the executable file originated locally or remotely. Code signing is commonly used to identify authorship within several distinct usage scenarios:

- Applications can be code signed to identify their ownership within comparatively anonymous software distribution mechanisms using the Web, the *File Transfer Protocol* (FTP), or e-mail. This type of code signing establishes the origin for downloadable JAR, tar, zip, or CAB file software distributions, for example.
- Code signing can provide Web users more control over mobile code that is available to their Web browsers. Mobile code is code that travels a network in its lifetime in order to execute on a destination machine. The term is usually associated today with active Web content that executes on the client's machine via technologies such as Java, JavaScript, VBScript, ActiveX, and MS Word macros.
- Device drivers can be code signed to inform an operating system of the authorship of that driver. For example, the device drivers for Windows 98, Windows ME, and Windows 2000 operating systems should preferentially be certified by Microsoft's device driver certification laboratory[25]. The entity signs the device driver executable in order to certify that the device driver in question has indeed been successfully demonstrated by a Microsoft certification laboratory to correctly run on that operating system.
- A recent news report[20] has stated that Microsoft will be using code signing as a security mechanism within its forthcoming Windows XP operating system. The article stated: "Microsoft is to incorporate a 'signed application' system in Whistler [that is, Windows XP], the intention being to furnish users with a super-secure mode of operation that just plain stops [unsigned] code executing on the machine."

Code Signing Does Not Provide Total Security

A fundamental problem with code signing is that it cannot provide any guarantee about the good intentions of the signer or the quality, intent, operations, or safety of the code. The VeriSign and Thawte CAs, for example, combat this limitation somewhat for executables signed by certificates they issue by requiring the entities receiving their certificates to sign a "software publisher's pledge" not to sign a piece of malicious software. If they subsequently learn of violations of this agreement, they ask the owner to correct the problem.

If the owner refuses, then they cancel the owner's digital certificate and potentially bring a lawsuit against the offender. The code-signing literature has documented that the latter has occurred at least once[21].

Another problem is that the digital signing by even a reputable entity can be forged if the private key of the signer becomes known. This forging can occur when the criminally minded exploit any of numerous potential vulnerabilities, including hacking into the key store on the signer's machine, carelessness on the part of the signer exposing this information, or an error in a CA PKI key distribution system.

Perhaps the best summary of these issues is provided by Schneier, who wrote:

> "Code signing, as it is currently done, sucks. There are all sorts of problems. First, users have no idea how to decide if a particular signer is trusted or not. Second, just because a component is signed doesn't mean that it is safe. Third, just because two components are individually signed does not mean that using them together is safe; lots of accidental harmful interactions can be exploited. Fourth, "safe" is not an all-or-nothing thing; there are degrees of safety. And fifth, the fact that the evidence of attack (the signature on the code) is stored on the computer under attack is mostly useless: The attacker could delete or modify the signature during the attack, or simply reformat the drive where the signature is stored." (Quoted from page 163 of [17]).

Mobile Code Security

Mobile code security is a two-edged sword: it seeks to protect computer systems receiving potentially hostile mobile code and it also seeks to protect mobile code from potentially hostile users of those computer systems.

Code signing has emerged as a major adjunct to mobile code security. Because mobile code probably represents the dominant use of code signing that occurs today, this section examines how code signing assists mobile code security.

There is substantial and growing literature on mobile code security (for example, see [3] through [16]). The literature identifies four distinct approaches to mobile code security, together with a few hybrids that merge two or more methods. Each of the four approaches has an inherent trust model that identifies the assumptions upon which the approach is based. Rubin and Geer[4] list these four approaches as being:

- The *sandbox approach,* which restricts mobile code to a small set of safe operations. This is the historic approach used by Java applets. In the approach, each Java interpreter implementation attempts to adhere to a security policy, which explicitly describes the restrictions that should be placed on remote applets. "Assuming that the policy

itself is not flawed or inconsistent, then any application that truly implements the policy is said to be secure. ... The biggest problem with the Java sandbox is that any error in any security component can lead to a violation of the security policy. ... Two types of applets cause most of the problems. Attack applets try to exploit software bugs in the client's virtual machine; they have been shown to successfully break the type safety of JDK 1.0 and to cause buffer overflows in HotJava. These are the most dangerous. Malicious applets are designed to monopolize resources, and cause inconvenience rather than actual loss."[4] The trust model assumed by the sandbox approach is that the sandbox is trustworthy in its design and implementation but that mobile code is universally untrustworthy.

- In code signing, the client manages a list of entities that it trusts. When a mobile code executable is received, the client verifies that it was signed by an entity on this list. If so, then it is run; otherwise it does not run. This approach is most commonly associated with Microsoft's ActiveX technology. "Unfortunately, there is a class of attacks that render ActiveX useless. If an intruder can change the policy on a user's machine, usually stored in a user file, the intruder can then enable the acceptance of all ActiveX content. In fact, a legitimate ActiveX program can easily open the door for future illegitimate traffic, because once such a program is run, it has complete access to all of the user's files. Such attacks have been demonstrated in practice."[4] The trust model for this approach assumes that it is possible to distinguish untrustworthy authors from trustworthy ones and that the code from trustworthy authors is dependable.
- The *firewalling approach* involves selectively choosing whether or not to run a program at the very point where it enters the client domain. "Research shows that it may not always be easy to block unwanted applets while allowing other applets ... to run. The firewalling approach assumes that applets can somehow be identified. ... This approach is fundamentally limited, however, by the halting problem, which states that there is no general-purpose algorithm that can determine the behavior of an arbitrary program."[4]

  A related and more viable alternative is the playground architecture that has been used to separate Java classes that prescribe graphics actions from all other actions. The former are loaded on the client, whereas the latter are loaded on a "sacrificial" playground machine for execution and then reporting of the results to the browser. Because this approach requires byte-code modification, it cannot be used in conjunction with the usual approach to code signing.
- The *Proof-Carrying Code* (PCC) technique is a theoretical approach that statistically checks code to ensure that it does not violate safety policies. "PCC is an active area of research so its trust model may change. At present, the design and implementation of the verifier are considered trustworthy but mobile code is universally untrustworthy."[4]

The most common hybrid approach occurs for Java's JDK 1.1 and Java 2. Each combines the sandbox approach, which was the security mechanism for JDK 1.0, with code signing. This hybrid originated from the realization that the inherent restrictions of the sandbox model kept applications from doing "interesting and useful things." Therefore, a mechanism for running applications outside of the sandbox, code sharing, was devised to supplement the sandbox-based original. Specifically, in JDK 1.1 a signed applet enjoys unlimited access to system resources, just like local applications do, provided that the corresponding public key is trusted in the executing environment. This system evolved within Java 2 to optionally provide a consistent and flexible policy for applets and applications, determined by the policies established within a protection domain.

The literature is unanimous that the net result of this hybrid version "introduces the same security problems [as those] inherent in the ActiveX code-signing approach."[4] For this reason, Bernard Cole[11] has stated "neither [the sandbox nor the code signing] model is appropriate to the new environment of small information appliances, connected embedded devices, numerous web-enabled wireless phones and set-top boxes."[11] Indeed, several articles (for example, perhaps the best collection is contained in[13]) contained worrying descriptions of how to compromise specific sandbox and code-signing products.

The literature (see [3] through [16]) is also clear that despite the demonstrable weaknesses of both the sandbox and code-signing approaches as mechanisms for securing mobile code, they are the best practical alternatives available today. In the meantime, researchers are currently exploring enhanced mobile code security by making hybrids containing three—or all four—of the above mechanisms.

Researchers have also begun to investigate alternative techniques. For example, Zhao[16] reports that "Additional innovative authentication functions are needed for mobile code. One approach is to apply digital fingerprinting to authenticate mobile code. Analogous to 'biometric authentication' for access control, a digital fingerprint of mobile code is a unique authentication code that is an integral and intrinsic part of the thing being authenticated. It is placed into the mobile code during its development by using digital watermarking techniques."

Major Code-Signing Systems

Code-signing systems are often functions of specific applications. For example, Thawte[22] is a CA that provides the following certificate types:

- The *Apple Developer Certificate* is used by Apple MacOS-based application developers to sign software for electronic distribution.
- The *JavaSoft Developer Certificate* can be used with JavaSoft's JDK 1.3 and later to sign Web applets.
- A *Marimba Channel Signing Certificate* is used to sign Castanet channels on the Marimba platform.

- A *Microsoft Authenticode Certificate* is used with the Microsoft InetSDK developer tools to sign Web applets (for instance, ActiveX controls) as well as `.CAB`, `.OCX`, `.CLASS`, `.EXE`, `.STL`, and `.DLL` files, and other potentially harmful active content on Microsoft OS platforms. These Authenticode certificates work only with Microsoft IE 4.0 and later browsers.
- *VBA Developer Certificates* are identical to the Microsoft Authenticode certificates. They are used by developers to sign macros in Office 2000 and other VBA 6.0 environments.
- *Netscape Code-Signing Certificate*s are used to sign Java applets, browser plug-ins, and other active content on the Netscape Communicator platform.

Despite this diversity, the clearly dominant code-signing systems today come from Microsoft, Netscape, and JavaSoft. Although these three systems generally adhere to the same set of standards, their approaches are highly diverse from each other. Each has its own certificate type. Each system approaches code signing with different orientations, goals, and expectations.

Interoperability Problems

Although all code signing uses similar technology, interoperability problems currently impact code signing. These problems may originate from interoperability problems within the underlying PKI infrastructure, from certificate differences, or from different (vendor) approaches to code signing itself.

PKI Infrastructure Interoperability

The PKI Forum has identified ten impediments to the widespread adoption of PKI[23], the most significant being the "lack of interoperability" between PKI products. Because of this, the technical working group of the PKI Forum is currently concentrating on addressing PKI interoperability problems: "The Technical Working Group continues its focus on multi-vendor interoperability projects. Over the last six months, it has sponsored monthly interoperability "bake-offs" based on the *Certificate Management Protocol* (CMP) standard, with participation from a growing number of vendors. In addition, two workshops have been held to date on application-level interoperability through the use of digital certificates, with remote testing ongoing. Looking forward, the Technical Working group plans to initiate two new interoperability projects in the areas of Smart Card/Token Portability and CA interoperability, and it will be defining a large-scale, multi-vendor interoperability project for public demonstration in the first quarter of 2001."[24]

Certificate Interoperability

Numerous potential interoperability issues stem from the certificates themselves because certain certificates are themselves tied to specific types of applications.

However, not every certificate is a code-signing certificate. Rather, code-signing certificates are special certificates whose associated private keys are used to create digital signatures. In addition, the id-kp-codesigning value within the extended key usage field of the certificate itself (see Section 4.2.1.13 of RFC 2459) needs to be set to indicate that the certificate can be used for code signing.

In any case, code-signing certificates must be packaged in the appropriate format [*Public Key Cryptographic Standards* (PKCS)]), and the various code-signing approaches (for example, Microsoft, Netscape, JavaSoft) expect both the signing certificates and the code that is to be signed to conform to different file format requirements.

These differences between code-signing systems introduce opportunities for incompatibility, even if each approach otherwise rigorously adheres to the same basic certificate standards.

Not all certificates can be used to support all potential certificate uses, even if they originate from the same CA. For example, the Java Developer Certificates are not interoperable (exchangeable) with any other certificates at this time. Fortunately, it is possible to buy certificates that can be used for many (but not all) potential uses. For example, a single certificate can support Microsoft Authenticode, Microsoft Office 2000/VBA Macro Signing, Netscape Object Signing, Apple Code Signing, and Marimba Channel Signing.

Code Signing System Interoperability

Probably the least understood of the potential interoperability problems are due to different vendor approaches to code signing itself. Perhaps McGraw and Felten have provided the best insight to code-signing system interoperability within Appendix A of their book *Securing Java*[15]. Unfortunately, those insights were in regard to an earlier version of Java, which has evolved considerably since then.

Certificate Issues

Each of the three major code-signing systems (Microsoft, Netscape, JavaSoft) has its own certificates. Each provides its own certificate stores to house certificates within its system.

Each of the three systems supports mechanisms by which certificates may be exported from a given user's certificate store and imported into a different user's certificate store on the same or on a different machine. The Microsoft and Netscape systems also have provisions for importing certificates between code-signing systems.

Certificates are usually exported between PKI systems or certificate stores in the PKCS-12 format (**`.p12`** files if Netscape or **`.pfx`** files if Microsoft Authenticode), which contains both certificate and key pair information within the same file. Certificates can also be exported in the PKCS-7 format (for example, **`.cer`** or **`.spc`** files).

The latter approach lacks information to permit the certificate to be used for code signing by the importing system unless the missing elements can be retrieved via other mechanisms.

Code-Signing Certificates

The Netscape certificate utility (that is, *signtool –L*) indicates which of the certificates located within a certificate store can be used for code signing. By contrast, all certificates (except for those explicitly prohibited from doing code signing according to the provisions of RFC 2459 Section 4.2.1.13) within a Microsoft certificate store can be used for code signing within the Microsoft system. This means that a certificate that is unable to be used for code signing in a Netscape system can be imported into the Microsoft system and be successfully used for code signing there.

This difference stems from RFC 2459 Section 4.2.1.13, which deals with the extended key usage field. The relevant text of the standard is as follows:

> "If the extension is flagged critical, then the certificate MUST be used only for one of the purposes indicated. If the extension is flagged noncritical, then it indicates the intended purpose or purposes of the key, and may be used in the correct key/certificate of an entity that has multiple keys/certificates. It is an advisory field and does not imply that usage of the key is restricted by the certification authority to the purpose indicated. Certificate using applications may nevertheless require that a particular purpose be indicated in order for the certificate to be acceptable to that application."

What has occurred is that Netscape has implemented its system such that certificates can be used only for the purposes specified in the extended usage field. Netscape does this for both critical and noncritical markings. Microsoft, by contrast, provides that restriction solely to certificates that have been marked "critical," permitting certificates without a critical marking to be used for any activity possible. Both approaches are legal, and both fully conform to the standard.

Code Signing from an End User's Perspective

The results obtained when you try to execute signed code is a function of your underlying operating system, the browser you are using, and whether or not the executable is a Java applet. This should not be surprising, because similar differences also occur with unsigned code. For example, a Microsoft executable file will execute on a Microsoft Windows operating system but is unlikely to execute on operating systems that do not recognize that format. Similarly, a Java applet cannot be directly invoked on a Windows operating system, because that operating system does not recognize the `.jar` file extension. However, it will cleanly execute when accessed off of a Web page, regardless of the underlying operating system.

References

[1] "A Closer Look at the E-signatures Law," by Linda Rosencrance, *Computer World,* October 5, 2000.

[2] "Standards Issue Mars E-signature," by Jaikumar Vijayan and Kathleen Ohlson, *Computer World,* July 10, 2000.

[3] "Mobile Code and Security," by Gary McGraw and Edward Felten, *IEEE Internet Computing,* Volume 2, Number 6, November/December 1998.

[4] "Mobile Code Security," by Aviel Rubin and Daniel Geer, *IEEE Internet Computing,* Volume 2, Number 6, November/December 1998.

[5] "Securing Systems Against External Programs," by Brant Hashii, Manoj Lal, Raju Pandey, and Steven Samorodin, *IEEE Internet Computing,* Volume 2, Number 6, November/December 1998.

[6] "Secure Web Scripting," by Vinod Anupam and Alain Mayer, *IEEE Internet Computing,* Volume 2, Number 6. November/December 1998.

[7] "Secure Java Class Loading" by Li Gong, *IEEE Internet Computing,* Volume 2, Number 6, November/December 1998.

[8] "Mobile Code Security: Taking the Trojans out of the Trojan Horse," by Alan Muller, University of Cape Town. April 5, 2000.
http://www.cs.uct.ac.za/courses/CS400W/NIS/papers00/amuller/essay1.htm

[9] "Understanding the keys to Java Security—The Sandbox and Authentication" by Gary McGraw and Edward Felten, *JavaWorld Magazine,* May 1997.

[10] "Repair Program or Trojan Construction Kit?" by Greg Guerin, September 7, 1999.
http://www.amug.org/~glguerin/opinion/crypto-repair-kit.html

[11] "Security, Reliability Twin Concerns in Net Era," by Bernard Cole, *Electrical Engineering Times,* July 24, 2000.

[12] "Java Security: From HotJava to Netscape and Beyond," by Drew Dean, Edward Felten, and Dan Wallach, Proceedings of 1996 IEEE Symposium on Security and Privacy, May 1996.

[13] "Formal Aspects of Mobile Code Security," by Richard Drews Dean, PhD thesis, Princeton University, January 1999.
http://www.cs.princeton.edu/sip/pub/ddean-dissertation.php3

[14] "A Flexible Security Model for Using Internet Content," by Nayeem Islam, Rangachari Anad, Trent Jaeger, and Josyula Rao, IBM Thomas J Watson Research Center, June 28, 1997.
http://www.ibm.com/java/education/flexsecurity/

[15] *Securing Java—Getting Down to Business with Mobile Code,* by Gary McGraw and Edward Felten, ISBN 0-471-31952-X, John Wiley & Sons, 1999.

[16] "Mobile Code: Emerging Cyberthreats and Protection Techniques," by Dr. Jian Zhao, Proceedings of the Workshop on Emerging Threats Assessment—Biological Terrorism, July 7–9, 2000, Dartmouth College, Hanover, NH.

[17] *Secrets and Lies—Digital Security in a Networked World,* by Bruce Schneier, ISBN 0-471-25311-1, John Wiley and Sons, 2000.

[18] Telephone conversation between Bob Moskowitz and Eric Fleischman on September 26, 2000.

[19] E-mail correspondence between Joseph M. Reagle, Jr., of the W3C and Eric Fleischman on December 6, 2000.

[20] **http://www.theregister.co.uk/content/4/14592.html**

[21] **http://www.halcyon.com/mclain/ActiveX/Exploder/FAQ.htm**

[22] **https://www.thawte.com/cgi/server/step1.exe?zone=devel**

[23] **http://pkiforum.org/About/Overview/sld037.htm**

[24] **http://www.pkiforum.org/News/2000/PKI-Forum-third-meeting-20000919.htm**

[25] **http://www.microsoft.com/hwtest/Signatures/**

[A longer version of this article can be obtained from the author.]

ERIC FLEISCHMAN has university degrees from Wheaton College (Illinois), the University of Texas at Arlington, and the University of California at Santa Cruz. He currently works in data communications security. He is employed as an Associate Technical Fellow by The Boeing Company. Eric was formerly employed by the Microsoft Corporation, AT&T Bell Laboratories, Digital Research, and Victor Technologies. He can be contacted at **Eric.Fleischman@boeing.com**

Book Review

Internet Performance Survival Guide

Internet Performance Survival Guide: QoS Strategies for Multiservice Networks, by Geoff Huston, ISBN 0-471-37808-9, John Wiley & Sons, 2000.

Many readers of IPJ are familiar with the name Geoff Huston. He contributes articles frequently. I find his style to be very lucid and his writings to be very well structured and organized.

I have need at my job to begin implementation of *Quality of Service* (QoS) strategies to deal with an ever-increasing demand for *Virtual Private Network* (VPN) tunnels over shared media. So, when I came across the title of this book and saw who wrote it, I jumped at the opportunity to review it for IPJ.

Organization

This book is organized more like a textbook than a reference manual. If you are looking for a quick and dirty guide that simply lists all the tricks of the trade and gives examples of how to implement them on specific equipment, then this book is not for you. If, however, you are looking for a well-written text that will help you to understand the issues, the practices that address them, and the theory that underlies these practices, then this is an excellent book.

The book begins with a chapter that explains in detail the problems that administrators and engineers on heterogeneous, multiprotocol networks face today. There is a quick historical survey of the evolution of networking and how that has shaped the nature of the problem. In a very topical fashion, this introduction covers the basic techniques that can be used to implement QoS, but also explains the complexity involved with these techniques, their limitations, and why they are not widely deployed yet. The book continues from there, starting with a low-level view of the building blocks of the network and gradually building to higher- and higher-level topics.

The second chapter begins with some details about the performance features built into the Internet Protocol, and in particular IPv6. This chapter continues into TCP and covers all the well-known performance features that are built into it, and then moves on to routing, switching, and *Multiprotocol Label Switching,* or MPLS. MPLS is a unified approach to switching across large networks, and it has particular applications to QoS. This topic is one of the main reasons I sought for this book, and I am glad it was covered in such detail. The second chapter ends with a survey of the various transmission systems that are available today, and discusses in detail the performance characteristics and problems that are peculiar to each.

The third chapter is a well-organized exposition of the various types of performance-tuning techniques that are available. The author keeps the discussion at a reasonably abstract level, yet is not afraid to discuss the details of the application of these techniques to the specifics of the network when such details are important. In particular, the use of QoS techniques in conjunction with the *Open Shortest Path First* (OSPF) routing protocol is discussed.

The fourth chapter combines the building blocks of Chapter 2 and the techniques of Chapter 3 into an architectural view that spans the network. The author discusses the metrics that can be used to analyze network performance, the protocols that can be used to implement service strategies, the tradeoffs that are inherent in the problem, and the policy choices that need to be made in order to come up with a clear design. In particular, the Integrated Service and Differentiated Service models are discussed separately, and then the author shows how these can be combined into an end-to-end network design. As with Chapter 3, the author explains important specific cases such as the use of the *Resource Reservation Protocol* (RSVP) with ATM.

The fifth chapter moves on to explain how the architectures that have been described can be used to attack the various kinds of problems that exist on real networks. The emphasis is clearly on the end user of the system and how to measure the levels of service being provided and to bring into play the techniques already discussed to assure a consistent level of service. The organization of this chapter seemed less clear than that of the previous chapters, but that is perhaps due more to the nature of the complexity of the problems being discussed than to the author's limitations or inattention.

The sixth chapter provides little new material, per se, and is more of a perspective on the material already provided. However, it contributes highly to the content of the book in two important ways. First, it provides more of a top-down view of QoS to complement the material in the preceding four chapters, which present a mostly bottom-up view. Secondly, it acts as a natural bookend for the first chapter. The first chapter raises the issues and poses the questions. The middle of the book examines the protocols, techniques, and architectures in detail. The last chapter then attempts to answer the questions that were initially raised.

The author does an excellent job of presenting material that is complex, vast, and is still in the process of evolving in the field. He is very diligent about managing the level of detail, and is careful to first cover the material topically before diving into the details. The examples are appropriate and have been carefully chosen.

One of the features of the material that is most appreciated is the practical perspective that the author brings to his work. The theory never gets out of hand, and is always balanced by a real-life approach to problems that, unfortunately, can never be completely solved. And, the author's observations always seem in tune with the experiences of the reader.

The material is well organized, and readers will appreciate the effort expended on the textual conventions that help to organize and structure the material. The diagrams that accompany the text are clear and well-placed, and they contribute to the reader's comprehension.

A glossary in the back helps a reader who has not thoroughly read the preceding sections of the book. The index is also well done, and the reference material is copious and pertinent.

Recommended

Overall, I would recommend this book to any professional who manages large, integrated networks, particularly those professionals who work for Internet Service Providers in an engineering capacity. I think this reflects the particular interests of the author, but that is as it should be.

—David P. Feldman, Tudor Investment Corporation
David.Feldman@Tudor.com

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

Fragments

ICANN Considers Structural Reform

Stuart Lynn, President and CEO of *The Internet Corporation for Assigned Names and Numbers* (ICANN) recently proposed a sweeping series of structural reforms designed to lead ICANN towards attainment of its core mission. "The current structure of ICANN was widely recognized as an experiment when created three years ago," noted Board Chairman Vint Cerf. "The rapid expansion of and increasing global dependence on the Internet have made it clear that a new structure is essential if ICANN is to fulfill its mission."

ICANN was formed three years ago as an entirely private global organization designed to assume responsibility for the DNS root from the U.S. government and to coordinate technical policy for the Internet's naming and address allocation systems. In the new proposals, the basic mission remains intact, but the means of achieving that mission changes. "What has become clear to me and others is that a purely private organization will not work," said Lynn. "The Internet has become too important to national economic and social progress. Governments, as the representatives of their populations, must participate more directly in ICANN's debates and policymaking functions. We must find the right form of global public-private partnership—one that combines the agility and strength of a private organization with the authority of governments to represent the public interest."

Noting that current organizational inertia and obsession with process over substance has impeded agility, Lynn laid out a roadmap designed to instill confidence in key stakeholders and to ensure that ICANN can be more effective. This roadmap entails restructuring the Board of Directors into a Board of Trustees composed in part of trustees nominated by those governments who participate in the ICANN process; in part by the chairs of proposed new "policy councils" that would replace the existing supporting organizations and that would provide expert advice; and in part by trustees proposed by a broadly-based nominating committee and appointed by the Board itself. The roadmap is designed to bring all critical stakeholders to the table, something that has been difficult to achieve with the present structure and has slowed ICANN's progress and its ability to fulfill its responsibilities. It is also designed to establish a broad-based funding mechanism sufficient to support the critical mission of ICANN.

"We need to build a stronger organization, supported by our key stakeholders, led by the best team that can be assembled, and properly funded," Lynn said. "We must be structured to function effectively in this fast-paced global Internet environment." "A key requirement is to keep the best of the present ICANN," added Cerf, "in ensuring transparency, openness, and participation, while creating an ICANN that can act responsibly and quickly. That will mean rejecting practices that have emphasized process over achievement. Above all, ICANN must be—and be seen to be—effective and supportive of technical innovation and of a reliable Internet."

A paper written by Lynn that explains the reasons for change and the roadmap for reform is posted on the ICANN web site:
http://www.icann.org/general/lynn-reform-proposal-24feb02.htm

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-10/5
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

June 2002 Volume 5, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

The networking industry is full of acronyms, as the table of contents for this issue clearly illustrates. According to the dictionary, an acronym is "...a word formed from the initial letter or letters of each of the successive parts or major parts of a compound term." While neither BEEP nor ENUM are strictly speaking acronyms, these "short names" are becoming ever more prevalent and difficult to keep track of. We promise to continue to provide acronym expansion whenever possible.

BEEP is an example of a technology that came to life in a very short time. While IETF standards often take years from initial idea to protocol specification, BEEP seems to have happened in just over a year. There is already a textbook on BEEP from which our first article is adapted. Marshall Rose gives an overview of the BEEP framework and explains how you can get involved in its further development.

ENUM refers to the use of the *Domain Name System* (DNS) to look up telephone numbers and subsequently route telephone calls to the right destination using the Internet as the underlying routing fabric. This integration of the traditional telephone network with the Internet is becoming a reality and several standardization bodies are working on technologies to make this as seamless as possible. Geoff Huston explains the mechanisms and politics behind ENUM.

Our series "One Byte at a Time" examines the *Dynamic Host Configuration Protocol* (DHCP). This protocol is widely used to provide IP address and other basic routing information to clients. This is particularly useful for mobile devices, but it can be used in any network environment. Since the IP addresses are assigned as leases with a configurable time limit, DHCP also provides for effective address management. Douglas Comer explains the details of DHCP and its predecessor BOOTP.

As always, we appreciate your feedback. Send your comments and questions to **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

An Overview of BEEP

by Marshall Rose, Dover Beach Consulting

The *Blocks Extensible Exchange Protocol* (BEEP) is something like "the missing link between the application layer and the *Transmission Control Protocol* (TCP)."

This statement is a horrific analogy because TCP is a transport *protocol* that provides reliable connections, and it makes no sense to compare a protocol to a layer. TCP is a highly-evolved protocol; many talented engineers have, over the last 20 years, built an impressive theory and practice around TCP. In fact, TCP is so good at what it does that when it came to survival of the fittest, it obliterated the competition. Even today, any serious talk about the transport protocol revolves around minor tweaks to TCP. (Or, if you prefer, the intersection between people talking about doing an "entirely new" transport protocol and people who are clueful is the empty set.)

Unfortunately, most application protocol design has not enjoyed as excellent a history as TCP. Engineers design protocols the way monkeys try to get to the moon—that is, by climbing a tree, looking around, and finding another tree to climb. Perhaps this is because there are more distractions at the application layer. For example, as far as TCP is concerned, its sole reason for being is to provide a full-duplex octet-aligned pipe in a robust and network-friendly fashion. The natural result is that while TCP's philosophy is built around "reliability through retransmission," there isn't a common mantra at the application layer.

Historically, when different engineers work on application protocols, they come up with different solutions to common problems. Sometimes the solutions reflect differing perspectives on inevitable tradeoffs; sometimes the solutions reflect different skill and experience levels. Regardless, the result is that the wheel is continuously reinvented, but rarely improved.

So, what is BEEP and how does it relate to all this? BEEP integrates the best practices for common, basic mechanisms that are needed when designing an application protocol over TCP. For example, it handles things like peer-to-peer, client/server, and server/client interactions. Depending on how you count, there are about a dozen or so issues that arise time and time again, and BEEP just deals with them. This means that you get to focus on the "interesting stuff."

BEEP has three things going for it:

- It's been standardized by the *Internet Engineering Task Force* (IETF), the so-called "governing body" for Internet protocols.
- There are open source implementations available in different languages.
- There's a community of developers who are clueful.

The standardization part is important, because BEEP has undergone a lot of technical review. The implementation part is important, because BEEP is probably available on a platform you're familiar with. The community part is important, because BEEP has a lot of resources available for you.

Application Protocols

An application protocol is a set of rules that says how your application talks to the network. Over the last few years, the *Hypertext Transfer Protocol* (HTTP) has been pressed into service as a general-purpose application protocol for many different kinds of applications, ranging from the *Internet Printing Protocol* (IPP)[1] to the *Simple Object Access Protocol* (SOAP)[2]. This is great for application designers: it saves them the trouble of having to design a new protocol and allows them to reuse a lot of ideas and code.

HTTP has become the reuse platform of choice, largely because:

- It is familiar.
- It is ubiquitous.
- It has a simple request/response model.
- It usually works through firewalls.

These are all good reasons, and—if HTTP meets your communications requirements—you should use it. The problem is that the widespread availability of HTTP has become an excuse for not bothering to understand what the requirements really are. It's easier to use HTTP, even if it's not a good fit, than to understand your requirements and design a protocol that does what you really need.

That's where BEEP comes in. It's a toolkit that you can use for building application protocols. It works well in a wide range of application domains, many of which weren't of interest when HTTP was being designed.

BEEP's goal is simple: you, the protocol designer, focus on the protocol details for your problem domain, and BEEP takes care of the other details. It turns out that the vast majority of application protocols have more similarities than differences. The similarities primarily deal with "administrative overhead"—things you need for a working system, but aren't specific to the problem at hand. BEEP mechanizes the similar parts, and lets you focus on the interesting stuff.

Application Protocol Design

Let's assume, for the moment, that you don't see a good fit between the protocol functions you need and either the e-mail or the Web infrastructures. (We'll talk more about this later on in the section "The Problem Space".) It's time to make something new.

First, you decide that your protocol needs ordered, reliable delivery. This is a common requirement for most application protocols, including HTTP and the *Simple Mail Transfer Protocol* (SMTP).[3] The easiest way to get this is to layer the protocol over TCP.

So, you decide to use TCP as the underlying transport for your protocol. Of course, TCP sends data as an octet stream—there aren't any delimiters that TCP uses to indicate where one of your application's messages ends and another one begins. This means you have to design a framing mechanism that your application uses with TCP. That's pretty simple to do—HTTP uses an octet count and SMTP uses a delimiter with quoting.

Since TCP is just sending bytes for you, you need to not only frame messages, but have a way of marking what's in each message. (For example, a data structure, an image, some text, and so on.) This means you have to design an encoding mechanism that your application uses with the framing mechanism. That's also pretty simple to do—HTTP and SMTP both use *Multipurpose Internet Mail Extensions* (MIME).[4]

Back in the early 1980s, when I was a young (but exceptionally cynical) computer scientist, my advisor told me that protocols have two parts: *data* and *control*. It looks like the data part is taken care of with MIME, so it's onto the control part. If you are fortunate enough to know ahead of time every operation and option that your protocol will ever support, there's no need for any kind of capabilities negotiation. In other words, your protocol doesn't need anything that lets the participants tell each other which operations and options are supported. (Of course, if this is the case, you have total recall of future events, and really ought to be making the big money in another, more speculative, field.)

The purpose of negotiation is to find common ground between two different implementations of a protocol (or two different versions of the same implementation). There are lots of different ways of doing this and, unfortunately, most of them don't work very well. SMTP is a really long-lived, well-deployed protocol, and it seems to do a pretty good job of negotiations. The basic idea is for the server to tell the client what capabilities it supports when a connection is established, and then for the client to use a subset of that.

Well, that's just the first control issue. The next deals with when it's time for the connection to be released. Sometimes this is initiated by the protocol, and sometimes it's required by TCP because the network is unresponsive. To further complicate things, if the release is initiated by the protocol, maybe one of the computers hasn't finished working on something, so it doesn't want to release the connection just yet.

Some application protocols don't do any negotiation on connection release, and just rely on TCP to indicate that it's time to go away—even though this is inherently ambiguous. Is ambiguity a good thing in a protocol? Computers lack subtlety and nuance, so in protocols between computers, ambiguity is a bad thing. For example, in HTTP 1.0 (and earlier), you often didn't know whether a response was truncated or not. For a more concrete example, interested readers will be amused by page 2 of RFC 962.[5]

The final control issue deals with what happens between connection establishment and release. Most application protocols tend to be client/server in nature: one computer establishes a connection, sends some requests, gets back responses, and then releases the connection. But, are the requests and responses handled one at a time (in lock-step), or can multiple requests be outstanding, either in transit or being processed, at the same time (asynchronously)?

In the original SMTP, the lock-step model was implicitly assumed by most implementors; later on, SMTP introduced a capability to allow limited pipelining. Regardless, as soon as we move away from lock-stepping, it looks as though we'll need some way of correlating requests and responses.

Although this is a step in the right direction, some application protocols need even more support for asynchrony. The reasoning is a little convoluted, but it all comes down to performance. There's a lot of overhead involved in terms of establishing a connection and getting the right user state, so it makes sense to maximize the number of transactions that get done in a single connection. While this helps in terms of overall efficiency, if the transactions are handled serially, then transactional latency—the time it takes to transit the network, process the transaction, and then transit back—isn't reduced (and may even be increased); a transaction might be blocked while waiting for another to complete. The solution is to be able to handle transactions in parallel.

Earlier I mentioned how, back in the 1980s, protocols had two parts, *data* and *control*. Today, things have changed. First of all, I'm still cynical, but more comfortable with it, and—perhaps as important—many might argue that protocols now have a third part, namely *security*.

The really unfortunate part is that security is a moving target on two fronts:

- When you deploy your protocol in different environments, you may have different security requirements.
- Even in the same environment, security requirements change over time.

This introduces something of a paradox: modern thinking is that security must be tightly integrated with your protocol, but at the same time, you have to take a modular approach to the actual technology to allow for easy upgrades. Worse, it's very easy to get security very wrong. (Just ask any major computer vendor!) Few applications folks are also expert in protocol security, and obtaining that expertise is a time-consuming, thankless task, so there's a lot of benefit in having a security mechanism menu, developed by security experts, that applications folk can pick from.

Now the good news: there's already something around designed to meet just those requirements. It's called the *Simple Authentication and Security Layer* (SASL), and a lot of existing application protocols have been retrofitted over the last four years to make use of it.

Well, let's see what all this means. Without ever having talked about what your application protocol is going to do to earn a living, we have to develop solutions for:

- Framing messages
- Encoding data
- Negotiating capabilities (versions and options)
- Negotiating connection release
- Correlating requests and responses
- Handling multiple outstanding requests (pipelining)
- Handling multiple asynchronous requests (multiplexing)
- Providing integrated and modular security
- Integrating all these things together into a single, coherent framework

So, going back to the question "Why use BEEP?", the answer is pretty simple: if you use BEEP, you simply don't have to think about any of these things. They automatically get taken care of.

Now maybe you're the kind of hardcore engineer that really wants to solve these problems yourself. Okay, go right ahead! But first, I'll let you in on a little secret: engineers have been solving these problems since 1972. In fact, they keep solving them over and over again. For each problem, there are usually two or three good solutions, and while individual tastes may vary, the sad fact is that you can make any of them work great if you're willing to put in the hours. But why put in the hours if they have nothing to do with the primary reason for writing the application protocol to begin with? Isn't there something more productive that you'd care to do with your life than design yet another framing protocol?

So, what's really *new* about BEEP? The short answer is: not much. The innovative part is that some folks sat down, did an analysis of the problems and solutions, and came up with an integrated framework that put it all together. That's not really innovation, but it's really good news if you're already familiar with the building blocks that BEEP uses.

Doesn't all this stuff add a lot of overhead? The short answer is: nope. The reason is a little more complex. BEEP is fairly minimalistic—it provides a simple mechanism for negotiating things on an à la carte basis. If you don't want privacy, no problem; don't turn it on. If you don't want parallelism, that's easy; just say "no" if the other computer asks for it. The trick here is two-fold:

- BEEP's inner mechanisms (for example, framing) are pretty lightweight, so you don't incur a lot of overhead using them (even if you don't use all the functionality they provide).
- BEEP's outer mechanisms (for example, encryption) are all controlled via bilateral negotiation, so you can decide exactly what you want to get and pay for.

There's no free lunch, but if you want to start with something "lean and mean," BEEP doesn't slow you down, and when you want to bulk up (say, by adding privacy), BEEP lets you negotiate it. You incur only the overhead you need. (This overhead *will* show up, regardless of whether you use BEEP or grow your own mechanisms.)

It turns out that this philosophy can yield some interesting results. For example, take a look at this high-level scripting fragment:

```
::init -server example.com -port 10288 -privacy strong
```

This fragment is invoking a procedure to establish a BEEP session. With the exception of the last two terms, it looks pretty conventional.

The last two terms tell the procedure to "tune" the session by looking at the security protocols supported in common, selecting one that supports "strong privacy," and then negotiating its use. What's interesting here is that neither the person who designed the application protocol nor the person who wrote the application making the procedure call has to be a security expert. The choice to use strong privacy, and how it gets transparently used, is all an issue of provisioning. Of course, the application protocol designer may still provide security guidelines to the implementor; naturally, the implementor may bundle a wide range of security protocols with the code. However—and this is key—everyone got to focus on what they do best (even the security guys), and it still comes together into a working system.

The cool part here is how easily this all integrates into an evolving protocol. Back in the good ol' days (say the mid-1980s) when the *Post Office Protocol* (POP)[6] was defined, this kind of flexibility wasn't available. Whenever someone wanted to add a new security mecha-

nism for authentication or privacy, you had to muck with the entire protocol. With BEEP's framework, you just add a module that works seamlessly with the rest of the protocol. This means less work for everyone, and presumably fewer mistakes getting the work done.

Now we've come full circle: the reason for using BEEP is because it makes it a lot easier to specify, develop, maintain, and evolve new application protocols.

The Problem Space

BEEP works for a large class of application protocols. However, you should always use the right tool for the right job. Before you start using BEEP for a project, you should ask yourself whether your application protocol is a good fit for either the e-mail or Web models.

Dave Crocker, one of the Internet's progenitors, suggests that network applications can be broadly distinguished by five operational characteristics:

- Server push or client pull
- Synchronous (interactive) or asynchronous (batch)
- Time-assured or time-insensitive
- Best-effort or reliable
- Stateful or stateless

For example:

- The World Wide Web is a pull, synchronous, time-insensitive, reliable, stateless service.
- Internet mail is a push, asynchronous, time-insensitive, best-effort, stateless service.

This is a pretty useful taxonomy.

So, your first step is to see whether either of these existing infrastructures meet your requirements. It's easiest to start by asking if your application can reside on top of e-mail. Typically, the unpredictable latency of the Internet mail infrastructure raises the largest issues; however, in some cases it's a non-issue. For example, in the early 1990s, some of the earliest business-to-business exchanges were operated over e-mail (for example, USC/ISI's FAST project). If you can find a good fit between your application and Internet e-mail, use it!

More likely, though, you'll be tempted to use the Web infrastructure, and there are a lot of awfully good reasons to do so. After all, when you use HTTP:

- There's lots of tools (libraries, servers, etc.) to choose from.
- It's easy to prototype stuff.
- There's already a security model.
- You can traverse firewalls pretty easily.

All of this boils down to one simple fact: it is pretty easy to deploy things in the Web infrastructure. The real issue is whether you can make good use of this infrastructure.

HTTP was originally developed for retrieving documents in a LAN environment, so HTTP's interaction model is optimized for that application. Accordingly, in HTTP:

- Each session consists of a single request/response exchange.
- The computer that initiates the session is also the one that initiates the request.

What needs to be emphasized here is that this is a perfectly fine interaction model for HTTP's target application, as well as many other application domains.

The problem arises when the behavior of your application protocol doesn't match this interaction model. In this case, there are two choices: make use of HTTP's extensibility features, or simply make do. Obviously, each choice has some drawbacks. The problem with using HTTP's extensibility features is that it pretty much negates the ability to use the existing HTTP infrastructure; the problem with "just making do" is that you end up crippling your protocol. For example, if your application protocol needs asynchronous notifications, you're out of luck.

A second problem arises due to "the law of codepaths." The HTTP 1.1 specification, RFC 2616[10] is fairly rigorous. Even so, few implementors take the time to think out many of the nuances of the protocol. For example, the typical HTTP transaction consists of a small request, which results in a (much) larger response. Talk to any engineer who's worked on a browser and they'll tell you this is "obvious." So, what happens when the "obvious" doesn't happen?

Some time ago, folks wanted a standardized protocol for talking to networked printers. The result was something called the *Internet Printing Protocol* (IPP)[1]. IPP sits on top of HTTP. At this point, the old "obvious" thing (small request, big response) gets replaced with the new "obvious" thing—the request contains an arbitrarily large file to be printed, and the response contains this tiny little status indication. A surprising amount of HTTP software doesn't handle this situation particularly gracefully (that is, long requests get silently truncated). The moral is that even though HTTP's interaction model doesn't play favorites with respect to lengthy requests or responses, many HTTP implementors inadvertently make unfortunate assumptions.

A third problem deals with the unitary relationship between sessions and exchanges. If a single transaction needs to consist of more than one exchange, it has to be spread out over multiple sessions. This introduces two issues:

- In terms of stateful behavior, the server computer has to be able to keep track of session state across multiple connections, imposing a significant burden both on the correctness and implementation of the protocol (for example, to properly handle time-outs).
- In terms of performance, TCP isn't designed for dealing with back-to-back connections—there's a fair amount of overhead and latency involved in establishing a connection. This is also true for the security protocols that layer on top of TCP.

HTTP 1.1 begins to address these issues by introducing persistent connections that allow multiple exchanges to occur serially over a single connection, but still the protocol lacks a session concept. In practice, implementors try to bridge this gap by using "cookies" to manage session state, which introduces ad-hoc (in)security models that often result in security breakdowns (as a certain Web-based e-mail service provider found out).

This brings us to a more general fourth problem: although HTTP has a security model, it predates SASL. From a practical perspective, what this means is that it's very difficult to add new security protocols to HTTP. Of course, that may not be an issue for you.

If you can find a good fit between your application and the Web infrastructure, use it! (For those interested in a more architectural perspective on the reuse of the Web infrastructure for new application protocols, consider RFC 3205[7].)

Okay, so we've talked about both the e-mail and Web infrastructures, and we've talked about what properties your application protocol needs to have in order to work well with them. So, if there isn't a good fit between either of them and your application protocol, what about BEEP?

BEEP's interaction model is pretty simple, with the following three properties:

- Each session consists of one or more request/response exchanges.
- Either computer can initiate requests or notifications.
- It's connection-oriented.

By using BEEP, you get an amortization effect with respect to the cost of connection establishment and state management. This is largely derived from the first property. Similarly, the second property gives BEEP its ability to support either peer-to-peer or client-server interactions. What we really need to explain is the connection-oriented part.

To begin, all three of the interaction models we've looked at (BEEP, e-mail, and the Web) are connection-oriented. (Although e-mail may get delivered out of order, the commands sent over each e-mail "hop" are processed in an ordered, reliable fashion.) The connection-oriented model is the most commonly used for application protocols, but it does introduce some restrictions.

A connection-oriented interaction model means that data is delivered reliably and in the same order as it was sent. If you don't require ordered, reliable delivery, you don't need a connection-oriented interaction model. For example, Internet telephony applications don't fit this model, nor do traditional multicast applications.

So, BEEP is suitable for unicast application protocols (two computers are talking to each other). However, not all unicast applications need a connection-oriented model—for example, the *Domain Name System* (DNS) manages name-to-address resolutions just fine without it. In fact, if your protocol is able to limit each session to exactly one request/response exchange with minimalist reliability requirements, and also limit the size of each message to around 65K octets, then it's probably a good candidate for using the *User Datagram Protocol* (UDP) instead.

The IETF and BEEP

BEEP is an emerging standard from the *Internet Engineering Task Force* (IETF). The IETF is a voluntary professional organization that develops many of the protocols running in the Internet. (Of course, anyone is free to develop their own protocols to run in their own little part of the Internet, but if you want multi-vendor support, you need an organization like the IETF.) So why does the IETF care about BEEP?

The answer is that the largest area in the IETF deals with application protocols. There are usually over two dozen working groups developing different application protocols. And, the IETF has been doing this for a long, long time. It turns out that even though there are well-engineered solutions to the different overhead issues, BEEP is the first time that the IETF decided to develop a standard approach that integrates the best practices for each issue. Before BEEP, each working group would spend endless hours arguing about different solutions, and then, if any time was remaining, they might sit down and look at the actual problem domain. (Okay, this is an exaggeration... but not by much!)

So, here's the process by which BEEP got designed:

- Identify the common domain-independent problems.
- Determine the best solution for each problem.
- Integrate the solutions into a consistent framework.
- Declare victory.

Now, the obvious question is: how do you determine what's "best?"

The truth is that in some cases, the answer is obvious, and in other cases, the answer is arbitrary. (Protocol experts hate to admit this, but in some cases, there is no clear winner, and it's simply better to pick *one* and order another drink.) Since most of what BEEP does is hidden from the application designer and implementor, there's really not a lot of mileage in going through it here.

beepcore.org

Where can you find out more about BEEP? To start, you can always consult the two RFCs: the BEEP core framework[8] and the BEEP's mapping onto TCP[9]. However, it's probably better to start with the BEEP community Web site **`http://beepcore.org`** where you'll find:

- News about BEEP meetings and events
- Information about BEEP projects, programmers, and consultants
- Information about beepcore (open source) and commercial software
- BEEP-related RFCs, Internet-Drafts, and whitepapers

[This article is adapted from *Beep—The Definitive Guide,* by Marshall T. Rose, ISBN 0-596-00244-0, O'Reilly & Associates, 2002. Used with permission. **`http://www.oreilly.com/catalog/beep/`**]

References

[1] Herriot, R., Ed., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.0: Encoding and Transport," RFC 2565, April 1999.

[2] **`http://www.w3.org/TR/SOAP/`**

[3] Postel, J., "Simple Mail Transfer Protocol," RFC 821, August 1982.

[4] Freed, N., Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies," RFC 2045, November 1996.

[5] Padlipsky, M. A., "TCP-4 prime," RFC 962, November 1985.

[6] Rose, M. T., "Post Office Protocol: Version 3," RFC 1081, November 1988.

[7] Moore, K., "On the use of HTTP as a Substrate," RFC 3205, February 2002.

[8] Rose, M., "The Blocks Extensible Exchange Protocol Core," RFC 3080, March 2001.

[9] Rose, M., "Mapping the BEEP Core onto TCP," RFC 3081, March 2001.

[10] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., "Hypertext Transfer Protocol — HTTP/1.1," RFC 2616, June 1999.

MARSHALL T. ROSE is the prime mover of the BEEP Protocol. In his former position as the Internet Engineering Task Force (IETF) area director for network management, he was one of a dozen individuals who oversaw the Internet's standardization process. Rose was responsible for the design, specification, and implementation of several Internet-standard technologies, and wrote more than 60 of the Internet's Requests For Comments (RFCs). With a Ph.D. in information and computer science from the University of California, Irvine, Rose is the author of several professional texts.
E-mail: **`mrose@dbc.mtview.ca.us`**

ENUM—Mapping the E.164 Number Space into the DNS

by Geoff Huston, Telstra

Many communications networks are constructed for a single form of communication, and are ill suited to being used for any other form. Although the Internet is also a specialized network in terms of supporting digital communications, its relatively unique flexibility lies in its ability to digitally encode a very diverse set of communications formats, and then support their interaction over the Internet. In this way many communications networks can be mapped into an Internet application and in so doing become just another distributed application overlayed on the Internet. From this admittedly Internet-centric perspective, voice is just another Internet application. And for the growing population of *Voice over IP* (VoIP) users, this is indeed the case. Being able to transmit voice over the Internet is not enough. Allowing one Internet handset to connect to any other Internet handset is still not enough. In the same way that walkie-talkies became ubiquitous mobile phones only when there was a seamless integration with the telephone network, a truly useful VoIP approach will be one that supports seamless integration with the telephone network.

The basics of the telephony world are very simple indeed. Telephone handsets are little more than a speaker and a microphone. When a call is made, the network connects the microphone of one party to the speaker of the other, and vice versa. Of course you don't need a specialized telephone network to support the carriage of voice. As any user of a desktop computer would confirm, there are now a plethora of applications that can deliver a voice signal across the network. For an application to support a voice conversation, a conventional approach is to use a network base of the *User Datagram Protocol* (UDP) transport protocol, with a *Real-Time Protocol* (RTP) overlay, and the RTP payload is an encoded version of the original analogue voice signal. Carrying voice signals in real time across an Internet is a well-understood network service, with an accompanying set of existing protocols and associated applications.

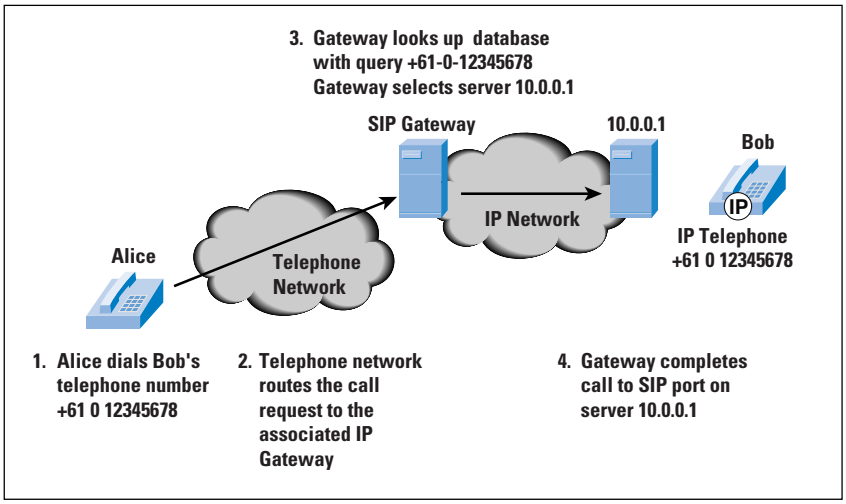
E.164 Addresses and IP Services

However, being able to transmit voice signals across a network is not enough. It was Strowger's step-by-step switching system of the late 19th century that transformed the telephone into a truly useful communications network, allowing any telephone subscriber to initiate a conversation with any other subscriber. This has evolved today into a global numbering plan where every device connected to the telephone network is assigned a unique numerical address. This numbering plan is administered by the *International Telecommunication Union* (ITU), and the plan, Recommendation E.164, involves the assignment of number prefixes to each country code administrator[1].

If the Internet voice domain interoperates seamlessly with the telephone network, supporting this E.164 numbering plain into the realm of the Internet is a critical step. To make Internet telephony truly useful, the Internet telephony world has to be able to interface to the telephone network by allowing Internet-connected telephone devices to make and receive calls to any other telephone device, whether the other device is connected to the Internet, connected to the telephone network, or connected to any other network that seamlessly interoperates with the telephone network. For this to work, one of the preconditions is that every Internet device that supports telephone operation needs to also have an alias in the form of a unique telephone address. But there's a bit more to it than simple numbering.

Each Internet telephone is also an IP device, and, for the Internet component of the end-to-end path, the voice traffic will be carried by IP packets. These packets obviously require the IP address of the Internet telephone device. So each Internet telephone requires both an Internet address and a telephone address. It is the mapping from a telephone number to an IP address that is the crucial part of this function.

Figure 1: Calling an IP Telephone



Consider an example. When Alice, on a normal telephone, wants to call Bob, on an Internet phone, all Alice needs to do is simply dial Bob's telephone number, or his E.164 address (Figure 1). Of course, because Bob's phone is connected to the Internet and can't directly receive Alice's call request, a gateway is necessary. The telephone system should be able to map Alice's call request to the Internet telephony gateway that is configured to act as Bob's gateway agent. The gateway then needs to translate Bob's E.164 phone number into an IP address. Then the gateway has to map the telephone network signals associated with Alice's call request to corresponding signals within an Internet session initiation protocol, and then send these IP packets to Bob's Internet phone. If Bob answers the call, the phone uses the same protocol to inform the gateway, which then sends a corresponding telephone call code across the telephone network to Alice.

When Bob accepts the call, the gateway can then pass all data originating from Alice to Bob's IP address, and all data received from Bob's IP address across to the telephone connection to Alice for the duration of the call. Alice never needs to know that Bob is using an Internet device. Alice dialed a phone number, heard it ring, and then heard Bob answer the call. For Alice, nothing has changed. Bob heard the phone ring, picked it up, and talked to Alice. For Bob, nothing has changed.

The simplest way to configure each gateway is to load each gateway with a configured list of E.164 phone numbers and corresponding IP addresses. This approach is currently very common, but, like all statically configured approaches, has its weaknesses. But what happens when the IP device is numbered dynamically using the *Dynamic Host Configuration Protocol* (DHCP), or if it's mobile, and moves from one service provider's IP network to another, or when the end subscriber changes providers and that subscriber's network is renumbered, or when the primary gateway fails and the providers want to switch to a secondary device? In other words, how can this mapping be dynamic rather than static?

The way a dynamic domain name-to-IP address mapping can be maintained on the Internet is through the Internet *Domain Name System* (DNS). The telephony gateway can use the the E.164 address as the DNS query, and request the DNS to return the corresponding IP address. In our example, when Alice rings Bob, the gateway can use the DNS to obtain Bob's current IP address. The gateway can then use the *Session Initiation Protocol* (SIP) to send to Bob's Internet phone a call request, which then starts Bob's phone ringing. If Bob changes IP address, then the corresponding change is a change in the DNS, not in the gateway itself. If the primary gateway fails and a secondary gateway is used, the secondary system can already access all necessary mappings through the DNS.

So the general approach of using the DNS to contain this mapping is one with some merit, but, as always, the devil is in the details. There are two parts to mapping a E.164 number into the DNS. The first is the nature of the transforms to be applied to the E.164 address to obtain a DNS query string, and the second is the form of the DNS response to this query.

Mapping E.164 Addresses into DNS Query Strings

One possible approach to mapping an E.164 number into the DNS is to simply place numbers as text blocks into the DNS. In this way, the number *+61-0-12345678* could be mapped to the DNS string **`61012345678.example.com.`** If this method were to be used for a sizable number of E.164 numbers, there are obvious DNS performance implications associated with the size of this DNS zone file, together with the issue of frequency of update of the zone and its cache characteristics.

There are also a large number of E.164 country code delegated authorities and, consequently, a large number of entities who would like to be the authority for parts of such a monolithic unstructured DNS zone file.

In order to avoid these issues, some structure in the E.164 address space has to be used to map into the hierarchical name structure used in the DNS. One helpful observation is that E.164 numbers and Internet domain names use opposite ordering. Whereas a fully qualified domain name, such as **`test.example.com,`** has the more specific parts to the left and the most general part, the root, on the right of the name, a telephone number code has the most general part, the reference to the country code prefix "**`+`**" to the left and the more specific parts to the right. If one were to reverse the order of E.164 symbols, then the two address domains would have a similar structure.

One of the first efforts to provide a mapping between E.164 number and the DNS was part of the TPC fax gateway service, started in 1993[2]. This approach uses a reversed E.164 number, and treats every digit as a node on the DNS name hierarchy. In our example, the E.164 address *+61 0 12345678* would map to the DNS query string **`8.7.6.5.4.3.2.1.0.1.6.tpc.int.`** (in the TPC service, the parent DNS zone of this mapping is **`tpc.int.`**)

This mapping has some very convenient properties. Each country code corresponds to a delegatable DNS domain, so that the international country code for Australia, *+61,* can have a corresponding DNS delegation for the zone **`1.6.tpc.int.`** Within the country code the DNS can be further delegated to operators in a manner that parallels the further delegation of E.164 common prefix number blocks.

This same mapping is used by ENUM, using a DNS name parent of **`e164.arpa.`** The mapping entails taking a complete E.164 address (including the country code), and then removing all nondigit symbols from the address. The digit string is reversed and a "**`.`**" is placed between each pair of digits. The string **`.e164.arpa.`** is then appended to make a complete DNS query string. Using this process, our example number *+61-0-12345678* is transformed into the DNS query:
`8.7.6.5.4.3.2.1.0.1.6.e164.arpa.`

Although this form of mapping is technically well suited to the DNS, it does mean that the DNS equivalent of the E.164 address is not very easily adapted to our conventional use of telephone numbers. The implication is that it is likely that Internet-based telephony applications will continue to present E.164 numbers in their user interfaces as conventional telephone numbers, and manipulate the DNS equivalent strings as internal objects.

The DNS Response

The telephone network supports more than simple voice conversations, and any serious attempt to bridge the telephone network and the Internet also should be able to also handle various forms of text messaging and paging services as well as document transmission undertaken as faxes. The desired outcome is that the interface between the telephone network and the Internet should be able to seamlessly redirect the telephone service to the appropriate Internet service. In other words, we are seeing a requirement that a set of services associated with the same E.164 address should be able to be mapped to a set of IP servers, rather than a single server with a single IP address.

The implication is that the DNS response to an ENUM query should have a richer functionality than simply returning a single IP address. In DNS terms, associating a conventional "A" DNS resource record with each ENUM domain name is not sufficiently flexible for our purposes.

The approach adopted by the TPC fax gateway service was to map a fax in the telephone environment to an e-mailed multimedia message in the Internet environment. To support this mapping, telephone numbers were mapped to DNS *Mail Exchange* (MX) resource records, and these records were mapped to a mail server's IP address in a second DNS lookup.

ENUM attempts to solve a more general model of providing mappings for any relevant service. One possible approach is to use a collection of DNS name roots, one for each mappable service. Thus, for example, **`fax.e164.arpa.`** could hold mappings for the fax service, while **`voice.e164.arpa.`** could hold mappings for voice services, and so on. However, this approach is not consistent with the generic architecture of the DNS, and the distribution of service information has the potential to lead to synchronization errors. Usefully, the DNS allows a collection of resource records to be associated with a DNS name, and this set of records is returned as the answer to a query. It is then left to the application to determine which particular record to use, with perhaps some preference hints provided in the DNS response. The approach used by ENUM takes advantage of this DNS capability, and ENUM uses the DNS to map an **`e164.arpa`** number onto a collection of service-specific *Uniform Resource Identifiers* (URIs)[3].

A gateway that uses ENUM to query the DNS will receive the complete collection of service-specific URIs in response to a request to translate an E.164 address to a URI. Depending on the type of service being requested, the gateway can then select the most appropriate URI and use the DNS a second time to translate the domain name part of the URI to an IP address using the URI-specific DNS resource record as a query term. The gateway can then use the full URI specification to open an IP session with the selected service port and complete the service transaction.

The URI resource records used by ENUM are *Naming Authority Pointers* (NAPTR) records[4]. This form of use of the DNS allows for entries where the entry itself can be decomposed into further delegations, using name formats that use URI syntax[5].

NAPTR fields contain numerous components:

- An *Order* field to specify the order in which multiple NAPTR records must be processed
- A *Preference* field to determine the processing order when multiple NAPTR records have the same order value
- A *Service* field to specify the resolution protocol and service
- *Flags* to modify the actions of further DNS lookups
- A *regular expression* to allow the query client to rephrase the original request in a DNS format
- A *Replacement* field to define the next DNS query object

The intended operation of ENUM is to first take the E.164 number and convert it to a query in the **`e164.arpa`** domain. The resultant set of services is specified by the returned collection of NAPTR records. The agent selects a service that matches the service characteristics of the original request, and takes the corresponding URI for further resolution by the DNS. The elements of this URI are further decomposed as per any rewrite rules in the NAPTR record. DNS queries are generated as per the sequence of preferred NAPTR rewrite operations. The ultimate result of this sequence of DNS queries is the specification of a protocol, an associated port address, and the IP address for a preferred server for the service.

An Example of the Use of ENUM

Let's say Bob's Internet telephone services are mapped to the E.164 address *+61-0-12345678*. When Alice tries to call Bob, the telephone network routes the call request toward the Internet gateway that is the nominated service agent for this E.164 number. The Internet gateway takes the call setup request with Bob's number and first reverses the digits, then inserts a "**`.`**" between each digit, and finally appends **`e164.arpa.`** The resultant DNS string is the fully qualified domain name **`8.7.6.5.4.3.2.1.0.1.6.e164.arpa.`** This name is then passed as a query to the DNS, to retrieve all associated NAPTR DNS resource records.

Bob has specified that he prefers to receive calls using SIP addressed to user **`bob`** at the server **`telebob.au`** by placing the following in the DNS:

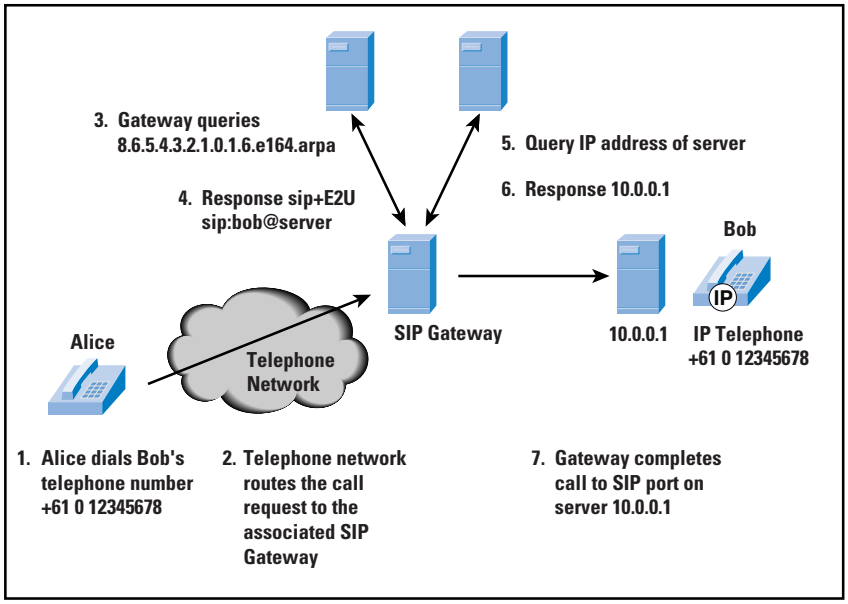
```
$ORIGIN 8.7.6.5.4.3.2.1.0.1.6.e164.arpa.
   IN NAPTR 100 10 "u" "sip+E2U"  "!^.*$!sip:bob@sip.telebob.au!" .
```

In this case the DNS entry uses an order value of 100 and a preference of 10. The "**u**" flag indicates that the rule is terminal and that the specified URI is to be used. The service field specifies that the SIP protocol is to be used, in conjunction with the E.164 to URI (E2U) resolution service[6]. The operation of the regular expression produces the URI of the form **sip:bob@telebob.au.**

For this call request, the gateway picks the **sip+E2U** service and performs the associated regular expression transform using the original E.164 number and the regular expression. This produces the **sip:** URI. The gateway then uses the DNS a second time to translate the domain part of the URI, **sip.telebob.au**, into an IP address using a DNS A record.

The gateway then opens up a session with UDP port 5060 on this SIP server to complete the call setup, requesting a voice session with the user Bob on this server. (Figure 2).

Figure 2: Using ENUM to Call an IP Phone



If, on the other hand, Alice is sending Bob a short text message, then Bob may want this to be delivered to him as mail. Bob would add the following entry into the DNS:

```
$ORIGIN 8.7.6.5.4.3.2.1.0.1.6.e164.arpa.
   IN NAPTR 100 10 "u" "sip+E2U"     "!^.*$!sip:bob@sip.telebob.au!"   .
   IN NAPTR 102 10 "u" "mailto+E2U"  "!^.*$!mailto:bob@mail.pobob.au!" .
```

In this case the gateway would use this **mailto:** URI and use the domain part of the URI as a MX DNS query. The DNS responses are a list of mail server names and associated preferences. The gateway then selects this more preferred server and resolves this name to an IP address by a further query to the DNS for an A address record.

The gateway can complete the original text message delivery request by opening a TCP session on port 25 of the mail server and sending the message as mail addressed to user **bob@mail.pobob.au.**

Services in ENUM

Other URIs can also be associated with an E.164 number, even services not normally associated with a mapping of a telephone function. These may include **http:** URIs, even other E.164 telephone numbers, specified by **tel:** URIs.

Let's complete the example of Bob, who wants his SIP phone, mail address, Web page, and mobile telephone to be referenced from a single telephone number.

```
$ORIGIN 8.7.6.5.4.3.2.1.0.1.6.e164.arpa.
    IN NAPTR 100 10 "u" "sip+E2U"     "!^.*$!sip:bob@sip.telebob.au!"   .
    IN NAPTR 100 10 "u" "mailto+E2U"  "!^.*$!mailto:bob@mail.pobob.au!" .
    IN NAPTR 100 10 "u" "http+E2U"    "!^.*$!http://www.webhostbob.au"  .
    IN NAPTR 103 10 "u" "tel+E2U"     "!^.*$!tel:+61-4-12341234"        .
```

Alice can enter the phone number *61012345678* into her browser and retrieve Bob's Web page in response. She can address e-mail to this number and thereby send mail to Bob. Or she can make a telephone call to Bob's SIP phone, and if it does not answer she can try Bob on his mobile phone. And she can do all this from a single number.

Numerous interesting technical issues still need to be resolved, such as the necessity and level of cacheing within the global ENUM system and the creation of a standard registry scheme for ENUM service definition.

The Politics of ENUM

There is quite some depth in the capabilities of the regular expression rewrite rules in ENUM, but the basic functionality is one of mapping a telephone number to a collection of service points that are associated with the telephone customer who was assigned that telephone number.

Despite this apparent functional simplicity, ENUM appears to have a powerful set of attractors for regulatory and social controversy.

A key benefit of moving into ENUM and the associated realm of IP-based voice communications is that service creation becomes a function of the edge and not the network. What were seen as telephone network functions such as no answer and busy redirect, call forwarding, number translation, and conference calls can all be implemented as edge applications driven by user scripts, rather than what we now see in the telephone network as value-added network-based services. One way of viewing this ENUM approach is that the DNS is functionally capable of assuming the role of service control point for telephone services, taking over the role undertaken by *Signaling System 7/Channel 7* (SS7/C7).

Service creation and signaling are slipping away from the hands of network operators into the hands of enterprises and eventually consumers, in much that same way that the Internet has redefined other services in terms of edge-based function instead of network mediation.

There is also the issue of ownership of these ENUM DNS zones, or to put it another way: who gets to populate the **e164.arpa** domain with all these URIs? It could be that this is a responsibility of existing telephone service providers, because after all these entities operate the E.164 address space in each country. It could also be that this is a responsibility of Internet Service Providers (ISPs), because the data in the resource records is describing Internet-based services. Or maybe the end subscribers get to populate the DNS with their own entries, based on a collection of services that may be sourced from a set of providers.

It is quite conceivable that we could see ISPs that have no direct role in carrying voice traffic wanting access to a country's E.164 number plant in order to provide various forms of ENUM services. Given that each element of an ENUM service collection can use URIs that refer to different ISP services, it is possible that the one ENUM record can be populated by URIs referring to numerous different service providers. This model of multi-agent access to such infrastructure resource records is a novel concept to many regulatory and operating regimes, where a single operator manages the entire associated infrastructure elements that are needed to deliver a service.

Some of the discussion about ENUM has been on more subtle aspects of this mapping. There's the choice of **e164.arpa** as the common DNS root for ENUM DNS entries. At an international level there's a lingering perception that "**arpa**" is too American and that a name root of "**int**" appears to be more neutral.

But there's something else lurking here, which has surfaced within the regulatory debate in the United States. North America has the .164 country code of "1," implying that under ENUM there is a single DNS domain for ENUM, namely **1.e164.arpa.** Single domains imply single operators, and single operators have an implication of a noncompetitive monopoly service regime. There has been a call for multiple E.164 DNS root locations for North America, allowing for two or more competing service operators using different DNS hierarchies to locate their ENUM services.

On the one side there is the view that such attempts to create multiple partially populated ENUM name hierarchies to support competitive service provision in ENUM-based services are no more than an incitement to address and service chaos. This chaos would, in turn, seriously hamper the uptake of ENUM services.

On the other hand, the competitive provision proponents of multiple DNS root domains argue that a regulatory-sanctioned monopoly is still a monopoly, and this monopoly situation will likely lead to high service prices for ENUM services. This escalated pricing structure would, in turn, seriously hamper the uptake of ENUM services.

As we have seen with the use of multiple services for an **`e164.arpa`** entry, the proponents of ENUM envisage a single telephone number as being an alias not only for your Internet phone service, but also for instant messaging, e-mail, your Web page, and any other service that is associated with you. One identifier is all that would be required to reach you, using a service protocol and service provider of your choice. The implication of such a use of a telephone number is, on a personal level, no more business cards cluttered with phone numbers, fax numbers, mobile numbers, e-mail addresses, Web addresses, and instant-messaging handles. Phone numbers are still the most widely used naming scheme in communications, and the use of these numbers as a universal locator has the advantage of being linguistically neutral as well as enjoying almost ubiquitous use. There are no international character set issues within this particular number space. All we need is just one ENUM address, or just one number, for all these services.

> "One number to rule them all, one number to find them, one number to bring them all and in the darkness bind them," is the ENUM version of Tolkien's saga[7].

But one person's ease of use is often another's opportunity to exploit. To be *Lord of the Numbers* would indeed be a powerful role if such uses of ENUM were to become widespread. In addition to the commercial opportunity in operating ENUM registries, ENUM can be seen as yet another erosion of personal privacy on the Internet. It can be viewed as one more step toward the use of single individual digital identity that could be used to track individuals within the Internet. On a more immediate and mundane level of concern it opens up the opportunity for spammers to use a wealth of new ways to drive you to complete distraction.

It appears that the technical components of ENUM are generally the most straightforward part. The regulatory and social implications of ENUM are more of a concern, and it is here that with ENUM we are entering into "the Land of Mordor where the shadows lie."

Further reading:

[1] List of ITU-T Recommendation E.164 Assigned Country Codes, available online at:
http://www.itu.int/itudoc/itu-t/ob-lists/icc/e164_717.pdf

[2] Malamud, C., and Rose, M., "Principles of Operation for the TPC.INT Subdomain: Remote Printing—Technical Procedures," RFC 1530, October 1993.

[3] Fälström, P., "E.164 Number and DNS," RFC 2916, September 2000.

[4] Mealling, M., and Daniel, R., "The Naming Authority Pointer (NAPTR) DNS Resource Record," RFC 2915, September 2000.

[5] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, August 1998.

[6] Handley, M., Schulzrinne, H., Schooler, E., and Rosenberg, J., "SIP: Session Initiation Protocol," RFC 2543, March 1999.

[7] Tolkien, J. R. R., *The Lord of the Rings,* George Allen and Unwin, London 1955.

[8] **http://www.enum.org** has a good overview of ENUM and its potential application as well as references to further ENUM resources.

[9] "Interim Approval for ENUM Provisioning," see the Fragments section in this issue of *The Internet Protocol Journal,* page 37.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Scientist in the Internet area for Telstra. He is also a member of the Internet Architecture Board, and is the Secretary of the APNIC Executive Committee. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons.
E-mail: **gih@telstra.net**

One Byte at a Time

Bootstrapping with BOOTP and DHCP

by Douglas Comer, Purdue University

The process of starting a computer system is known as *bootstrapping*. In most systems, the initial bootstrap sequence begins with code in ROM, which the CPU executes. The ROM code only contains a first step—it merely loads an image into the computer's RAM and branches to the image. There are two approaches used to obtain an image:

- *Embedded system:* On a diskless computer, the ROM code contains sufficient support software to permit network communication. The ROM code uses the network support to locate and download an image.
- *Conventional computer:* On a computer that has secondary storage (for instance, a PC), the ROM code loads the image from a well-known place on disk. Typically, the loaded image consists of an operating system that then controls the computer.

In either case, the image loaded by ROM is not tailored to the specific physical hardware. Instead, an image is *generic*, which means that before it can be used, it must be configured for the local hardware. In particular, the image does not contain such networking details as the computer's IP address, address mask, or domain name. Each of these items must be supplied before applications can use TCP/IP.

Early in the history of TCP/IP, designers chose to provide a separate mechanism for each item of configuration information. Thus, the *Reverse Address Resolution Protocol* (RARP) only allowed a computer to obtain its IP address. When subnet masks were introduced, ICMP Address Mask messages were added to allow a computer to obtain a subnet mask. The chief advantage of such an approach lies in flexibility—a computer can decide which items to obtain from a local file on disk and which to obtain over the network. The chief disadvantage becomes apparent when one considers the network traffic and delay. A given computer must issue a series of small request messages. More important, each response returns a small value (for instance, a 4-octet IP address). Because networks enforce a minimum packet size, most of the space in each packet is wasted.

BOOTP

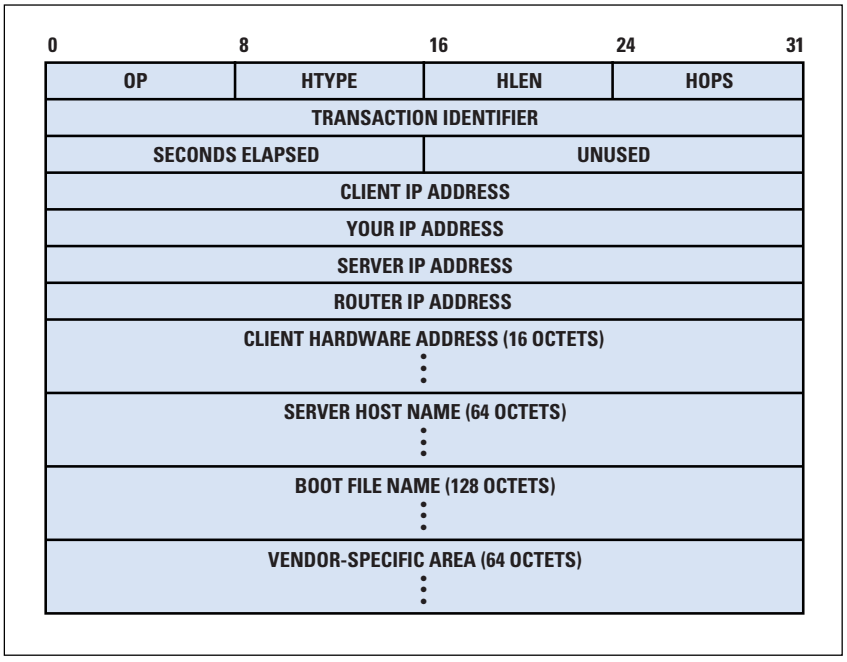
As the complexity of configuration grew, TCP/IP protocol designers observed that many of the configuration steps could be combined into a single step if a server was able to supply more than one item of configuration information. To provide such a service, the designers invented the *BOOTstrap Protocol* (BOOTP). To obtain configuration information, protocol software broadcasts a *BOOTP Request* message.

A BOOTP server that receives the request looks up several pieces of configuration information for the computer that issued the request, places the information in a single *BOOTP Response* message, and returns the reply to the requesting computer. Thus, in a single step, a computer can obtain information such as the computer's IP address, the server's name and IP address, and the IP address of a default router.

Like other protocols used to obtain configuration information, BOOTP broadcasts each request. Unlike other protocols used for configuration, BOOTP appears to use a protocol that has not been configured: BOOTP uses IP to send a request and receive a response. How can BOOTP send an IP datagram before a computer's IP address has been configured? The answer lies in a careful design that allows IP to broadcast a request and receive a response before all values have been configured. To send a BOOTP datagram, IP uses the all-1's limited broadcast address as a *DESTINATION ADDRESS*, and uses the all-0's address as a *SOURCE ADDRESS*. If a computer uses the all-0's address to send a request, a BOOTP server either uses broadcast to return the response or uses the hardware address on the incoming frame to send a response via unicast. (The server must be careful to avoid using ARP because a client that does not know its IP address cannot answer ARP requests.)

Thus, a computer that does not know its IP address can communicate with a BOOTP server. Figure 1 illustrates the BOOTP packet format. The message is sent using UDP, which is encapsulated in IP.

Figure 1: BOOTP Packet Format



Each field in a BOOTP message has a fixed size. The first seven fields contain information used to process the message. The *OP* field specifies whether the message is a *Request* or a *Response*, and the *HTYPE* and *HLEN* fields specify the network hardware type and the length of a hardware address. The *HOPS* field specifies how many servers forwarded the request, and the *TRANSACTION IDENTIFIER* field provides a value that a client can use to determine if an incoming response matches its request. The *SECONDS ELAPSED* field specifies how many seconds have elapsed since the computer began to boot. Finally, if a computer knows its IP address (for instance, the address was obtained using RARP), the computer fills in the *CLIENT IP ADDRESS* field in a request.

Later fields are used in a response message to carry information back to the computer that is booting. If a computer does not know its address, the server uses field *YOUR IP ADDRESS* to supply the value. In addition, the server uses fields *SERVER IP ADDRESS* and *SERVER HOST NAME* to give the computer information about the location of a computer that runs servers. Field *ROUTER IP ADDRESS* contains the IP address of a default router.

In addition to protocol configuration, BOOTP allows a computer to negotiate to find a boot image. To do so, the computer fills in field *BOOT FILE NAME* with a generic request (for instance, the computer can request the UNIX operating system). The BOOTP server does not send an image. Instead, the server determines which file contains the requested image, and uses field *BOOT FILE NAME* to send back the name of the file. Once a BOOTP response arrives, a computer must use a protocol like the *Trivial File Transfer Protocol* (TFTP) to obtain a copy of the image.

Automatic Address Assignment

Although it simplifies loading parameters into protocol software, BOOTP does not solve the configuration problem completely. When a BOOTP server receives a request, the server looks up the computer in its database of information. Thus, even a computer that uses BOOTP cannot boot on a new network until the administrator manually changes information in the database.

Can protocol software be devised that allows a computer to join a new network without manual intervention? Yes—several such protocols exist. For example, IPX and IPv6 can generate a protocol address from the computer's hardware address. To make automatic generation work correctly, the hardware address must be unique. Furthermore, if the hardware address and protocol address are not the same size, it must be possible to translate the hardware address into a protocol address that is also unique.

The AppleTalk protocols use a *bidding* scheme to allow a computer to join a new network. When a computer first boots, the computer chooses a random address. For example, suppose computer *C* chooses address *17*. To ensure that no other computer on the network is using the address, *C* broadcasts a request message and starts a timer. If no other computer is using address *17*, no reply will arrive before the timer expires; *C* can begin using address *17*. If another computer is using *17,* the computer replies, causing *C* to choose a different address and begin again.

Choosing an address at random works well for small networks and for computers that run client software. However, the scheme does not work well for servers. To understand why, recall that each server must be located at a well-known address. If a computer chooses an address at random when it boots, clients will not know which address to use when contacting a server on that computer. More important, because the address can change each time a computer boots, the address used to reach a server may not remain the same after a crash and reboot.

A bidding scheme also has the disadvantage that two computers can choose the same network address. In particular, assume that computer *B* sends a request for an address that another computer (for example, *A*) is already using. If *A* fails to respond to the request for any reason, both computers will attempt to use the same address, with disastrous results. In practice, such failures can occur for a variety of reasons. For example, a piece of network equipment such as a bridge can fail, a computer can be unplugged from the network when the request is sent, or a computer can be temporarily unavailable (for instance, in a hibernation mode designed to conserve power). Finally, a computer can fail to answer if the protocol software or operating system is not functioning correctly.

DHCP

To automate configuration, the *Internet Engineering Task Force* (IETF) devised the *Dynamic Host Configuration Protocol* (DHCP). Unlike BOOTP, DHCP does not require an administrator to add an entry for each computer to the database that a server uses. Instead, DHCP provides a mechanism that allows a computer to join a new network and obtain an IP address without manual intervention. The concept has been termed *plug-and-play networking*. More important, DHCP accommodates computers that run server software as well as computers that run client software:

- When a computer that runs client software is moved to a new network, the computer can use DHCP to obtain configuration information without manual intervention.
- DHCP allows nonmobile computers that run server software to be assigned a permanent address; the address will not change when the computer reboots.

To accommodate both types of computers, DHCP cannot use a bidding scheme. Instead, it uses a client-server approach. When a computer boots, the computer broadcasts a *DHCP Request* to which a server sends a *DHCP Reply*. (The reply is classified as a DHCP *offer* message that contains an address the server is offering to the client.)

An administrator can configure a DHCP server to have two types of addresses: permanent addresses that are assigned to server computers, and a pool of addresses to be allocated on demand. When a computer boots and sends a request to DHCP, the DHCP server consults its database to find configuration information.

If the database contains a specific entry for the computer, the server returns the information from the entry. If no entry exists for the computer, the server chooses the next IP address from the pool, and assigns the address to the computer.

In fact, addresses assigned on demand are not permanent. Instead, DHCP issues a *lease* on the address for a finite period of time. (When the administrator establishes a pool of addresses for DHCP to assign, the administrator must also specify the length of the lease for each address.)

When the lease expires, the computer must renegotiate with DHCP to extend the lease. Normally, DHCP will approve a lease extension. However, a site may choose an administrative policy that denies the extension. (For example, a university that has a network in a classroom might choose to deny extensions on leases at the end of a class period to allow the next class to reuse the same addresses.) If DHCP denies an extension request, the computer must stop using the address.

Optimizations in DHCP

If the computers on a network use DHCP to obtain configuration information when they boot, an event that causes all computers to restart at the same time can cause the network or server to be flooded with requests. To avoid the problem, DHCP uses the same technique as BOOTP: each computer waits a random time before transmitting or retransmitting a request.

The DHCP protocol has two steps: one in which a computer broadcasts a *DHCP Discover* message to find a DHCP server, and another in which the computer selects one of the servers that responded to its message and sends a request to that server. To avoid having a computer repeat both steps each time it boots or each time it needs to extend the lease, DHCP uses *caching*. When a computer discovers a DHCP server, the computer saves the server's address in a cache on permanent storage (for example, a disk file). Similarly, once it obtains an IP address, the computer saves the IP address in a cache. When a computer reboots, it uses the cached information to revalidate its former address. Doing so saves time and reduces network traffic.

DHCP Message Format

Interestingly, DHCP is designed as an extension of BOOTP. As Figure 2 illustrates, DHCP uses a slightly modified version of the BOOTP message format.

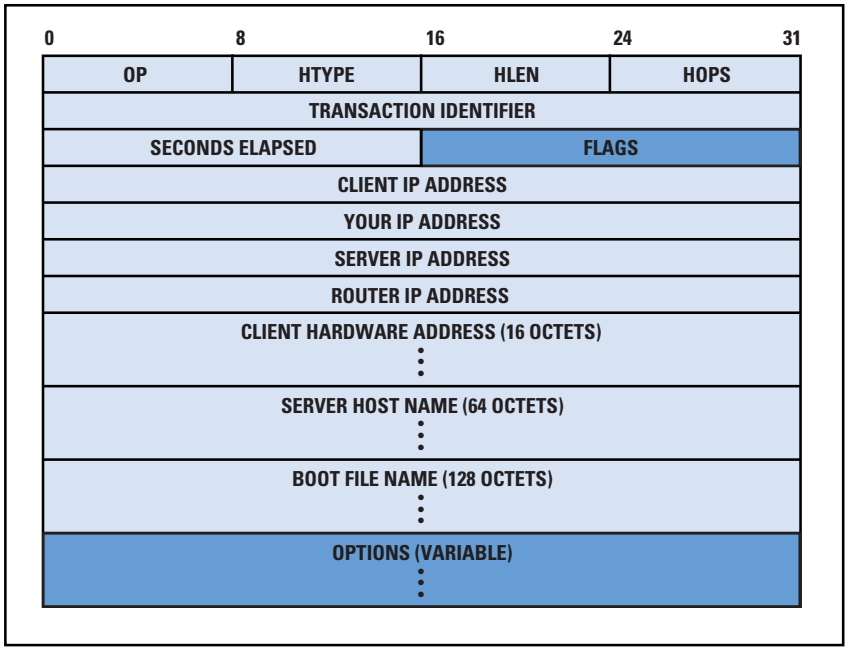


Figure 2: DHCP Message Format

Most of the fields in a DHCP message have the same meaning as in BOOTP; DHCP replaces the 16-bit *UNUSED* field with a *FLAGS* field, and uses the *OPTIONS* field to encode additional information. For example, as in BOOTP, the *OP* field specifies either a *Request* or a *Response*. To distinguish among various messages that a client uses to discover servers or request an address, or that a server uses to acknowledge or deny a request, DHCP uses a *message type option*. That is, each message contains a code that identifies the message type.

DHCP and Domain Names

Although DHCP makes it possible for a computer to obtain an IP address without manual intervention, DHCP does not interact with the Domain Name System. As a result, a computer cannot keep its name when it changes addresses. Interestingly, the computer does not need to move to a new network to have its name change. For example, suppose a computer obtains IP address **`192.5.48.195`** from DHCP, and suppose the domain name system contains a record that binds the name **`x.y.z.com`** to the address. Now consider what happens if the owner turns off the computer and takes a two-month vacation during which the address lease expires. DHCP may assign the address to another computer. When the owner returns and turns on the computer, DHCP will deny the request to use the same address. Thus, the computer will obtain a new address. Unfortunately, the *Domain Name System* (DNS) continues to map the name **`x.y.z.com`** to the old address.

For several years, researchers have been considering how DHCP should interact with the DNS. Although a dynamic DNS update protocol has been defined, it has not been widely deployed. Thus, many sites that use DHCP do not have a mechanism to update a DNS database. From a user's perspective, the lack of communication between DHCP and DNS means that when a computer is assigned a new address, the computer's name changes.

Summary

The bootstrapping sequence loads a generic image into a computer, either from secondary storage or over the network. Before application software can use TCP/IP protocols, the image must be configured by supplying values for internal parameters such as the IP address and subnet mask, and for external parameters such as the address of a default router; the process is known as *configuration*. Initially, separate protocols were used to obtain each piece of configuration information. Later, the *BOOTstrap Protocol,* BOOTP, was invented to consolidate separate requests into a single protocol. A BOOTP response provides information such as the computer's IP address, the address of a default router, and the name of a file that contains a boot image.

The *Dynamic Host Configuration Protocol* (DHCP) extends BOOTP. In addition to permanent addresses assigned to computers that run a server, DHCP permits completely automated address assignment. That is, DHCP allows a computer to join a new network, obtain a valid IP address, and begin using the address without requiring an administrator to enter information about the computer in a server's database. When DHCP allocates an address automatically, the DHCP server does not assign the address forever. Instead, the server specifies a lease during which the address may be used. A computer must extend the lease, or stop using the address when the lease expires.

For Further Study

Details about BOOTP can be found in reference [1], which compares BOOTP to RARP and serves as the official protocol standard. Reference [2] tells how to interpret the vendor-specific area, and reference [3] recommends using the vendor-specific area to pass the subnet mask. Most uses of BOOTP have been replaced by DHCP. Reference [4] contains the specification for DHCP, including a detailed description of state transitions. A related document, [5], specifies the encoding of DHCP options and BOOTP vendor extensions. Finally, reference [6] discusses the interoperability of BOOTP and DHCP. The chair of the DHCP working group, Ralph Droms, and Ted Lemon have written a book about DHCP [7].

References

[1] W. J. Croft, J. Gilmore, "Bootstrap Protocol," RFC 951, September 1985.

[2] J. K. Reynolds, "BOOTP Vendor Information Extensions," RFC 1084, December 1988.

[3] R. Braden (ed), "Requirements for Internet Hosts—Application and Support," RFC 1123, October 1989.

[4] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131, March 1997.

[5] S. Alexander, R. Droms, "DHCP Options and BOOTP Vendor Extensions," RFC 2132, March 1997.

[6] R. Droms, "Interoperation between DHCP and BOOTP," RFC 1534, October 1993.

[7] R. Droms and T. Lemon, *The DHCP Handbook: Understanding, Deploying, and Managing Automated Configuration Services,* ISBN 1578701376, MacMillian, 1999.

[This article is adapted from *Computer Networks and Internets, with Internet Applications, 3rd edition,* by Douglas Comer, with CD by Ralph Droms, ISBN 0130914495, Prentice Hall, 2001.]

Dr. DOUGLAS COMER is a professor of Computer Science at Purdue University, consultant to industry, and an internationally recognized authority on TCP/IP. He has written numerous research papers and textbooks, including the classic three-volume reference series *Internetworking with TCP/IP,* and currently heads research projects. He designed and implemented X25NET and Cypress networks, and the Xinu operating system. He was a principal on the CSNET project, is director of the Internetworking Research Group at Purdue, editor of the journal *Software—Practice and Experience,* a former member of the IAB, and a Fellow of the ACM.
E-mail: **comer@cs.purdue.edu**

Book Review

The Elements of Networking Style

The Elements of Networking Style, by M. A. Padlipsky, originally published by Prentice-Hall, 1985, ISBN 0132681110; now available from iUniverse, 2000, ISBN 0595088791.

Sometime in the autumn of 1986, I read Padlipsky on a flight from Boston to San Francisco, and about 15 minutes into it I began to get enraged. A few minutes later, I was snickering. By the time the attendants came around with profferings of alleged comestibles, I was laughing aloud, and a gentleman sitting near the window was grateful that there was a vacant seat between us.

Padlipsky brought together several strands that managed to result in the perfect chord for me over 15 years ago. I reread this slim volume (made up of a Foreword, 11 chapters (each a separate arrow from Padlipsky's quiver) and three appendixes (made up of half a dozen darts of various lengths and a sheaf of cartoons and slogans) several months ago, and have concluded that it is as acerbic and as important now as it was 15 years ago.

The instruments Padlipsky employs are a sharp wit (and a deep admiration for François Marie Arouet), a sincere detestation for the ISO Reference Model, a deep knowledge of the *Advanced Research Projects Agency Network* (ARPANET)/Internet, and wide reading in classic science fiction.

Arouet is better known by his pen name, Voltaire. He was a social rebel, a political agitator, and an acerbic satirist comparable to Swift. Isaiah Berlin, in a lecture published in Salmagundi 27 [1974], remarks:

> "Voltaire is the central figure of the Enlightenment, because he accepted its basic principles and used all his incomparable wit and energy and literary skill and brilliant malice to propagate the principles and spread havoc in the enemy's camp. Ridicule kills more surely than savage indignation..."

Padlipsky is pungent and sharp and witty ... and knowledgeable. His critiques of X.25, of the *International Organization for Standardization* (ISO) seven-layer cake, and of the standards process in general, are still relevant.

History

In the early 1970s, the CCITT (now the ITU), made up of PTTs and monolithic telcos, fixed upon a putative standard for a network interface protocol, X.25. First approved in 1976, and revised in 1977, 1980, 1984, 1988, and 1992, X.25 was unsatisfactory in its original form and remains less than effective.

One of the greatest drawbacks is that it is basically a store-and-forward mechanism, meaning that it has an intrinsic delay and (as noted by Sangoma Technologies) this delay is typically 0.6 seconds. It also requires a great deal of buffering space.

Padlipsky's "Critique of X.25" (Mitre Corporation Report, M82-50, September 1982; RFC 874 12 August 1983) is revised as Chapter 9 in *The Elements of Networking Style*. Padlipsky has restored, however, his original title: "Low Standards."

Flush with the failure of X.25, the *Consultative Committee for International Telegraph and Telephone* (CCITT) moved ahead.

In 1977, the British Standards Institute proposed to ISO that an architecture was needed to define the communications infrastructure. To me, this, as with *International Federation for Information Processing* (IFIP), CCITT, and similar efforts, shows how "the road to hell is paved with good intentions." Because X.25 was unsatisfactory, the IFIP Working Group was set up in the hope that that the technological community could forestall the highly political arena of ISO. (It didn't.)

ISO set up a technical committee [ISO/TC 97/SC 16]. The next year (1978), ISO published its "Provisional Model of Open Systems Architecture" [ISO/TC 97/SC 16 N 34]. This was labeled a "Reference Model," and referred to as the *Open Systems Interconnection Reference Model* (OSIRM or ISORM—pronounced "eye-sorm"—by Padlipsky).

In general, it was based on work done by Mike Canepa's group at Honeywell Information Systems, which came up with a seven-layered architecture, which itself owed a great deal to IBM's proprietary *Systems Network Architecture* (SNA). SNA had been announced in 1974, and its seven layers do not correspond exactly to OSI/ISORM's. TC 97/SC 16 turned over proposal development to the *American National Standards Institute* (ANSI), to which Canepa and his technical lead, Charlie Bachman, presented their layered model.

This, in turn, was the only proposal presented to the ISO subcommittee at a meeting in Washington in March 1978. It was accepted and published immediately. A "refined" version of the ANSI submission to ISO appeared in June 1979. This published version is nearly identical to Honeywell's of 1977.

Rage and Ridicule

While he eschews the history I've outlined here, Padlipsky is enraged by the standards process and its results. As Dave Walden and Alex McKenzie (both then at BBN, both now retired) pointed out in 1979, both virtual circuit and datagram services are valuable. "An international standard would do well to support both." [*IEEE Computer,* September 1979].

The 1977–1979 models were such that extant host-host protocols did not fit ISORM. ISO was trying to construct a set of geometric figures that would be a "tidy model." The ARPANET workers, of whom Padlipsky was one, were interested in getting things to actually work. They were into pushing bits around the system.

The irascible Padlipsky has described the OSI system as two high rises with parking garages. The two high-rises are seven-story buildings; the parking garages are the three-story X.25 structures.

John Quarterman once pointed out:

> "OSI specified before implementation. So specification took forever and implementation never happened, except for bits and pieces. In addition, heavy government backing (by the EC, now the EU, and various national governments) led some OSI participants to attempt to substitute official authority for technical capability. OSI and TCP/IP started at about the same time (1977). OSI wandered off into the weeds and TCP/IP won the race. Those governments that backed OSI bet on the wrong horse."

TCP/IP had clearly "won the race" by the early 1980s; it took till 1994 for the U.S. government to recognize the de facto standard by rescinding its *Federal Information Processing Standards* (FIPS). At that time, too, the *Defense Data Network* (DDN) was made up of IP router nets, not X.25-based nets.

In a totally different vein, there's Chapter 11: "An Architecture for Secure Packet-Switched Networks" (based on a presentation to the Third Berkeley Workshop on Distributed Data Management and Networking, August 1978). Here, Padlipsky suggests per-host processes. It was a really good notion.

Padlipsky's rants—and many of the chapters are just that—precede Quarterman's remarks by nearly a decade. But they are worth reading (and rereading).

I'm glad *The Elements of Networking Style* is available again.

—Peter H. Salus
peter@matrix.net

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

Fragments

Stephen D. Crocker Receives 2002 IEEE Internet Award

The *Institute of Electrical and Electronics Engineers* (IEEE) has named Stephen D. Crocker, chief executive officer of Shinkuro, Inc. in Bethesda, Md., as recipient of the 2002 *IEEE Internet Award*. The award recognizes Crocker for his leadership in the creation of key Internet protocols. It will be presented on 19 June, at INET 2002, in Arlington, Va.

In the formative days of the Internet and its predecessor, the ARPANET, Crocker led the development of crucial technologies, processes and organizations that continue to support the Internet today. At the University of California at Los Angeles, Crocker and his team developed protocols for the ARPANET such as the *Network Control Protocol*. NCP laid the groundwork for today's *Transmission Control Protocol* (TCP). Crocker also founded and led the *Network Working Group* (NWG), which has evolved to become the *Internet Engineering Task Force* (IETF).

In organizing the notes from the first few meetings of NWG, Crocker was anxious to expand the community and invite further discussion and responses, and thus named the series *Requests for Comments*. RFCs remain a mainstay of Internet protocol publishing today, and have played a big part in creating the environment of open and evolving standards of the Internet.

"The Internet Society is honored that INET 2002 was chosen as the venue to present this year's prestigious IEEE Internet Award," said Lynn St. Amour, president and CEO of the Internet Society (ISOC). "Dr. Stephen Crocker is highly regarded throughout the international Internet community and we're pleased that his contributions will be recognized at INET 2002 in front of his peers."

Crocker's many contributions to the Internet also include extensive work organizing the standards process of the IETF, where he has served as area director of security and on the Internet Architecture Board. Crocker previously worked for the University of Southern California Information Sciences Institute in Marina del Rey, the Aerospace Corporation in El Segundo, Calif., and at Trusted Information Systems, Inc., in Glenwood, Md. In 1994, he co-founded CyberCash of Reston, Va., and served as its senior vice president for development and chief technology officer. He also has started other ventures including Steve Crocker Associates in Bethesda, Md.; Executive DSL in Bethesda, Md.; and Longitude Systems in Chantilly, Va.

He has served on the Council of Visitors at the Marine Biological Laboratory, as part of the National Research Council Study of Information Systems Trustworthiness and currently chairs the ICANN Security and Stability Advisory Committee and the ISOC 2002 Jonathan B. Postel Service Award Committee. The author of numerous papers, Crocker also holds patents in relation to his security and electronic commerce work.

He received his bachelor's degree in mathematics and doctoral degree in computer science, both from UCLA, he and studied artificial intelligence at the Massachusetts Institute of Technology.

The IEEE is the world's largest technical professional society with more than 377,000 members in approximately 150 countries. Through its members, the IEEE is a leading authority on areas ranging from aerospace, computers and telecommunications to biomedicine, electric power and consumer electronics. Additional information is available at **http://www.ieee.org**

The Internet Society **http://www.isoc.org/** is a non-profit, non-governmental, open membership organization whose worldwide individual and organization members make up a veritable "who's who" of the Internet industry. It provides leadership in technical and operational standards, policy issues, and education. ISOC is the organizational home of the International Engineering Task Force, the Internet Architecture Board, the Internet Engineering Steering Group, and the IETF—the standards setting and research arms of the Internet community. For information about INET 2002 please visit **http://www.inet2002.org**

Interim Approval for ENUM Provisioning

The *International Telecommunication Union* (ITU) and the *Internet Architecture Board* (IAB) recently announced interim approval for a single domain for ENUM, a technology that builds a bridge between the public switched telephone network and the Internet.

Voice on IP networks today operate by translating telephone numbers to IP addresses and placing an H.323 or SIP call to the device. The interchange format and translation record has not heretofore been standardized, limiting the possibility of deployment of multi-corporate and international Voice on IP services. Under the ENUM proposal, E.164 numbers can be represented as Internet Domain Names,providing a scalable and standard way to translate the numbers, and opening the way to such services. ITU has begun approving delegations for the purposes of trials. "The lack of an interoperable standard way to turn a telephone number into an IP Address has been one factor limiting the deployment of Voice on IP services internationally," said Leslie Daigle, Chair of the IAB.

If desk-mounted computers or servers are given telephone numbers as well as mnemonic names, this system further enables common telephone handsets to place Voice or Video on IP calls to such computers. This is a significant step towards integrating Internet-based services with the global telephone network, and the current agreements between IAB and ITU will allow trials to take place.

Patrik Fältström, member of the *Internet Engineering Steering Group* (IESG), said that "the integration of the desktop telephone and computer allows corporations to simplify their internal networks."

Roy Blane, Chair of ITU-T's Study Group 2, concurred, saying that "In the long term this protocol may facilitate many new internet services. In the short term, countries wishing to trial the system can begin work on developing it."

This interim approval is made possible due to cooperation between ITU, IAB and the IETF. As outlined in the ENUM specification document, RFC 2916, sub-domains from a single domain will be delegated after acceptance by the registries according to the existing assignment of country codes in the telephone address space. Information on how the ENUM registration requests will be processed can be found at:
http://www.ripe.net/enum/

The IETF is an international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. The definition of the ENUM protocol, as proposed by the IETF can be found at **http://www.ietf.org/rfc/rfc2916.txt** The IETF is an organized activity of the Internet Society.

The ITU is a global organization where the public and private sectors cooperate for the development of telecommunications and the harmonization of national telecommunications policies. Study Group 2 of the *ITU Telecommunication Standardization Sector* (ITU-T), where work on ENUM is being carried out, is the Lead Study Group on Service definition, Numbering, Routing and Global Mobility and is responsible for the operational aspects of service provision, networks and performance. More information on the ENUM protocol, and the issues related to it, can be found at **http://www.itu.int/ITU-T/worksem/enum/index.html**

Committee on ICANN Evolution and Reform posts Recommendations

Following the publication in February of "President's Report: ICANN—The Case for Reform," by Stuart Lynn, President and CEO of *The Internet Corporation for Assigned Names and Numbers* (ICANN), a committee of the board has been examining the details of the restructuring proposal, receiving input from the community at large, and publishing several documents with recommendations. You can find pointers to all of these documents in the "Announcements" section at **http://www.icann.org**

Upcoming Events

INET 2002, the annual conference of the Internet Society, will be held June 18–21, 2002 at the Crystal Gateway Marriott, in Arlington, Virginia (5 minutes from downtown Washington, DC).
http://www.inet2002.org/

The *IETF* will be meeting in Yokohama, Japan, July 15–19, 2002 and in Atlanta, Georgia, USA, November 17–22, 2002.
http://www.ietf.org/meetings/meetings.html

ACM SIGCOMM 2002 is the annual conference of the *Special Interest Group on Data Communication* (SIGCOMM), a vital special interest group of the *Association for Computing Machinery* (ACM). This year, SIGCOMM will be held in Pittsburg, Pennsylvania, August 19–23.
http://www.acm.org/sigcomm/sigcomm2002/

ICANN will meet in Bucharest, Rumania, June 24–28, 2002 and in Shanghai, China, October 27–31, 2002.
http://www.icann.org/meetings/

The *Asia Pacific Network Information Centre* (APNIC) will hold its next Open Policy Meeting, September 3–6, 2002 in Kitakyushu, Japan. **http://www.apnic.net/meetings/index.html**

The next *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will take place February 19–28 in Taipei, Taiwan. **http://www.apricot2003.net/**

Errata List

This is the 17th issue of *The Internet Protocol Journal.* Inevitably, some minor, and a few major errors have made their way into print since our June 1998 issue. We are planning to publish a list of correctionson our Web site in the near future. Since the online material is a reflection of the printed version, we feel it would be inappropriate to simply "silently" correct the online editions, thereby rewriting history. Instead, a list of the errors along with the corrections will be presented.

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

September 2002 Volume 5, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

The *Internet Protocol Journal* (IPJ) does not have a marketing department. New subscribers learn about IPJ through our Web page, or perhaps by picking up a copy at an Internet conference or meeting such as the IETF. Word of mouth is perhaps the most effective "marketing tool." I was reminded of this in July when an article in IPJ was mentioned on the *SlashDot* Web site. Within a few days we received more than 900 new subscriptions, on the order of ten times the normal sign-up rate. I think this illustrates the power of the Web as a tool for information dissemination.

I am a big fan of visitor networks. Such networks, typically found in larger hotels, allow high-speed access to the Internet for a daily or weekly fee. Although most of the conferences and meetings I attend have purpose-built "terminal rooms," it is still nice to be able to work in your hotel room at speeds orders of magnitude better than what can be obtained with a dialup modem. Dory Leifer explains how visitor networks are designed and operated in our first article.

In a previous article we explored the basics of IEEE 802.11 wireless networking. Such networks are growing at an amazing rate. Reports about wireless network "wiretapping" are frequently found in the trade press. Gregory R. Scholz describes an architecture for securing wireless networks, using a variety of technologies and protocols.

Geoff Huston is back with another opinion piece, this time discussing the role of the *Internet Service Provider* (ISP) as a "common carrier." Many ISPs are finding themselves in the middle of disputes between customers, copyright owners, regulators and others. What role should an ISP play in this regard? Geoff provides some answers.

Please continue to provide your feedback to anything you read in this journal. Our "Letters to the Editor" section provides a sample of some of the correspondence we receive. As always, use **ipj@cisco.com** to contact us.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Visitor Networks

by Dory Leifer, DEL Communications Consulting

Visitor networks are LANs that are most often deployed in hotels, airports, cafés, college campuses, apartments, and other locations. They enable the public network access on an ad-hoc basis. Recently, 802.11 "hot spots" have gained increased attention; they represent one example of a visitor network.

Visitors attach devices such as a laptop or *personal digital assistant* (PDA) that they use only while traveling or, more often, they attach machines normally used in the office or home. These machines can be thought of as "visiting hosts."

This article explores some of the technical issues with IP visitor networks and considers practical options for service provider deployment on wired Ethernet and wireless networks. In exploring deployment options, the article focuses mainly on solutions that do not require client software on the visiting host. These clientless techniques are based on heuristics and, although they do not work effectively under all circumstances, they have proven to be quite useful in practice.

For this discussion, it is assumed that the service provided by the visitor network is for access in one location at a time. Therefore, the article does not address network hand-off for mobile clients that are moving from one network attachment point to another while attempting to maintain connectivity.

Traditional LANs vs. Visitor Networks

Traditional LANs have been well optimized for enterprise networks. They provide high bandwidth and an economical and universal method of delivering network connectivity. In comparison, visitor networks are a rather curious hybrid of a LAN and a public network, such as one used for dial-in network access. Their objective is to physically use LANs to deliver what has normally been considered a public network service: *universal access*.

In enterprise networks, traditional LANs are usually carefully administrated. Normally the connected hosts are owned and administrated by the same enterprise that operates the network. Hosts that are connected to the network are configured according to the designated protocol and address schemes. They are often configured for at least *Simple Mail Transfer Protocol* (SMTP), *Post Office Protocol* (POP), file, and print sharing. On visitor networks, the hosts are typically owned and configured by the visitors, while the service provider administrates the network.

This difference in administration creates a serious challenge for the visitor network. The network must support a wide range of configurations because they will differ from one visiting host to another. For example, if a host had previously been configured for a static IP address, that address is likely to be from a different subnet, perhaps from a private network that the visitor normally uses at the office. Even if a host gets some of its configuration from *Dynamic Host Configuration Protocol* (DHCP), *Domain Name System* (DNS) and SMTP servers may refer to addresses or names on a private network that are not reachable on the visitor network.

Traditional wired LANs normally span physically secure areas, so any person who has access to the Ethernet wall jack for the building can connect anything to the network. With a visitor network it may be undesirable to allow everyone access. For example, a visitor network deployed in a university library may be available only to students. Similar to public dial-in access, visitor networks often rely on authentication and authorization before granting service.

Whereas LANs are excellent at facilitating peer-to-peer services such as file and print sharing between connected hosts, visitor networks often attempt to minimize these direct interactions between visitors, instead establishing a set of services that the service provider itself offers or simply routing the IP packets off the LAN to an Internet Service Provider. Minimizing interactions between visitors is desirable because service providers will want to reduce the risk of a visitor's machine being attacked by another visitor. On some occasions, however, visitors who do trust each other may want to use the visitor network for file sharing, printing, or even network gaming.

Going Clientless

One of the most difficult choices for service providers deploying visitor networks is to decide whether or not to rely on the installation of specialized client software on the visiting host.

Client software allows specific network protocols to be passed between the client and the visitor network. Protocols such as *Point-to-Point Protocol over Ethernet* (PPPoE)[1], *Layer 2 Tunneling Protocol* (L2TP)[2], and Mobile-IP[3] support both authentication as well as IP tunneling to assist in routing and address assignment. On some wireless LANs and networks with high-end Ethernet switches, 802.1x (which will be discussed in more detail later) supports flexible authentication schemes and aids in data encryption[4]. Although these protocols implemented on the client can present a significant technical advantage for implementing visitor networks, they require at least some modification to the configuration on the visiting host.

The lowest common denominator for traveling laptops is a simple TCP/IP stack and a browser. If the service can accommodate the visitor with only these items, the visitor network becomes much more suitable to the broadest audience. Of course without authentication, tunneling, and client configuration available from client software, the visitor network must rely on a set of heuristics or, said by some, hacks, to perform its tricks. Subsequent sections of this article illustrate technically how a visitor network can operate without relying on the installation of client software.

The service provider may choose to distribute client software in a situation where the visitor may use the service repeatedly. In many other situations, however, it is not feasible. For example, the last thing that travelers want to find in a hotel room upon arriving at midnight and needing a network connection is a CD-ROM full of new software drivers to drop on their laptop before using the hotel's in-room Ethernet. Even if the provided software does nothing but change the configurations, such as select a Web proxy server, it may have negative consequences when the laptop is returned to the office. Such added steps could also discourage visitors from using the visitor network again.

Visitor Network Basics

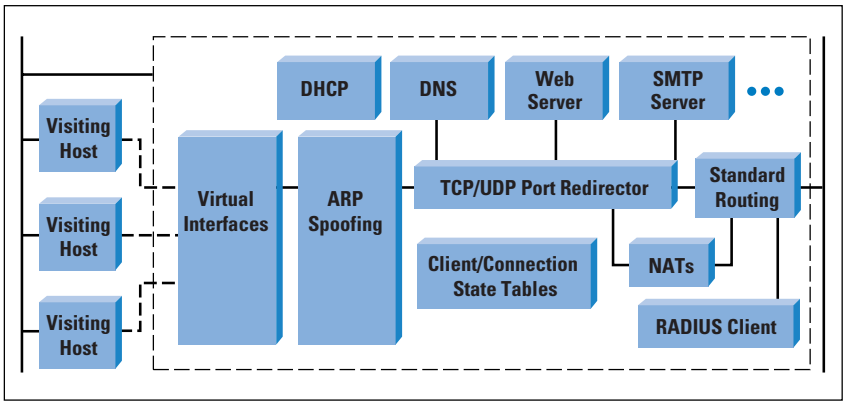
There are no hard guidelines or standards on what constitutes a visitor network. However, numerous vendors are selling devices that operate with wired and wireless networks, and act as gateways between the visitor network and the traditionally routed infrastructure. The typical visitor experience proceeds as follows (this is essentially a clientless example): the visiting host would not require the installation of special software, and in many cases would not require configuration changes:

- The visiting host is physically attached to the network by connecting to a twisted-pair Ethernet port.
- Visitors open their browser and attempt to load any page with the *Hypertext Transfer Protocol* (HTTP).
- Regardless of the specified URL, the browser loads a default page that requests authentication or billing information.
- When authenticated, the visitors now have general Internet access.
- An accounting record describing a visitor's session is generated and processed by the service provider's billing system, resulting in a charge on either the visitor's account or a corporate account.

Visitor Gateways

Visitor networks can be implemented with a special-purpose device called a "visitor gateway." Figure 1 illustrates the basic functional schematic of an example device. (Unfortunately, just about every vendor selling these devices uses a different name. This article uses the term in a generic sense and not to refer to any company's particular product.)

Figure 1: Visitor Gateway



The visitor gateway sits between the LANs used to provide service to the visitors and a standard routed interface. Physically, a visitor gateway is a device that appears much like a router or firewall, with minimally two Ethernet interfaces.

Hybrid of NAS and LAN

The following sections focus on the visitor gateway, specifically its operational model, its handling of various Internet packet types, *virtual LANs* (VLANs), authentication, and accounting.

Visitor gateways behave as a hybrid of a standard LAN and a *Network Access Server* (NAS). For illustration, one can compare the operation of the visitor gateway with the operation of a NAS. Like a NAS with individual modem ports, the visitor network gateway typically builds virtual port structures as new hosts are discovered on the connected LAN. These virtual interfaces are configured by the gateway to accommodate the IP addresses used and referred to by the visiting host. The visitor gateway may create a virtual port structure for every host based on its *Media Access Control* (MAC) address or VLAN identifier and treat every virtual interface as an independent subnet upon which the visiting host and the virtual interface of the visitor network are the only attachments. Think of the relationship as a logical point-to-point link.

Conversely, the NAS, using the *Point-to-Point Protocol* (PPP)[5] on a dial-in connection, has a significant advantage over the visitor gateway in this scenario. PPP allows the NAS to negotiate an acceptable IP address for the dial-in client, set the client's default gateway, and even in some cases configure the client's DNS. The NAS normally has at least *Password Authentication Protocol* (PAP) and *Challenge Handshake Authentication Protocol* (CHAP) for authentication. If the visiting host requests configuration through DHCP[6], the visitor network has an opportunity to assign private or public addresses that are mutually convenient for both parties. On the other hand, if the visiting host already has a static address configured for its native network, for example, then the visitor gateway must spoof or imitate the behavior of the configured subnet.

The appeal of PPP in the dial-in world led to the recent development of PPPoE for LANs. Although PPPoE has been used with service selection gateways to offer public *Digital Subscriber Line* (DSL), there has been little use of it on visitor gateways. This is likely to be true because of the lack of a ubiquitous client and the complexities of solving multilevel authentication and encryption involving the local link, local network, and private network. PPPoE certainly is worth future study for visitor networks.

ARP

Hosts learn Layer 2 MAC addresses using the *Address Resolution Protocol* (ARP). Although hosts and routers respond only when asked about the IP address of their interfaces or those on a proxy-ARP table, visitor gateways usually respond with their own MAC address to any ARP requests from the attached visiting hosts, effectively proxying for the host's default gateway (if one is configured). The visitor gateway can also configure the interface address of its virtual port based on the host's IP address. In this manner, the gateway auto-configures itself to accommodate the visitor, who can continue to use his/her configured address.

Used on a standard shared LAN, this technique only goes so far. If, for example, one host on the visitor network shared its default router configuration with the IP addresses of another host (not that uncommon for private network numbers), then when the first host attempted to get the MAC address of its default router, it would end up with two responses, one from the visitor gateway and one from the other host on the LAN.

TCP/UDP Port Redirector

The visitor gateway for each *Transmission Control Protocol* (TCP) and *User Datagram Protocol* (UDP) packet received from the visiting host decides whether to pass the packet through or direct it to a local service such as DNS, SMTP, or Web server. It makes this decision based on some configured policy from the service provider (such as to redirect all SMTP) and from authorization states of the visitors. For example, if the service provider wishes to charge visitors $10 for daily access at a hotel, the port redirector could reflect HTTP requests to the local Web server that would, in turn, present the option to the visitor. Subsequent HTTP requests presumably would always be passed transparently through the gateway to the intended address.

The operation of the redirector is fairly simple. It works as a backwards network address-port translator. Instead of modifying the source, it modifies the destination and then applies standard IP forwarding on the resulting packets.

DNS

Visitor gateways typically implement proxies for domain name service requests and channel all DNS requests from the visiting host through the proxy. This serves at a minimum to reflect DNS requests to a closer DNS server, a useful performance advantage if the visitor's configured DNS server is a considerable distance away. Of greater significance is that it allows general Internet access by the visitor even if the configured DNS server is on a private network, which is now unreachable because the visitor's laptop has been moved from the office.

Redirecting to a DNS server not of the visitor's choosing may work smoothly until the visitor attempts to resolve domain names known only to the real DNS server on the private network. There is, of course, a limit to how well you can hide reality.

One common problem encountered by visitor networks is with a Web proxy on a private network. If the visitor refers to a Web proxy by name, the visitor gateway may choose to respond, inventing an IP address for the proxy and then assuming, by itself, operation of the proxy function. This technique has to be used with some care because hosts often cache DNS responses; these are effectively convenient lies that could end up being carried as "dirty entries" on the visitor's machine for longer than intended.

Rewriting DNS queries and responses does open the opportunity for the service provider to "assume" (some may say "hijack") sites. This opens the door to the possibility that, for example, `yahoo.com` is resolved to an address that is not Yahoo but rather a Web site with an affiliation to the service provider. Although this is a policy and business issue for the service provider, it is likely to irritate quite a number of visitors and reduce the perceived value of the service.

NATs

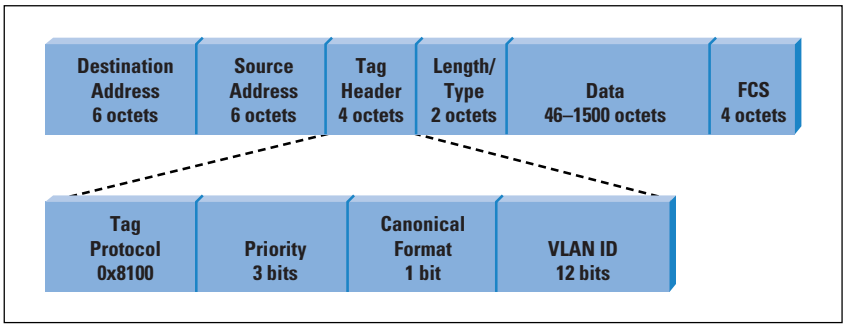
Visitor network gateways frequently use *Network Address Translation* (NAT), and often with port translation, in order to conserve IP addresses by sharing a small address pool with a large number of visitor hosts. In addition, NAT is required by the gateway if the source address used by the visiting host is not routable by the rest of the network back to the visitor gateway. This is almost always the case when the visiting host is using a static preconfigured IP address from another network. The gateway may choose its application of NAT based on policy. For example, two visitors may be configured for DHCP but one is assigned a private "Net 10" (RFC 1918) address that is passed through a NAT while another is assigned a routable address. In practice this flexibility is useful for service in apartments where the visitors are expected to "visit" for months. The service provider may choose to offer tiered services, one with a routable address suitable for the customers to run servers, and another with a private address suitable only for outgoing connections (e-mail, HTTP, and so on).

VLANs

The visitor gateway—modeling its relationship with visiting hosts as a virtual point-to-point link—may attempt to ignore the fact that hosts are on a shared network. However, certain interactions between hosts are inevitable on a shared LAN. For example, if a visitor's Windows laptop is configured for file sharing with no security enabled, other visitors may see, or worse, have permission to write to, critical files.

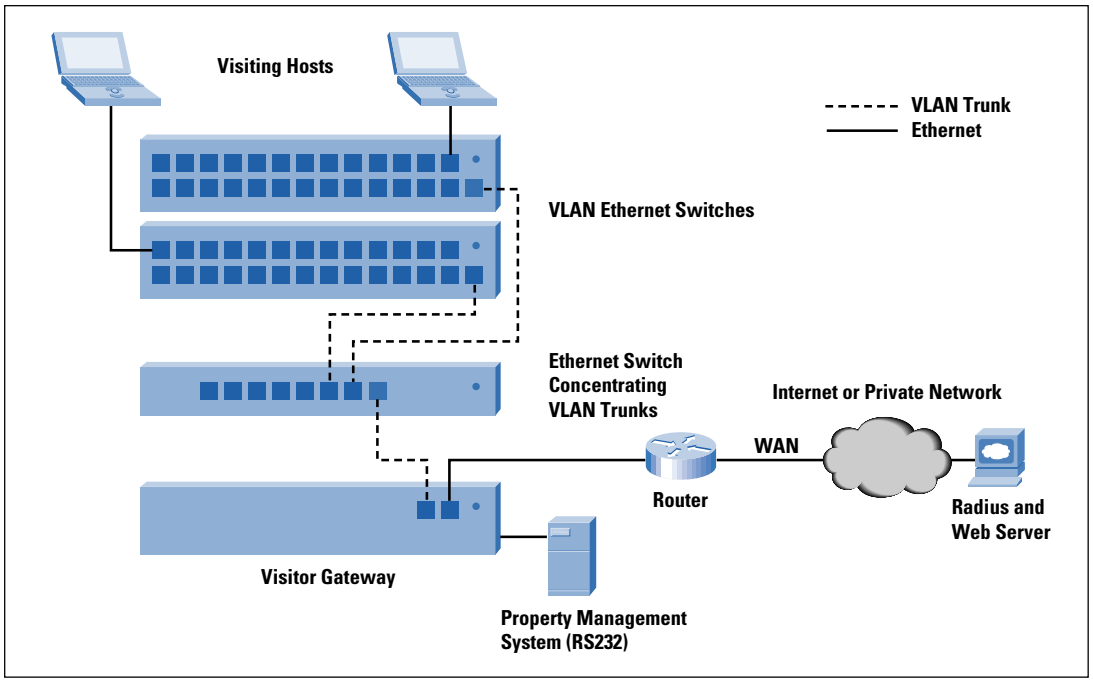
Virtual LANs provide a solution for isolating individual clients. On a wired Ethernet, many modern Ethernet switches can be configured to implicitly treat each port as a member of a different VLAN. For example, port 1 could be on VLAN 11; port 2 on VLAN 12; and so on. The visitor gateway is connected to one or more "trunk" ports that are configured as a member of all VLANs. This effectively allows another level of addressing so the visitor gateway can individually address a single Ethernet network connected to a port. The VLAN switches then act as simple concentrators. If a visiting host attempts to broadcast or multicast, these frames end up only traveling to the gateway and are not seen by other visiting hosts.

Figure 2: VLAN Frame Format



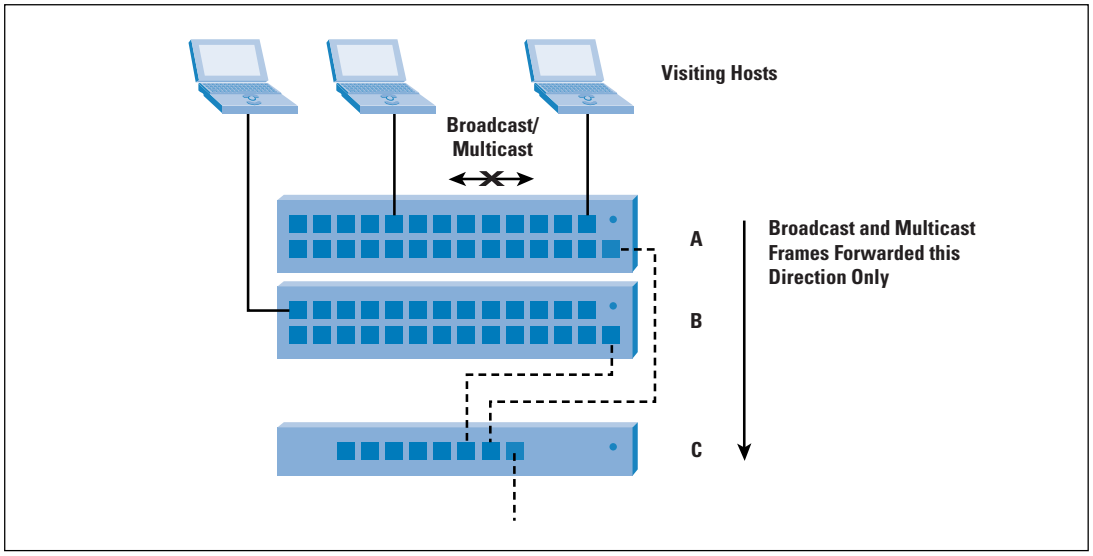
The VLAN frame format is shown in Figure 2. IEEE 802.1q defines the "tagging"[8]. The VLAN-enabled Ethernet switch can add the appropriate headers to standard Ethernet frames, and it forwards these through the trunk port. Optionally, another Ethernet switch concentrates the trunk traffic and attaches to the visitor gateway, as seen in Figure 3. One potential catch is that some Ethernet switches will not pass the oversized (maximum 1504 octet) VLAN frames; others attempt to be "overly aware" of the VLAN membership rules and insist on configuration of each of the VLANs, a challenging prospect if you are concentrating thousands of ports, each with a unique VLAN identifier.

Figure 3: VLAN Configuration



Some Ethernet switch vendors have implemented a nonstandard technique whereby broadcasts and multicasts are forwarded exclusively to a designated port, the theory being that if a host's broadcast and multicast frames do not get forwarded to other hosts, the hosts effectively will not "see" each other because they do not see ARP requests or higher-layer service advertisements. In some ways, this is simpler than using VLANs and provides some isolation over standard Ethernet networks.

Figure 4: Switch Multicast Blocking



Combinations of these switches with normal ones can lead to some interesting frame forwarding scenarios. For example, as seen in Figure 4, Ethernet switches A and B are each connected to Ethernet switch C. Visiting hosts are attached to the ports on A and B. A and B are designed to have the forwarding restriction described, but C is a normal switch. This means that a broadcast from a visitor connected to A will not be seen by other visitors on A (by nature of the restriction) but it will be forwarded "upstream" to C, which will then forward it to B. Because B received it coming from the upstream, it will forward it to all the visiting hosts on B, causing the isolation technique to fail. A, B, and C all need to have the forwarding restriction.

Web-Based Authentication and Policy

Visitor networks often avail themselves of the one reliable way to converse with a human without additional client software: the Web browser. By selectively reflecting HTTP requests to the local gateway, the gateway can perform or facilitate several operations:

- Authenticate users with traditional username/password—The visitor gateway may, in turn, use a *Remote Access Dial-In User Service* (RADIUS)[7] authentication request to validate the user.
- Provide links within a "walled garden"—sites that can be visited without authentication—These sites are implemented with either a Web proxy inside the gateway or access control lists effective on the individual visiting host's virtual interface.
- Gather and validate credit card information through third-party credit card processing Web sites
- Offer visitors Web pages they can use to subscribe to services or to change service parameters

Using the browser can have a significant advantage, even over installed client software. The browser allows a conversation with a human user instead of a software client. This affords the network provider a wide variety of options, such as dealing politely with an authentication rejection, providing additional troubleshooting help, or confirming "conditions of use" before the user accepts charges. It is also a place for offering the user other products and services through Web links.

A central repository for visitor policy and configuration is especially important when a large number of gateways are deployed in disparate physical locations. An interesting option for visitor gateways is for them to learn policy by participating in the exchange of HTTP between the visiting host and an external Web server. The visitor gateway can piggyback the origin and state of a visiting host in a URL and refer the visitor's browser to a Web site. This origin information when presented to a service selection application running in a provider's data or operation center allows the application to determine which gateway the visitor is attached to as well as the visitor's virtual port identification and MAC address.

With the origin information, the service selection gateway can present the visitor with any number of billing, quality of service, or IP addressing options that apply to his/her connection. When the service selection application needs to affect the policy information stored in the visitor gateway, it can use a similar piggyback technique in the return direction.

Accounting

Finding an easy-to-deploy accounting method is crucial for service providers to generate accurate billing. The visitor gateway may send RADIUS accounting records in response to connections and disconnections made by visiting hosts. Disconnections can be determined by *Simple Network Management Protocol* (SNMP) traps from the physical layer devices or by repeated interval polling of the visiting host using ARPs or pings. Because RADIUS has been widely deployed by service providers for dial-in or other networks, it is very possible that the existing accounting system would be able to support the visitor gateway if it, too, offers RADIUS.

In hotels, accounting information can be sent directly to the hotel's *Property Management System* (PMS), causing users to see an access charge on their folio. This is normally accomplished by connecting a standard low-speed serial interface between the visitor gateway and the PMS. The visitor gateway posts the charges by exchanging records with the PMS. A simple record format is used to identify a room and associated charge. Although the format and exchange protocols are usually simple, they are rarely standard. Interfacing to a PMS may require the vendor of the visitor gateway to pay a license fee to the company selling the PMS before it can implement a PMS protocol. Additionally, after implementation, the visitor gateway vendor may need to go through certification for each PMS to which the gateway will be connected. Even if the equipment vendor pays the license, service providers are rarely free to go to a hotel and attach to the hotel PMS—often the service provider is shocked that the hotel insists that they be reimbursed for "interface license fees" charged by the PMS vendor to "enable the protocol."

The 802.1x Standard and Wireless LANs

Techniques of implementing visitor networks using wireless LANs (WLANs) have been both widely publicized and debated. Wireless 802.11 "hot spots" and the like have been the subject of great publicity because these WLANs are so convenient and cost-effective to deploy that they allow service providers to economically deploy them in areas that would be impractical to serve with wired networks. However, WLANs continue to be the topic of great debate because they have been plagued by the lack of compatibility and weaknesses in security architectures.

The 802.1x standard, recently ratified by the IEEE, holds the best promise in offering a standard authentication scheme for LANs. The 802.1x standard operates with client software. In one sample scenario, the visiting host, also known as the "supplicant," receives an *Extensible Authentication Protocol* (EAP) request/identity message from the visitor network via an Ethernet switch, a WLAN access point, or a visitor gateway, any of which function as the "authenticator." The authenticator then relays the client's identification to an authentication server. The server then decides if the supplicant is to be allowed access and responds appropriately to the authenticator.

With WLANs, the *Wired Equivalent Privacy* (WEP) keys can be loaded as part of the exchange so the client and access points can operate without manual key selection. WEP has been used for several years as a method of encrypting user data over the air interface. Without WEP (or even with it, as we have seen), anyone with a laptop and a receiver can spy on the exchanged traffic[10, 11].

Microsoft ships an 802.1x client in the standard distribution of Windows XP, an important move forward in making the protocol universal. Other software vendors are shipping or have announced product for older versions of Windows, Macintoshes, Linux, and some PDAs. The 802.1x standard client implementations, however, may need firmware support on the host adapters, and support may never be available on a large number of 802.11 cards already deployed. Furthermore, all 802.1x standards are not alike because they may implement different authentication schemes. Microsoft's current implementation uses the *Extensible Authentication-Transport Level Security* (EA-TLS) protocol, which requires a *Public Key Infrastructure* (PKI)[9]. Some critics contend that this creates additional deployment burdens on organizations with small networks. If a common provider, such as Boingo or T-Mobile, provides the visitor network in "hot spots" (that is, cafés and airports), the PKI requirement should not be an issue.

On Ethernet switches, 802.1x implemented directly on the switches may be adequate if the policy for visitor access is relatively simple. For example, if users on a particular network are all trusted employees working for the same business, the work of the authentication/authorization scheme is then to determine whether or not to allow someone to access the network, simply "port on" or "port off." A more sophisticated approach would allow users to be classified as belonging to a set of classes. On some switches, 802.1x would allow each port to assume a set of VLAN memberships. For example, VLAN 120 would allow unrestricted Internet access, VLAN 119 would restrict access to a set of Web servers, and VLAN 118 would restrict access further to only an authentication server. The authentication system using 802.1x would direct the switch port configuration.

In practice, the control required by visitor networks needs to be far more flexible, and perhaps should be left to the visitor gateway. The gateway, as diagrammed in Figure 1, can control the routing system as well as higher-level protocol proxies based on policy. Besides, leaving the authentication behind the switches allows network implementors the flexibility of using virtually any Ethernet switch, or even other media such as Ethernet framing over xDSL.

The switch-based 802.1x approach, however, may have a significant advantage over the visitor gateway in that after the authentication is out of the way, the Ethernet switch can switch traffic simply at full speed without additional per-packet overhead.

Security Concerns—Better Just to Bootstrap?

Visitor networks are particularly vulnerable to hacking and snooping by virtue of their physical locations, especially if serviced by WLANs. Unfortunately, security is one of the few things that a service provider cannot deliver to visitors without their explicit cooperation and participation. The service providers face a difficult choice to either stay out of the solution or attempt to deliver adequate security through client configuration or special software distribution. The answer is difficult to determine; however, at least two factors to consider are whether the network is wired or wireless, and what the expectations of the visitors will be.

Weaknesses in WEP commonly offered on wireless LAN products have been very well publicized[10,11]. These weaknesses involve the encryption protocols and the fact that most implementations use manually configured keys. The latter is of little use on a visitor network because the network provider would need to disclose the same keys to everyone. Better proprietary systems have been deployed using PKI, and 802.1x is also a possibility. WEP may be replaced by much stronger *Advanced Encryption Standard* (AES) in *Offset Codebook* (OCB) mode as part of the IEEE 802.1i working group[12]. No solution has been both standardized and universally deployed. The lack of a standard and universal solution to replace WEP requires that the service provider who chooses another form of security customize a wireless solution. They may need to distribute specialized client software and/or restrict their service to supporting a set of wireless cards and drivers.

Simple Ethernet switches can provide some isolation between ports, but the learning bridge algorithms they use are designed to efficiently deliver Ethernet frames, not provide a secure service. With many switches, it takes one frame with a sham source MAC address to convince the switch to spill someone else's traffic onto the wrong port. "Man in the middle" attacks are often trivial after a visiting host is tricked into sending its traffic somewhere else; the opportunities of doing this to another machine on the same LAN are abundant.

As an end user of a visitor network, trusting an unfamiliar service provider in an unknown environment is a fundamentally insecure process. So, why not let the visitor network provide the basic IP connectivity in order to bootstrap the connection, and then let the visitors themselves implement the security on top? One reason is that unsuspecting users getting hacked at their favorite hotel chain does not bode well for the hotel if the incidents end up in the press. Guests probably feel pretty secure using the hotel phone for a dial-in network connection without any encryption; many also feel secure locking the door with the sliding chain.

One reasonable compromise is matching the security of a dial-in connection. A wired Ethernet, assuming that it cannot be easily coaxed to spill traffic between ports, could present an acceptable risk level. On the other hand, a poorly protected wireless network is like a hotel door without a lock.

If the visitor network offers no protection, then the burden is placed completely on the visitors to implement their own end-to-end security. Using *Virtual Private Network* (VPN) software that implements *IP Security* (IPSec) is one possibility. Unfortunately, even that is not always straightforward, given the complexities with using protocols such as IPSec over NATs[13]. Other protocols such as *Transport Layer Security* (TLS) and *Secure Shell* (SSH), which operate above the network layer, may be a better option. In addition, several proprietary VPN protocols are designed to tunnel through NATs. Those without any security solution could compromise not only their personal data but also the security of their employer's networks.

Any long-term security solution is going to demand proper client configuration and compatible software. Ultimately, development of standards and client sophistication will make this possible, but in the meantime, we will need to choose between ease of connecting to an insecure network and dealing with the potential multiple layers of authentication and encryption before gaining access. Sadly, faced with this choice and looking forward to a 7 a.m. meeting, the trusty hotel phone and modem jack on the laptop might look pretty inviting.

Summary

Visitor networks allow service providers to provide access in public places. These networks can be implemented in a way that either may or may not require specialized client software on the visiting host. Client software allows service providers to more carefully control the behavior of the visiting host but, at the same time, may limit the user base to those who have the software installed.

Visitor networks often rely on a visitor gateway to perform functions generally not required on a traditional LAN. The gateway, which shares certain characteristics with a NAS, is responsible for routing, address assignment, translation, TCP/UDP redirection, authentication, accounting, and affecting policy.

The visitor gateway exchanges packets with the visiting hosts via LANs. On Ethernet, VLANs are often best suited to visitor networks because they allow the gateway to address each client separately providing the greatest level of isolation compared to other Ethernet options.

WLANs represent an important advance toward the universal deployment of visitor networks in "hot spots." However, the lack of a common and effective solution may force service providers to choose between ease of access and security. Visitors may choose to implement a VPN or security scheme on top of the raw IP access offered by the visitor network.

References

[1] L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, R. Wheeler, "A Method for Transmitting PPP over Ethernet (PPPoE)," RFC 2516, February 1999.

[2] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, B. Palter, "Layer 2 Tunneling Protocol, L2TP," RFC 2661, August 1999.

[3] S. Glass, T. Hiller, S. Jacobs, C. Perkins, "Mobile IP Authentication, Authorization, and Accounting Requirements," RFC 2977, October 2000.

[4] IEEE Standards for Local and Metropolitan Area Networks: Port-Based Network Access Control, IEEE Std 802.1X-2001, June 2001.

[5] W. Simpson, "The Point-to-Point Protocol (PPP)," STD 51, RFC 1661, July 1994.

[6] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131, March 1997.

[7] A. Rubens, W. Simpson, S. Willens, C. Rigney, "Remote Authentication Dial-In User Service (RADIUS)," RFC 2058, January 1997.

[8] IEEE standard for local and metropolitan area networks: Virtual Bridged Local Area Networks, IEEE Std 802.1Q-1998.

[9] B. Adoba, D. Simon, "PPP EAP TLS Authentication Protocol," RFC 2716 (experimental), October 1999.

[10] N. Borisov, I. Goldberg, D. Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11," **http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html**

[11] E. Danielyan, "IEEE 802.11," *The Internet Protocol Journal,* Volume 5, Number 1, March 2002.

[12] D. Whiting, R. Housley, "AES Encryption & Authentication Using CTR Mode with CBC-MAC," Status of Project IEEE 802.11i, July 2002, **http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/2-001.zip**

[13] B. Adoba, "IPSec-NAT Compatibility Requirements," Internet-Draft, **http://www.ietf.org/internetdrafts/draft-ietf-ipsec-nat-reqts-01.txt**, March 1, 2002.

DORY LEIFER is a principal with DEL Communications Consulting, Inc. He had co-founded PublicPort in 1998, an Ann Arbor, Michigan, startup that developed one of the first visitor network gateways. After Tut Systems acquired PublicPort, he held a director of marketing position with Tut until 2001. Leifer spent 11 years with the University of Michigan and Merit Network and during that time contributed to the *Internet Engineering Task Force* (IETF). He has taught tutorials in access technologies for various seminars and tutorials, including NetWorld+Interop. He holds a B.S. in Computer Science from Rensselaer Polytechnic Institute and an M.S.E. in Industrial and Operations Engineering from the University of Michigan. Leifer currently resides in the San Francisco Bay Area and can be reached by e-mail at **leifer@del.com**

An Architecture for Securing Wireless Networks

by Gregory R.Scholz, Northrop Grumman Information Technology

Wireless networks are described as both a boon to computer users as well as a security nightmare; both statements are correct. The primary purpose of this article is to describe a strong security architecture for wireless networks. Additionally, the reader should take from it a better understanding of the variety of options available for building and securing wireless networks, regardless of whether all options are implemented. The security inherent with IEEE 802.11 wireless networks is weak at best. The 802.11 standard provides only for *Wired Equivalent Privacy,* or WEP, which was never intended to provide a high level of security[1]. For an overview of 802.11 and WEP, see reference[2]. Wireless networks can, however, be highly secure using a combination of traditional security measures, open standard wireless security features, and proprietary features. In some regard, this is no different than traditional wired networks such as Ethernet, IP, and so on, which have no security built in but can be highly secure. The design described here uses predominantly Cisco devices and software. However, unless explicitly stated to be proprietary, it should be assumed that a described feature is either open standard or, at least, available from multiple vendors.

Customer needs

Customer needs range from highly secure applications containing financial or confidential medical information to convenience for the public "hot spot" needing access to the Internet. The former requires multiple layers of authentication and encryption that ensures a hacker will not be able to successfully intercept any usable information or use the wireless network undetected. The latter requires little or no security other than policy directing all traffic between the wireless network and the Internet. Security is grouped into two areas: maintaining confidentiality of traffic on the wireless network and restricting use of the wireless network. Some options discussed here provide both, whereas others provide for a specific area of security.

The level of security required on the wireless network is proportional to the skill set required to design it. However, the difficulty of routine maintenance of a secure wireless network is highly dependant on the quality of the design. In most cases, routine maintenance of a well-designed wireless network is accomplished in a similar manner to the existing administrative tasks of adding and removing users and devices on the network. It is also assumed that security-related services such as authentication servers and firewall devices are available on the wired network to control the wireless network traffic.

It is not necessarily the case that one can see the user or device attempting to use the wireless network. This is the most alarming part of wireless network security. In a wired network, an unauthorized connected host can often be detected by link status on an access device or by actually seeing an unknown user or device connected to the network. The term "inside threat" is often used to refer to authorized users attempting unauthorized access. This is the inside threat because they exist within the boundaries that traditional network security is designed to protect. Wireless hackers must be considered more dangerous than traditional hackers and the inside threat combined because if they gain access, they are already past any traditional security mechanisms. A wireless network hacker does not need to be present in the facility. This new inside threat may be outside in the parking lot. *War Driving*[3] is the new equivalent to the traditional war dialing. All that is required to intercept wireless network communications is to be within range of a wireless access point inside or outside the facility.

Physical Wireless Network

In a highly secure environment, a best practice is to have the wireless access points connect to a wired network physically or logically separate from the existing user network. This is accomplished using a separate switched network as the wireless backbone or with a *Virtual LAN* (VLAN) that does not have a routing interface to pass its traffic to the existing wired network. This network terminates at a *Virtual Private Network* (VPN) device, which resides behind a firewall. In this manner, traffic to and from the wireless network is controlled by the firewall policy and, if available, filters on the VPN device. The VPN device will not allow any traffic that is not sent through an encrypted tunnel to pass through, with the exception of directed authentication traffic described later. With this model, the wireless clients can communicate among themselves on the wireless network, but there is no access to internal network resources unless fully encrypted from the wireless client to the VPN. This design may be further secured by configuring legitimate wireless-enabled devices to automatically initiate a VPN tunnel at bootup and by enabling a software firewall on the devices that does not allow communication directly with other clients on the local wireless subnet. In this manner, all legitimate communication is encrypted while traversing the wireless network and must be between authenticated wireless clients and internal network resources.

Authentication

Many security measures available relate to access controlled through individual user authentication. Authentication can be accomplished at many levels using a combination of methods. For example, Cisco provides *Lightweight Extensible Authentication Protocol* (LEAP)[4] authentication based on the IEEE 802.1x[5] security standard. LEAP uses *Remote Authentication Dial-In User Service* (RADIUS)[6] to provide a means for controlling both devices and users allowed access to the wireless network.

Although LEAP is Cisco proprietary, similar functionality is available from other vendors. Enterasys Networks, for example, also uses RADIUS to provide a means for controlling *Media Access Control* (MAC) addresses allowed to use the wireless network. With these features, the access points behave as a kind of proxy, passing credentials to the RADIUS server on behalf of the client. When these features are properly deployed, access to the wireless network is denied if the MAC address of the devices or the username does not match an entry in the authentication server. The access points in this case will not pass traffic to the wired network behind them. For security, the authentication server should be placed outside the local subnet of the wireless network. The firewall and VPN devices must allow directed traffic between the access points and the authentication server further inside the network and only to ports required for authentication. This design protects the authentication server from being attacked directly.

In addition to authenticating users to the wireless network, the VPN authentication and standard network logon can be used to control access further into the wired network. In this solution, the VPN client has the ability to build its tunnel prior to the workstation attempting its network logon, but after the device has been allowed on the wireless network. After the tunnel is built, specific rules on the VPN and the firewall allow the traditional network logon to occur. A robust VPN solution also treats the users differently based on the group to which they are assigned. Different IP address ranges are assigned to each group, allowing highly detailed rules to be created at the firewall controlling access to internal network resources based on user or group needs. The policy on the firewall must be as specific as possible to restrict access to internal resources to only those clients for whom it is necessary. Building very specific policy for users' access will also allow an *Intrusion Detection System* (IDS) to better detect unauthorized access attempts.

Encryption

LEAP also provides for dynamic per-user, per-session WEP keys. Although the WEP key is still the 128-bit RC4 algorithm proven to be ineffective in itself[7], LEAP adds features that maintain a secure environment. Using LEAP, a new WEP key is generated for each user, every time the user authenticates to use the wireless network. Additionally, using the RADIUS timeout attribute on the authentication server, a new key is sent to the wireless client at predetermined intervals. The primary weakness of WEP is due to an algorithm that was easy to break after a significant number of encrypted packets were intercepted. With LEAP, the number of packets encrypted with a given key can be tiny compared to the number needed to break the algorithm.

When using LEAP for user and device authentication, WEP encryption is automatically enabled and cannot be disabled. However, if added security is needed, a VPN, as described earlier, can provide any level of encryption desired. Using a VPN as the bridge between the wired and wireless network is recommended regardless of the underlying vendor or technology used on the wireless network. *IP Security* (IPSec) is a proven, highly secure encryption algorithm available in VPNs. By requiring all wireless network traffic to be IPSec encrypted to the VPN over the WEP-encrypted 802.11 Layer 2 protocol, any data passed to and from wireless clients can be considered secure. All traffic is still susceptible to eavesdropping, but will be completely undecipherable.

Aside from WEP and LEAP, some vendors provide other forms of built-in security. Symbol Technologies' Spectrum24 product provides Kerberos encryption when combined with a Key Distribution Center. Kerberos is more lightweight than IPSec and, therefore, may be better suited to certain applications such as IP phones or low-end *personal digital assistants* (PDAs). Other methods of automating the assignment and changing of WEP keys are also available, such as Enterasys' Rapid-Rekey[8]. Wireless vendors have realized that security has become of critical importance and most, if not all, are working on methods for conveniently securing wireless networks. When available, most vendors seemingly prefer to use open-standard, interoperable security mechanisms with proprietary security being additionally available.

Bringing it all together

Numerous options are available to secure a wireless network. A highly secure design will include, at a minimum, an authentication server such as RADIUS, a high-level encryption algorithm such as IPSec over a VPN, and access points that are capable of restricting access to the wireless network based on some form of authentication. When all the security options are tied together, the wireless network requires explicit authentication to allow a device and the user on the wireless network, the traffic on the wireless network is highly encrypted, and traffic directed to internal network resources is controlled per user or group by an access policy at the firewall or in the VPN.

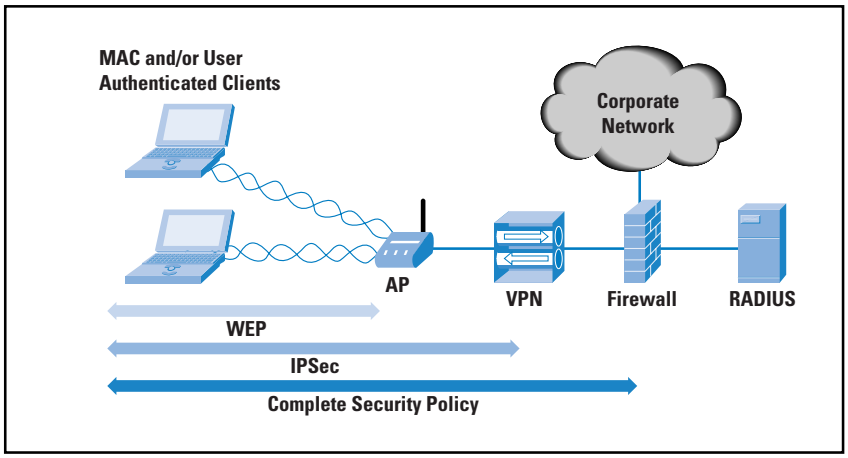


Figure 1: A Highly Secure Wireless Network

There is no substitute for experience and research when designing a network security solution. Using network security and design experience to exploit available technologies can further increase security of a wireless network. For example, grouping users into IP address ranges based on access requirements allows firewall access policy to help restrict unnecessary access. This can be accomplished using *Dynamic Host Configuration Protocol* (DHCP) reservations, assigning per-user or -group IP address ranges to the VPN tunnels or statically assigning addresses. Using a centralized accounts database for all authentication helps avoid inadvertently allowing an account that has been disabled in one part of the network to access resources through the wireless network. To use an existing user database for authentication while providing for dynamic WEP keys, use a LEAP-enabled RADIUS server that has the ability to query another server for account credentials. As with most network designs, a solid understanding of the available technologies is paramount to achieving a secure environment.

Utilizing all the security described in this article would yield the following design. When a device first boots up, it receives an IP address within a specified range on a segregated portion of the network. This IP range is based on the typical usage of the device and is most useful for machines dedicated to specific applications. As a user attempts to log onto a wireless device, a RADIUS server authenticates both the MAC address and the username of the device. If the user authentication is successful, access is granted within the wireless network. In order for traffic to leave the wireless network to access other network resources, a VPN tunnel must be established. Again, the IP address assigned to the tunnel can be controlled based on individual user authentication to help enforce access policy through the firewall. When the tunnel is established, firewall access policy will restrict access to resources on the network. Most, if not all, of the authentications required may be automated to use a user's existing network logon and transparently complete each authentication. This is not the most secure model, but it would be as secure as any single signon environment.

Summary

A secure wireless network is possible using available techniques and technologies[8] [9] [10]. After researching needs and security requirements, any combination of the options discussed here, as well as others not discussed, may be implemented to secure a wireless network. With the right selection of security measures, one can ensure a high level of confidentiality of data flowing on the wireless network and protect the internal network from attacks initiated through access gained from an unsecured wireless network. At a minimum, consider the current level of network security and ensure that the convenience of the wireless network does not undermine any security precautions already in place in the existing infrastructure.

References

[1] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Standard 802.11, 1999 Edition.

[2] "802.11," Edgar Danielyan, *The Internet Protocol Journal,* Volume 5, Number 1, March 2002.

[3] "War Driving," Andrew Woods, **http://www.personaltelco.net/index.cgi/WarDriving**, last viewed August 11, 2002.

[4] "Cisco Aironet® Product Overview," Cisco Systems, **http://www.cisco.com/univercd/cc/td/doc/product/wireless/airo_350/350cards/pc350hig/pc_ch1.htm,** last viewed August 11, 2002.

[5] "IEEE Standard for Local and Metropolitan Area Networks—Port-Based Network Access Control," IEEE Standard 802.1X, 2001.

[6] "Remote Authentication Dial-In User Service," C. Rigney, S. Willens, A. Rubens, and W. Simpson, IETF RFC 2865, June 2000.

[7] "Security of the WEP Algorithm," Nikita Borisov, Ian Goldberg, and David Wagner, **http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html,**, last viewed August 11, 2002.

[8] "802.11 Wireless Networking Guide," Enterasys Networks, June 2002, **http://www.enterasys.com/support/manuals/hardware/4042_08.pdf,** last viewed August 11, 2002.

[9] "Wireless LAN Security in Depth," Sean Convery and Darrin Miller, Cisco Systems, **http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/safwl_wp.htm,** last viewed August 11, 2002.

[10] "Making IEEE 802.11 Networks Enterprise-Ready," Arun Ayyagari and Tom Fout, Microsoft Corporation, May 2001, **http://www.microsoft.com/windows2000/docs/wirelessec.doc,** last viewed August 11, 2002.

GREGORY SCHOLZ holds a BS in Computer and Information Science from the University of Maryland. Additionally, he has earned a number of certifications from Cisco and Microsoft as well as vendor-neutral certifications, including a wireless networking certification. After serving in the Marine Corps for six years as an electronics technician, he continued his career working on government IT contracts. Currently he works for Northrop Grumman Information Technology as a Network Engineer supporting Brook Army Medical Center, where he performs network security and design functions and routine LAN maintenance. He can be reached at: **gscholz@wireweb.net**

Opinion: The ISP—The Uncommon Carrier

by Geoff Huston, Telstra

There is a long-standing role in the communications industry where a provider of public carriage services undertakes the role of a *common carrier*. What's so special about the role of a common carrier, and why is this role one that is quite uncommon in the *Internet Service Provider* (ISP) world?

Side comment: There once was a time when you could not trust the messenger. There once was a time when not only did you pay to have your message sent, but you paid to *receive* messages. And there was no guarantee that the message would not be read by the messenger. The contents of your note could have been used to determine how much the receiver should pay for the message. Your message could have been copied and sold to other parties. If you can't trust the messenger, then communications becomes a risky business.

The Messenger

Throughout history the position of a messenger has been a mixed blessing. To be the bearer of bad news was not an enviable role, and rather than being rewarded for the effort of delivering the message, the messenger might have been in dire straits, given the level of wrath of the recipient. The option of reading the message before delivering it could be seen as a personal survival strategy, as well as being a prudent business move—bad news could be discarded immediately, whereas good news could have the potential of extracting a higher delivery fee from the recipient. Although this scenario would have been good for the messenger, such a mode of operation was not beneficial to all. For the parties attempting to use the messenger service, message delivery could be a very haphazard affair. The message might or might not get delivered, the delivery time was variable, as was the cost of delivery, and if the message itself was intended to be a secret, then one could confidently anticipate that the messenger would compromise this secrecy.

The Common Carrier

For a communications network to be truly useful, numerous basic attributes must be maintained. These include predictability, so that a message passed to a communications carrier is delivered reliably to the intended recipient. Integrity is also necessary, because a message must not be altered by the carrier in any way. Privacy is also an essential attribute, because the message must not be divulged to any party other than the intended recipient, nor should even the existence of the message be made known to any other party. And above all there must be a solid foundation for trust between the carrier and the clients of the service. So in this form of social contract, what does the carrier get in return?

Apart from payment for the service, the carrier is absolved from liability regarding the content of the messages, and from the actions of the customers of the service. This form of social contract is the basis for the status of a common carrier.

It may have taken some time, but this role is well understood by the public postal network. And as many national postal operators encompassed the role of national telephone carrier, the common carrier role has been an integral part of the public telephone network.

The ISP's Role

But in the world of the ISP the position of common carrier is very uncommon indeed.

There once was a time when folk did not need to encrypt their letters nor speak in scrambled code to undertake a private conversation. The assumption, made law in many countries, was that the entity entrusted with public communications, the common carrier, was barred from deliberately inspecting the contents of the plain transmission, and various dire penalties were in place if a public carrier's employees or agents divulged anything they may have learned by virtue of being public carriers. Various measures were put in place to execute interception and monitoring, but these measures required due process and reference to some law enforcement agency and also the judiciary to ensure that the rights of the public user were adequately safeguarded.

The issues of the role of a common carrier and the current role of an ISP are clearly seen when looking at the reactions to unsolicited commercial e-mail, or spam. Every day ISPs receive strident demands of the form: "One of your users is sending unsolicited messages—disconnect him now!" Internet users are, in effect, holding the ISP responsible for the actions of its customers. A similar expectation of the ISP's responsibility for the actions of its customers is seen in response to various forms of hacking, such as port scanning. Similar messages are sent to ISPs, demanding the immediate disconnection of those customers who are believed to be originating such malicious attacks. From a small set of complaining messages some years back, the volume of such demands for ISP action is now a clamor that is impossible for any ISP to ignore.

What should the ISP do? Many responsible ISPs see it as appropriate to conduct an investigation in response to such complaints. ISPs often include provisions in their service contracts with their customers to allow them to terminate the service if they believe that their investigation substantiates the complaints on the basis of a breach of contract. When disconnected, such customers are often blacklisted by the ISP to ensure that they cannot return later and continue with their actions. Surely this is an appropriate response to such antisocial actions?

This may be the case, but it is not necessarily consistent with the role of the ISP as a common carrier. A common carrier is not a law enforcement agency, nor is it an agent of the judiciary. It may be entirely appropriate for a common carrier to investigate, under terms of strict privacy, a customer's activities and inspect the contents of traffic passed across the network if it has reasonable grounds to suspect that the integrity of the network itself is under threat. Equally, it is probably inappropriate for a common carrier to extend the scope of such investigations on the basis of external allegations of activities that are not related to the integrity of the service itself.

The assumption that an ISP is, in some way, responsible for the actions of its customers has been extended further in some countries, such that the ISP is, in part, responsible for the content carried over its network, including content that originates with a customer of its service. This expectation that ISPs should actively control and censor content passed across their network is not just an expectation of some Internet users. This expectation appears in numerous legislative measures enacted in many countries. The *Communications Decency Act* in the United States legislature is an example of such an expectation of the active role of the ISP in controlling content passed across its network.

Who Will You Call?

Perhaps the issue here is one of expediency. Where can a user direct a complaint after receiving yet another piece of unsolicited, and possibly highly offensive, e-mail, apart from the ISP of the sender of the message? Where else can users direct a complaint after being the subject of yet another port scan of their system, but to the ISP? And what else can an ISP do in response? The ISP often has little choice but to investigate such complaints in good faith, and take corrective action if the complaint is substantiated. In the absence of any effective regulatory framework that would allow such investigations to be undertaken by an appropriate external agency, the ISP is in a difficult position.

Whereas it may be the correct common carrier position to disclaim all responsibility for the actions of its customers together with the content passed across its network, to ignore such complaints marks the ISP as a haven for such antisocial activities. Adopting such a position often has a negative impact on the ISP's ability to interconnect with other ISPs, because ISPs also tend to hold each other responsible for the actions of their customers and the content passed across their network. ISPs tend to avoid extending interconnection services to those ISPs that disclaim any such responsibility. So the expedient response is for the ISP to assume some level of responsibility for its customers and the content of its network and act accordingly.

But short-term expedient measures should not be confused with long-term effective solutions. The problem with these short-term responses lies in the uniquely privileged position of the carrier. Even rudimentary forms of data mining of each customer's communications patterns and the content of their communications can yield vast quantities of valuable information. Such information can allow a carrier to discriminate between customers, compromise the integrity of the customer's use of the network, and actively censor the content passed across the network. Positions of privilege without accompanying checks and balances are readily abused. There is already the widespread expectation and acceptance that an ISP has the ability and duty to inspect network content and monitor customers' activities with respect to various forms of antisocial and often malicious activities. But how can checks and controls be enforced such that the information gained through such monitoring activities is not used for other purposes? Such monitoring is not without cost, and the option of recouping some revenue to balance this expenditure by regarding this information as a business asset is always present. The regulatory impost of a common carrier role is intended to be an economically efficient response to this issue. The common carrier role is intended to reduce the social power of public carriers and protect the public's open, uncensored, and equal access to the carrier's services.

It is often said that the road to hell is paved with the best of intentions—that the ultimate outcome of the solution is potentially far worse than the immediate problem being addressed. The ultimate outcome of erosion of the common carrier role is that public users of a public communications service can confidently expect their communications to be monitored, potentially stored and cross referenced, and possibly later acted on.

Public Policy

Today the short-term expedient measures abound. There is enormous pressure on ISPs from both the Internet's user base and numerous legislatures to take an active position of being responsible—and liable, for the content on the networks and the actions of their clients. If left unchecked, this will have severe longer-term consequences for free speech, basic personal privacy, and uncensored, nondiscriminatory, universal access to the Internet. And when the user base comes to recognize the debased value of such a compromised communications system, they will inevitably look to other means of communication that have retained their essential integrity as a common carriage service.

Perhaps it is time for the debate regarding the role and responsibilities of an ISP to be placed on the agenda of public policy makers. Perhaps it is time to recognize that ISPs are indeed common carriers, and that they have a clearly bounded set of responsibilities with respect to both content and the actions of clients of the service.

Perhaps it is time to consider how best to enforce social norms on the Internet without compromising the basic integrity of the carrier as a neutral party to the content being carried across the network. Perhaps it is time to recognize that in this domain the Internet is not entirely novel, and what we have learned from a rich history of carriage provision in society has direct relevance to the Internet today.

The Internet is simply too valuable a communications service to have its long-term potential as a universal communications service mindlessly destroyed on the altar of short-term expediency.

Disclaimer: I am by profession neither a lawyer nor a public policy maker. However, by virtue of working in the ISP industry, I have an increasing level of interest in the activities of these folk, for the reasons outlined above. I should also note that personal opinion comes in many forms. The above is one such form.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Scientist in the Internet area for Telstra. He is also a member of the Internet Architecture Board, and is the Secretary of the APNIC Executive Committee. He is author of *The ISP Survival Guide*, ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks*, ISBN 0471-378089, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons.
E-mail: **gih@telstra.net**

Letters to the Editor

ENUM

Ole,

I was looking at the June 2002 issue of *The Internet Protocol Journal,* and noticed what might be a misprint. In the story on ENUM, the next-to-last paragraph on page 21 has a sentence reading:

> North America has the .164 country code of "1," implying that under ENUM there is a single DNS domain for ENUM, namely **`1.e164.arpa.`**

I suspect it should read "... there is a single DNS domain for North America..." or something like that. (The ".164" should probably also be "E.164"—you don't refer to it as just ".164" elsewhere in the article.)

A more substantive comment on Marshall Rose's BEEP article in the same issue: It was a good overview, but I would have liked to see a mention of which application protocols are likely to use BEEP (assuming that none has already) in the near future. The middle of page 11 explains why the IETF thinks this is a good idea and why new application protocols need BEEP, but it was hard to tell whether it actually is being actively considered for use by any IETF working group.

Overall, I liked the issue, and particularly Peter Salus's review of Padlipsky's book—I came across it in the late 1980s, and actually met Michael sitting in a hallway at one of the Interop conferences before they got too big for Silicon Valley and I stopped attending. I still remember some of his cartoons and slogans (e.g., something to the effect "... the ITU is planning to have an 11-layer model because it's a sacred number in Bali..."). I've also found the articles in some of the other recent issues of the IPJ—e.g., the articles on wireless LANs (particularly the discussion of security issues) and code signing/mobile code in the March 2002 issue—very helpful, and have pointed colleagues to them.

Best wishes.

—Eric M. Berg
Managing Director,
Technology Forecast Publications
PricewaterhouseCoopers Technology Centre
`Eric.Berg@us.pwcglobal.com`

Geoff Huston responds:

While we all try hard to eliminate various errors in manuscripts prior to publication, there are always a few author-mishaps that manage to sneak past the eagle eyes of the editor, and this is one of them.

The offending sentence should read:

> North America has the .E164 country code of "1," implying that under ENUM there is a single DNS domain for ENUM in North America, namely **`1.e164.arpa.`**

Thanks for pointing this out.

—Geoff

More about ENUM

Ole,

In the June 2002 issue of IPJ (Volume 5, Number 2), Geoff Huston wrote an interesting article about ENUM. The technical side of ENUM (using DNS to map E164 numbers to services) seems rather straightforward. But its implications on both technical and social issues are much more complex and (in my opinion) interesting. I am not an expert on the subject, but I'd like to share a few thoughts about this. First, two technical issues come to mind.

The first one is about the use of the *Domain Name System* (DNS). The DNS has been very successful as a distributed replicated database of hostname-to-IP address (and reverse) mappings. Will it be able to handle gracefully all the stuff people intend to put in it? This is not certain, as shown by ICANN's cautious attitude concerning the creation of new Top Level Domains. Content Distribution Networks, for example, often use lots of domain names with short TTLs, reducing the effectiveness of DNS caching (Geoff mentions this caching issue for ENUM). After all, DNS stands for "Domain Name System," not "General Purpose Infinitely Scalable Distributed Dynamic Database."

The second issue is about the status of addresses and names in the Internet. Simplifying things, we can say the following happens when somebody wants to access an Internet service with an E.164 number: The E.164 number is translated into a DNS name, and a DNS lookup gives back an URI. If the URI is a simple URL, the domain name in the URL is DNS-looked-up for an IP address, and then packets are sent to that IP address. If the URI is not a simple URL (such as a URN), some other resolving process implying the DNS occurs anyway.

That makes two levels of indirection, but, moreover, creates an "interesting" situation: IP addresses are "addresses," i.e., network-friendly identifiers, whose structure is tied to the network topology.

Such identifiers are not user friendly, so user-friendly identifiers called "names" have been created, and a "domain name system" set up to translate names into addresses. E.164 numbers are really telephone addresses. They are tied to the telephone network topology and are surely not user friendly. There are no user-friendly names in the telephone system.

The strange thing is that with ENUM, E.164 numbers are not linked anymore to the network topology, but rather become names intended for user usage. In a sense, they even are "meta names," since they translate to DNS names (that translate to addresses). But they obviously have not become user-friendly in the process.

I must admit I oversimplify a bit since I don't distinguish between names and addresses identifying level 3 (network) resources (i.e., hosts) and those identifying level 7 (application) resources (e-mails, Web pages, etc.), but this doesn't invalidate the idea.

Addresses are what the network needs, and names are what the users need. This brings me to the politics aspects of ENUM: who administers/controls/owns the namespace? A namespace is only partly technical; defining a namespace includes defining how and by whom the namespace is operated. The DNS is technically a big success, but the politics side is controversial, as shown by domain-name disputes or the setting up of alternative domain-name systems. It seems that social aspects are often more difficult to deal with than technical issues are to solve.

When I was studying networking we were taught how the technical differences between the Internet and the telephone network took their roots into a fundamental difference of culture. Now that the Internet culture seems to have won on the technical aspect (IP over broadband ISDN), wouldn't it be a strange outcome for the Internet namespace to be owned by telephone companies?

To conclude, I think this ENUM stuff shows that the Internet community really needs to work on the namespace issue, to ensure a technically and socially sound namespace for the Internet.

—Christophe Deleuze, Ph.D.
R&D Senior Engineer
ActiVia Networks
Christophe.Deleuze@ActiVia.net

Book Review

Carrier-Scale IP Networks

Carrier-Scale IP Networks: Designing and Operating Internet Networks, edited by Peter Willis, ISBN 0-85296-982-1, The Institute of Electrical Engineers, London, United Kingdom, 2001

My heart jumped when I saw the nondescript brown box, about the thickness of a book, sitting by the receptionist. It was finally here! I had waited almost two months in great anticipation for this book to show up. Was it going to be the all-encompassing handbook for the network designers, operators, and managers in large-scale IP environments? The first few lines in the text indicated that it just might be: "The aim of this book is to give the reader an understanding of all the aspects of designing, building and operating a large global IP network."

The definition of "large-scale" as given by the author and for the purposes of this review follows: Provides services for millions of end users, high-speed (greater than 100 Mbps) transit services, and is reliable, scalable, and manageable.

One thing to keep in mind is the way this book was constructed. The 16 chapters had 29 authors. Almost all authors came from some area of British Telecom (BT) and all were subject matter experts in the chapter they wrote. The 16 chapters are grouped roughly into four sections: Designing and building IP networks, transmission and access networks, operations, and development of future networks. Sadly, all of this is squeezed into 293 pages.

Designing and building IP networks

For the reader new to designing and building large-scale IP networks, the first few chapters are gold. For the reader already experienced in this area, it may bring back nostalgic feelings for the good old days of exponential growth. A lot of ground is covered, including the obligatory overview of IP, sufficient enough to give a nontechnical person the key concepts of IP routing, but can by skipped by those with even basic knowledge in this area. The examples given throughout this chapter (and the rest of the book) come directly from the design of BT's and Concert's backbone. A whole chapter, "The Art of Peering," not to be mistaken for an excellent paper of the same name[1], gives excellent key concepts in peering. Some coverage is even given to the logistics and difficulties in building points of presence globally, going so far as to mention earthquake bracing for equipment bays.

The next set of chapters give the reader detail about the transmission network (for some, be prepared to think *Synchronous Optical Network* [SONET] when you read *Synchronous Digital Hierarchy* [SDH]), and access networks, including various forms of broadband, wireless, dial, and satellite.

The technical information was squeezed into these chapters, not enough for a good technical treatise, but enough to give readers good grounding in a technology that is unfamiliar to them. The coverage was closer to being marketing material. These chapters alone are not enough to bring those new to the field up to speed if they are to design or operate such a network.

BT opened itself up and gave us a view into the operations of its network. Individuals who have worked in an environment like this will find something familiar. We get to see how BT structures the people, processes, and technologies. This is something that is not usually open to inspection by people outside of an organization. Planning and developing the operations side of the house is a difficult job. These chapters may give a kick-start to those coming into such a role.

I was disappointed with the two final chapters. Of course anything listed as being "the future" will one day become the present, but I digress. These two chapters seem like the odd couple that just did not fit with the rest of the chapters. The first chapter is on Traffic Engineering. It is really a primer on *Multiprotocol Label Switching Traffic Engineering* (MPLS TE). The second chapter covers *Virtual Private Networks* (VPNs), both the MPLS and *IP Security* (IPSec) types.

Recommendation

The authors set out with a lofty goal, and did not quite hit the mark. This book would be appropriate for someone trying to get a feel for what goes on inside of a carrier-scale network. People already in the business would be better served by just paying attention to what goes on around them.

Perhaps a small focused group could set out to create a book (or should I say tome) covering the elements of design, the foundation of support, and the basics of management. Something timeless is required here, independent of the protocol du jour, to develop the next generation of competent netheads.

—Kris Foster
kris.foster@telus.com

[1] "The Art of Peering: The Peering Playbook," William B. Norton, Equinix

Fragments

Stephen Wolff receives Postel Service Award

In June 2002, Internet pioneer Stephen Wolff was honored by the *Internet Society* (ISOC) for his significant contributions on behalf of the Internet. A founding member of the ISOC, Wolff is considered one of the "fathers of the Internet" and was directly involved with its development and evolution.

Wolff received the *Postel Service Award,* named for Dr. Jonathan B. Postel, an Internet pioneer and head of the organization that administered and assigned Internet names, protocol parameters, and *Internet Protocol* (IP) addresses. He was the primary architect behind what has become the *Internet Corporation for Assigned Names and Numbers* (ICANN), the successor organization to his work. The recipient of the award receives a $20,000 cash honoraria.

"We are pleased to recognize Steve with the Postel Award," said ISOC President/CEO Lynn St.Amour, "especially as his contributions are well know to ISOC, having previously been commended by ISOC's board for helping transform the Internet from an activity serving the particular goals of the research community to a worldwide enterprise which has energized scholarship and commerce in dozens of nations."

The 1994 commendation from the ISOC board also states that "The personal leadership of Dr. Wolff, often under conditions of public controversy, has been an indispensable ingredient in surmounting a daunting array of technical, operational and economic challenges. His extraordinary commitment to the growth and success of the Internet reflect the highest standard of service to the networking community and command our respect and admiration."

As Director of the Division of Networking and Communications Research and Infrastructure at the US National Science Foundation, he was responsible for NSNET, the *National Research and Education Network* (NREN), and for NSF's support of basic research in networking and communications. While at the NSF he was among the founders of the interagency and international research networking management and advisory structure whose descendants today include the Large-scale Networking (LSN) working group and the PITAC.

Wolff left the federal government and joined Cisco Systems, Inc. in 1995, where he works in the University Research Program—Cisco's program supporting academic investigators with unrestricted grants for research on computer networks.

Wolff was educated at Swarthmore College, Princeton University, and Imperial College. He taught electrical engineering at the Johns Hopkins University for ten years and subsequently spent fifteen years leading a computing- and network-related research group at the U.S. Army Research Laboratory. In 1983 he took a sabbatical half-year as a Program Director in the Mathematics Division of the U.S. Army Research Office.

ISOC is a not-for-profit membership organization founded in 1991 to be the international focal point for global cooperation and coordination in the development of the Internet. Through its current initiatives in support of education and training, Internet standards and protocol, and public policy, ISOC has played a critical role in ensuring that the Internet has developed in a stable and open manner. For 10 years ISOC has run international network training programs for developing countries which have played a vital role in setting up the Internet connections and networks in virtually every country that has connected to the Internet. For more information, please visit: **`http://www.isoc.org/`**

ISOC to Run `.org`?

Recently ICANN posted a preliminary Staff Report on the selection of a new registry operator to assume responsibility on January 1, 2003 for the **`.org`** registry. The report, which is subject to public comment and comment by all the bidders before being submitted for approval to the ICANN Board of Directors, recommends that the Board select the *Internet Society* (ISOC) as the successor registry operator for the **`.org`** registry, currently operated by VeriSign.

This preliminary report follows an extensive bid solicitation and evaluation process that was launched last April. Eleven bids were received in response to a Request for Proposals. These bids were analyzed and evaluated by three evaluation teams that operated independently of each other.

"We received eleven very strong and thoughtful proposals," noted Stuart Lynn, President of ICANN. "We appreciate the response of the institutions behind these proposals. The ISOC proposal was the only one that received top ranking from all three evaluation teams. On balance, their proposal stood out from the rest." Lynn also emphasized the openness and transparency of the solicitation and evaluation process.

Two evaluation teams focused on technical issues: one from Gartner, Inc., an international consulting and research organization that specializes in information technologies, and the other a team mainly composed of CIOs of major universities. Another team was provided by ICANN's *Non Commercial Domain Name Holders* constituency; the NCDNHC team focused on the effectiveness of the proposals to address the particular needs of the **`.org`** registry. The staff report integrates these evaluations and other factors into the preliminary recommendation.

ISOC is an international not-for-profit organization of over 6,000 individual and 150 organizational members with chapters in over 100 countries. It provides leadership in addressing issues that confront the future of the Internet, as well as being a home for the *Internet Engineering Task Force* (IETF) and the *Internet Architecture Board* (IAB).

In operating the **.org** registry, ISOC will team with Afilias, an operating registry that recently launched the **.info** *top level domain* (TLD) that was authorized by ICANN as one of seven new TLDs over this past year.

"Afilias will provide ISOC with the necessary experience at operating a large registry," said Lynn. "The **.info** registry already houses about 1 million domain names, which is on a scale that approaches the much older **.org** registry."

ICANN is re-assigning the **.org** registry under a revised agreement among ICANN, VeriSign, and the U.S. Department of Commerce that was signed in May 2001. Under that agreement, VeriSign was permitted to keep its registrar business, NSI (that it was obligated to sell under the prior agreements) provided that it agreed to relinquish **.org** at the end of December 2002, and subject to other provisions of the revised agreements. As part of those revised agreements, VeriSign agreed to endow the new operator with US$ 5 million to help fund operating costs, provided that the new operator was a not-for-profit organization.

Following an open and transparent process, ICANN has posted all eleven applications online together with all supplemental material and community comments received. The preliminary staff report and the evaluations are posted at:
http://www.icann.org/tlds/org/preliminary-evaluation-report-19aug02.htm.

Applicants and any member of the community are invited to send comments on the preliminary report and evaluations by e-mail to:
org-eval@icann.org

Upcoming Events

The *IETF* will meet in Atlanta, Georgia, USA, November 17–21, 2002.
http://www.ietf.org/meetings/meetings.html

ICANN will meet in Shanghai, China, October 27–31, 2002.
http://www.icann.org/meetings/

The next *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will take place February 19–28, 2003 in Taipei, Taiwan. **http://apricot2003.net/**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

December 2002 Volume 5, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

In December 1999 we published Part One of a two-part article on Internet Multicast. Some readers have asked "what happened to Part Two?" Finally, in this issue we are able to bring you the second article, "Internet Multicast Tomorrow." Multicast remains a technology with limited Internet-wide deployment, but numerous research activities are underway that may change this situation. Ian Brown, Jon Crowcroft, Mark Handley and Brad Cain provide an overview of current developments in multicast.

If all computer networking was a simple matter of "plug-and-play," I suppose this journal would not exist. Nevertheless, it is encouraging to see developments that aim to simplify configuration of network devices, particularly those that move around a lot. The Zeroconf working group of the *Internet Engineering Task Force* (IETF) has been developing standards for "configuration-free" networks. Edgar Danielyan explains the details in our second article.

We continue to receive numerous letters in response to our articles. Your feedback is very much appreciated, because it helps us develop material for future issues. Please keep your letters coming to **ipj@cisco.com**

The long-awaited online subscription system is now ready for deployment and you will be able to try it out in the very near future at **www.cisco.com/ipj**. With this system, you can update your mailing address as well as select delivery options, online notification of new issues and so on. As with any computer based system, I anticipate that we, with your help, will uncover a few bugs. Please report any problems you may encounter to **ipj@cisco.com**.

A new important resource is available from the *Internet Society* (ISOC). *The Internet Report* is a catalogue of IETF documents, including RFCs and Internet Drafts, that document the technology, protocols and operating procedures that form the Internet. The report includes RFCs, IETF Working Group drafts as well as individual drafts. The Internet Report is maintained by Geoff Huston. You can access the report online at **http://ietfreport.isoc.org/**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Internet Multicast Tomorrow

by Ian Brown, University College London,
Jon Crowcroft, University of Cambridge,
Mark Handley, ICIR,
Brad Cain, Storigen Systems

This article is part of a pair, the first of which looked at the state of play in IP multicast routing[0]. In this article, we look at the broader problems and future activities with multicast. We divide the areas into routing, addressing, transport, security, operations, and research.

There has been quite a bit of debate about the nature of compelling applications for multicast recently.[44] It is certainly the case that we do not completely understand the "market" for multicast—this is at least in part because multicast does not yet provide a complete set of functions for all the applications and services we might imagine. This is a typical "chicken and egg" situation, though: To put an extreme version of the argument, the application writers do not see any multicast deployed; the *Internet Service Providers* (ISPs) do not see any multicast applications; and the router vendors do not see any multicast service demand from ISPs. (The same problem afflicts IPv6, Integrated and possibly Differentiated Services, and mobile IP, of course.)

As we discussed in the part I of this article[0], this situation has been somewhat alleviated by streaming applications for audio and video from the classical content providers in the entertainment and news industries. And although we are still seeing some problems, we are also seeing broader interest and development.

The next section presents recent work on routing and addressing. After that we look at transport. Subsequently, we discuss security. Then we look at operations and management. Finally, we examine some of the research ideas that are available.

Routing and Addressing

The single biggest step recently in multicast routing and addressing has been the recognition that the demand for large-scale multicast is largely for one-to-many or single source. Combined with the ability to select sources at the receiver (as a means to prevent denial-of-service attacks) in the *Internet Group Management Protocol* (IGMP)v3, this has made a significant improvement to ISPs' willingness to deploy the service[42].

Source-Specific and Single-Source Multicast

The origins of the idea were thesis work at Stanford by Hugh Holbrook on Express multicast[43]. This is a specialized multicast architecture for one-to-many multicast groups. In this way, Express is a subset of the current multicast model in that it allows only a single sender to a multicast group. The advantages of Express are that certain aspects of multicast routing and addressing are easier solved by ignoring the many-to-many case. Many feel that the most likely large-scale applications of multicast are one-to-many, a fact that explains why Express is becoming popular as a short-term solution.

Express addresses are *channels* that are 64-bit addresses (that is, source address plus group address). Express sources transmit to a channel and advertise that channel. Receivers learn about these channels through advertisements or through other means (that is, URL) and initiate an Express join. Routers propagate these joins directly toward the source, building a source rooted multicast forwarding tree.

The Express model offers two primary benefits. First, Express simplifies the complexity of multicast routing. Secondly, Express simplifies the assignment of multicast addresses for IPv4. Because Express channels are 64 bits, a source can select any lower 32 bits (any group address) for its channel and not collide with another.

In order to implement Express with IPv4 multicast protocols, a special range of multicast addresses was defined. The 232/8 address has been allocated by the *Internet Assigned Numbers Authority* (IANA) for single-source multicast experimentation. In this range, an address has meaning only when "coupled" with a source address. Another way to explain it is that this address range is reserved for the lower 32-bit Express addresses. With this scheme, Express requires no modification to multicast data packets.

Express can be implemented with two protocols that have already been developed: IGMPv3[42] and *Protocol Independent Multicast Sparse Mode* (PIM-SM).

IGMPv3 extends IGMP to allow source-specific joins to a multicast address. This capability can be used to carry 64-bit (S,G) joins to a router. When a router receives the IGMPv3 join, it must be able to build the source-specific tree with a multicast routing protocol. PIM-SM, widely deployed in service provider networks, already possesses this capability. The combination of IGMPv3 and PIM-SM allows Express to be implemented without creating more protocols; this is one of the most powerful benefits of the Express model.

Interdomain Multicast

Currently there are four fairly widely deployed multicast routing protocols: *PIM Dense Mode* (PIM-DM), PIM-SM or *Source-Specific Multicast* (SSM), *Multicast OSPF* (MOSPF), and the *Distance Vector Multicast Routing Protocol* (DVMRP). Because of the different properties of these protocols, there are many difficulties in connecting heterogeneous routing domains together[38]. In general, most problems arise when connecting explicit join type protocols with flood-and-prune protocols. With service providers rolling out multicast using PIM-SM, connecting DVMRP and PIM-DM flood-and-prune is becoming common.

In order to connect two multicast routing domains, a *Multicast Border Router* (MBR) needs to exist between the two domains. This router must implement a shared forwarding cache architecture[39]. In this model, each multicast routing protocol running on a MBR submits its forwarding cache entries to a shared cache. This cache is the "bridge" between the trees in the different domains.

In order that the appropriate trees are created in each domain (on either side of a MBR), signaling must exist to bring sources from one domain to receivers in the other domain. This is part of the complication in connecting flood-and-prune protocol domains to explicit join protocol domains. In an explicit join protocol such as PIM-SM, joins are sent by edge routers to either a source or a *Rendezvous Point* when a host joins. A flood-and-prune protocol works quite differently, in a sense assuming that packets are desired; trees are pruned when edge routers receive new source packet but have no local listeners.

The signaling aspect of joining two domains can be accomplished with a variety of means. There are many options, but two stand out as providing the best methods of connecting domains. The first is to use *Domain Wide Reports* (DWRs)[36] in flood-and-prune domains. DWRs are similar to IGMP reports except that they are sent on a domain-wide basis. When a border router receives a DWR report, it can join a group on behalf of an entire domain. The second solution is to use the *Multicast Source Discovery Protocol* (MSDP)[37]. MSDP is currently used to send source lists between PIM-SM domains. It can also be used to connect domains by having the MBR also participate in MSDP. Sources can then be learned from an explicit join protocol domain; the MBR can then join the sources and flood them into attached flood-and-prune protocols domains.

Address Allocation

The schemes to provide dynamic distributed address allocation have not been successful to date. But with many multicast services being limited to either a single domain or a single source, the pressure is off. Instead, source-specific addresses are unique in any case. For many-to-many multicast (sometimes known as *Internet Standard Multicast* [ISM]), the problem has also been alleviated by the use of GLOP[61], which allocates sections of the address space by mapping Autonomous System numbers of a provider into Class D prefixes. This is potentially inefficient, but solves the contention, collision, revocation, or resolution problem that *Multicast Address Set Claim* (MASC) and *Multicast Address Allocation* (MALLOC)[60] attempt to do in a distributed dynamic manner.

In the longer term this address allocation, as well as scalable solutions to many-to-many multicast in the local domain and interdomain, await further development on bidirectional trees ["Bi-dir PIM" and the *Border Gateway Multicast Protocol* (BGMP)], which we discuss next. It is likely that these will need IPv6 to scale to serious usage.

Bidirectional PIM-SM

The PIM-SM multicast routing protocol builds both source and shared trees for the distribution of multicast packets. PIM-SM shared trees are rooted at special routers called Rendezvous Points and are uni-directional in nature. Shared tree traffic always flows from the Rendezvous Point down to the leaf routers. In some types of multicast applications, namely many-to-many type applications, a unidirectional tree may be inefficient.

Other multicast protocols such as *Core Based Trees* (CBT) and BGMP provide bidirectional shared trees. Bidirectional trees[40] do not have these inefficiencies in many-to-many applications. In a bidirectional tree, traffic from a source is forwarded directly onto the shared tree at the closest point; the traffic is then forwarded both "up" and "down" the tree to all receivers. This is in contrast to a unidirectional tree when the source packets are sent first to the Rendezvous Point (or root) and then down the tree. Recently, two proposals have been submitted that add bidirectional tree capabilities to PIM-SM[40].

BGMP

BGMP[33] is a new inter-domain multicast routing protocol that addresses many of the scaling problems of earlier protocols. BGMP attempts to bring together many of the ideas of previous protocols and adds features that make it more service provider friendly. BGMP is designed to be a unified inter-domain multicast protocol in much the same way that the *Border Gateway Protocol* (BGP) is used for unicast routing.

BGMP is an inter-domain protocol in that it adopts particular design features of BGP familiar to providers. Two of these features follow: it uses TCP connections for the transfer of routing information and it has a state machine (with error notifications) similar to BGP.

In order to accommodate different applications and backward compatibility, BGMP can build three types of multicast trees, both unidirectional source and shared trees and bidirectional shared trees. Unidirectional trees are useful for single-source applications and for backward compatibility with other multicast routing protocols. Shared trees are useful for many-to-many applications (for example, multiplayer gaming, videoconferencing) and multicast forwarding state to scale for these types of applications.

One of the unique properties of BGMP is that its shared trees are rooted at an Autonomous System that is associated with the multicast group address of the tree. Having the root of the tree at the Autonomous System that is associated with the address is logical because there are likely members in that domain. Rooting the trees at an Autonomous System level also provides stability and inherent fault tolerance.

BGMP requires a way to discover which Autonomous Systems "own" which multicast addresses; this can be accomplished through the use of the MASC protocol or through globally assignable multicast addresses (for example, IPv6 multicast). The MASC protocol allocates temporary assignments from the IPv4 group D address space; it then distributes these assignments into *Multiprotocol BGP* (MBGP) so that BGMP will know which Autonomous System is associated with which group and, therefore, where to send join messages.

If globally assignable addresses are available, then BGMP can use any static address architecture for obtaining an Autonomous System from a multicast group address.

The combination of BGMP and a large multicast address space (for example, IPv6 address space) provide the best scaling for all types of multicast applications.

Transport and Congestion Control: Calling Down Traffic on a Site

Multicast is a multiplier. It gives an advantage to senders, but without their knowledge. Multicast (and its application level cousin, the CU-SeeMe reflector) can "attract" more traffic to a site than it can cope with on its Internet access link. (CU-SeeMe is a popular Macintosh- and PC-based Internet videoconferencing package that currently does not directly use IP multicast.) A user can do this by inadvertently joining a group for which there is a high-bandwidth sender, and then "going for a cup of tea." This problem will be averted through access control, or through mechanisms such as charging[58], which may result from the deployment of real-time traffic support.

The problem is seen as critical by ISPs who have a shared bottleneck in their access technology—this is the case for cable modem and in some cases for *Asymmetric Digital Subscriber Line* (ADSL), where a large number of fast lines converge on a slower interface to the backbone. Here, a single user may attract more traffic than this link can handle, without seeing a problem that he or she causes for other users (unicast or other multicast lower-capacity separate sessions using the same shared bottleneck). The use of IGMPv3 with authenticated join and con-figuration management would appear to be a possible solution to these woes. Alternatively, the use of TCP-friendly multicast congestion control (as envisaged for reliable multicast, but also as emerging in some *Real-Time Transport Protocol* (RTP)[4] applications), would also solve this problem.

Congestion Control

One of the critical areas to clarify is the role of congestion control in multicast transport protocols[1]. From an early stage, it was established that coexistence with TCP was a critical design goal for protocols that would operate in the wider Internet. Thus systems such as *TCP Friendly (Reliable) Multicast Congestion Control* (TFMCC)[8], *Pragmatic General Multicast Congestion Control* (PGMCC)[53], and receiver-driven congestion control[54] all extend the classic work by Raj Jain[15] and Van Jacobson[17] and subsequent evolution[16] on TCP congestion avoidance and control.

Recently, this line of thinking has even been extended back into the unicast world in the application of such control schemes to *User Datagram Protocol* (UDP)-like flows in the work on the *Datagram Congestion Control Protocol* (DCCP)[62], suitable for adaptive multimedia flows on RTP, for example.

Reliable Multicast

There is a clear requirement for some sort of analog to TCP for multicast applications that need a level of reliability. The *Internet Research Task Force's* (IRTF's) *Reliable Multicast Research Group* (RMRG) group[3] has developed numerous prototypical solutions to the problem, which turns out to be quite a large design space (not "one size fits all").

The IETF *Reliable Multicast Transport* (RMT) working group has now been chartered to develop single-source reliable multicast transport solutions that meet the current Internet constraints[1]. That group has developed a building block approach[12], which is based partly on abstracting components from existing work such as *Reliable Multicast Transport Protocol* (RMTP) II[18], *Receiver Driven Layered Congestion Control* (RLC)[7], *Multicast File Transfer Protocol* (MFTP)[28], *Pragmatic General Multicast* (PGM)[41], and many other protocols.

Some applications of RMT products are likely to be infrastructural rather than of direct use to the ISPs' customers—for example, distributing software to mirror sites seems to be one popular compelling use.

However, reliable multicast is sometimes regarded as something of an oxymoron. When people talk about "Reliable Multicast," they usually mean a single protocol at a single "layer" of a protocol stack, typically the transport layer (although we have seen people propose it in the network and even link [ATM!] layers too), that can act as any layered protocol can—to provide common functionality for applications (higher layers) that need it.

So what is wrong with that? Well, possibly three things (or more):

- *Fate sharing:* Fate sharing in unicast applications means that as long as there is a path that IP can find between two applications, then TCP can hang on to the connection as long as the parties like. However, if either party fails, the connection certainly fails.

  Fate sharing between multicast end points is a more subtle idea. Should "reliability" extend to supporting the connection fork recipients failing? Clearly this will be application specific (just as timing out on not getting liveliness out of a unicast connection is for TCP—we must permit per-recipient timeouts and failures).

- *Performance:* When A talks to B, the performance is limited by one path. Whatever can be done to improve the throughput (or delay bound) is done by IP (for example, load sharing the traffic over multiple paths). When A talks to B, C, D, E, or F, should the throughput or delay be that sustainable by the slowest or average?

- *Semantics:* As well as performance and failure modes, N-way reliable protocols can have different service models. We could support reliable one-to-n, reliable n-to-one, and reliable n-to-m.

Applications such as software distribution are cited as classic one-to-n requirements. Telemetry is given as an n-to-one reliable protocol. Shared whiteboards are cited as examples of n-to-m applications.

It is interesting to look at the reliability functions needed in these. The one-to-n and n-to-one protocols are effectively *simplex* bulk transfer applications. In other words, the service is one where reliability can be dealt with by "rounding up" the missing bits at the end of the transfer. Because this does not need to be especially timely, there is no need for this to be other than end to end, and application based. (Yes, we know telemetry could be time sensitive, but we are trying to illustrate major differences clearly for now.)

On the other hand, n-to-m processes such as whiteboards need timely recovery from outages. The implication is that the "service" is best done somewhat like the effect of having $n \times (m-1)/2$ TCP connections. If used in the WAN, the recovery may best be distributed, because requests for recovery will implode down the very links that are congested or error prone and cause the need for recovery.

Now there are different schemes for creating distributed recovery. If the application semantics are that operations (application data unit packets worth) are sequenced in a way that the application can index them, then any member of a multicast session can efficiently help any other member to recover (examples of this include Mark Handley's Network Text tool[16].) On the other hand, packet-based recovery can be done from data within the queues between network or transport and application, if they are kept at all members in much the same way as a sender in a unicast connection keeps a copy of all unacknowledged data.

The problem with this is that *because* it is multicast, we do not have a positive acknowledgement system. Therefore, there is no way to inform *all* end points when they can safely discard the data in the "retransmit" queue. Only the application really knows this!

Well, this is not to say that there is not an obvious toolkit for reliable multicast support—it would certainly be good to have RTP-style media timestamps (determined by the application, but filled in by the system). It would be good to have easy access to a timestamp-based receive queue so applications could use this to do all functions discussed previously. It might be advantageous to have virtual Token Ring, expanding ring search, token tree, and other toolkits to support retransmit "helper" selection.

Table 1 illustrates this in terms of where functions might be put to provide reliability (retransmit), sequencing, and performance (adaptive playout, say, versus end to end, versus hop-by-hop delay constraint).

Table 1: Reliable Multicast Semantics

	Recovery	Sequency	Dalliance
Network	not in our internet	ditto	int-serv
Transport	one–many	yes	adaptive
Application	many–many	operation semantics	adaptive

Router Assist for Reliable Multicast

As mentioned in previous sections, one of the difficulties in end-to-end multicast signaling is the "implosion" of signaling at a source from many receivers. This problem has been addressed in numerous ways, including the use of timers, the use of servers to aggregate signaling, and the use of router-assisted mechanisms. We now discuss three protocols that make use of router assistance in order to better scale end-to-end multicast protocols.

PGM[41] is a *negative acknowledgement* (NAK)-based router-assisted reliable multicast protocol. PGM uses routers to aggregate receiver-to-source signals (for example, the NAKs) as they flow toward the source. PGM router support also includes a subcasting ability whereby repairs will flow down only to receivers who have requested them.

Extending the ideas of router assist in PGM is the *Generic Multicast Transport Service* (GMTS). GMTS provides generic, fixed, simple services for any end-to-end multicast transport protocol. These services include such features as signal aggregation with predicates and sophisticated subcasting ability. GMTS was used as a basis for *Generic Router Assist* (GRA)[34], which is similar, IETF standards oriented, and a bit more streamlined.

Securing Multicast

Multicast security is more difficult than unicast security in several areas. The key exchange protocols used between unicast hosts do not scale to groups. Rekeying is required more often to maintain confidentiality as group membership changes. And the efficient authentication transforms used between two unicast hosts cannot protect traffic between mutually distrustful members of a group.

These problems are being worked on by the IETF *Multicast Security* (msec) and IRTF *Group Security* (gsec) working groups. Because of the wide range of application requirements in group communication, their work is based upon a building block approach similar to that of the RMT group.

The blocks being developed are data security transforms, group key management and group security association, and group policy management[49]. An application may use different blocks together to create a protocol that meets its specific requirements.

Data Security Transforms

A data security transforms block provides confidentiality and authentication services for data being transported between group members. Confidentiality is reasonably easy to provide using standard encryption algorithms. Authentication is more difficult, because the algorithms used in unicast protocols such as *IP Security* (IPSec) would not allow a group member to authenticate data as being from another specific group member. This is because the secret used to authenticate the traffic must be shared between all sending and receiving parties. Public-key signatures would solve this problem, but are an order of magnitude slower than symmetric authentication algorithms and hence especially unsuitable for real-time traffic and low-powered communications devices.

Instead, blocks such as the *Timed Efficient Stream Loss-tolerant Authentication Protocol* (TESLA)[55] are being developed that trade off small amounts of functionality (such as immediate rather than slightly delayed authentication) to retain the efficiency benefits of symmetric algorithms. TESLA senders use a hash chain of keys $k_{n...1}$ to sign data, where: $k_n = hash(k_{n-1})$

They release each key in the chain a short interval after the data the key has signed. As long as other group members received the data during that interval, they can be confident that the signature was made by the sender. If keys are lost during transmission, receivers can recompute any key earlier in the sequence simply by repeatedly applying the hash function used to any later key received. Finally, they can be sure that keys are coming from the sender because the first key in the sequence is digitally signed, while only the sender can know the later keys in the sequence (because by definition, a hash function must not be reversible).

Group Key Management and Group Security Association

To use data security transforms, group members need to possess the cryptographic keys necessary to encrypt or decrypt and sign or authenticate data. They also need to agree on parameters such as specific encryption algorithms. This building block allows this information to be shared between group members.

The Group Key Management architecture[47] provides a unified model for key management blocks. A central *Group Controller/Key Server* (GCKS) provides *Traffic Encrypting Keys* (TEKs) or *Key Encrypting Keys* (KEKs) to new group members after authenticating them with a unicast protocol. The GCKS may also delegate some of its functions to other entities, improving scalability.

In groups with simple security requirements, this may be the only communication required between a group member and GCKS. But if group changes need to be cryptographically enforced, further TEKs, encrypted using a KEK, may be provided to members by multicast or a more scalable protocol such as the *Logical Hierarchy of Keys* (LHK)[56] that does not require every rekey message to be sent to every group member. Alternatively, noninteractive mechanisms such as hash trees may be used to update keys[48]. Finally, group members may explicitly de-register with the GCKS using a one- or two-step message.

Three key management building blocks are being developed. The *Group Domain of Interpretation* (GDOI) builds on the *Internet Security Association Key Management Protocol* (ISAKMP)[52] to allow the creation and management of security associations for IPSec and other network or application layer protocols[46]. *Multimedia Internet Keying* (MIKEY) is targeted at real-time multimedia communications, particularly those using the Secure RTP, and can be tunneled over the *Session Initiation Protocol* (SIP)[45]. And a *Group Secure Association Key Management Protocol* (GSAKMP), along with a GSAKMP-Light profile, have also been developed[51].

Group Policy Management

The final building block defines policies such as which roles various entities may play in the group; who may hold group information such as cryptographic keys; the cryptographic algorithms used to protect group data; and proof that the creator of a given policy is authorized to do so. A group policy token is used to hold all of this information[50]. All or part of tokens can be made available to users in policy repositories or by using other out-of-band mechanisms.

Operational Deployment of Multicast

As mentioned previously, multicast seems to be difficult to deploy. One problem is that it has only recently moved from the research community (and typically implemented using tunnels) into the service community (running native IP multicast routing).

This means that debugging multicast sessions, applications, and routing is a common activity. However, because of the dynamic nature of multicast addresses and the anonymous nature of the multicast service model, debugging is somewhat more difficult than for the equivalent unicast case.

Fortunately, all current native multicast paths are at least computed from underlying unicast ones, and it is possible to use tools such as *mtrace* and *mrm* to query the underlying router system to try to figure out where things are going on. Of course, the relevant *Management Information Bases* (MIBs) need to be designed, but mere *Simple Network Management Protocol* (SNMP) access to the variables defined in these may not be enough.

Many multicast sessions are global, and not surprisingly, someone, somewhere, sometime in the session will have a problem. In a way, you only have to look at multicast as a way of sampling large pieces of the Internet at one time to see why it is difficult to understand. In fact, a research project called *Multicast-Based Inference of Network-Internal Characteristics* (MINC)[9, 57] is using that very observation to build tools of more general use.

MRM

One recent tool that has been developed to facilitate multicast monitoring and debugging is the *Multicast Reachability Monitor* (MRM)[32]. MRM consists of two parts; a MRM management station configures test senders and test receivers in multicast networks. A multicast test sender or test receiver is any server or router that supports the MRM protocol and can source or sink multicast traffic. MRM provides the ability to dynamically test particular multicast scenarios; this capability can be used for fault isolation and general monitoring of sessions.

MRM is typically used to configure MRM-capable routers as test senders and test receivers from a management station. Routers configured as test senders send multicast packets periodically to a configured multicast group at a configured rate. Routers configured as test receivers monitor traffic to a group and keep statistics that can be reported back via *RTP Control Protocol* (RTCP) packets. Test receivers can be configured to send RTCP reports when a given condition has been reached or when polled by a management station. Although the MRM protocol is simple itself, it provides powerful capabilities that can be used by future multicast debugging applications.

Research Ideas in Multicast Routing and Addressing

The seeming complexity exhibited by the full panoply of multicast protocols has led some people to develop doubts as to the eventual deployment of multicast. It is far too early to say whether these doubts are well founded. The slow pace of deployment is a symptom not just of this complexity, but also of the underlying complexity of handling growth and evolution of *any* type in such a large system as the Global Internet.

Having said that, it is worth mentioning four of the approaches that have been discussed in the Internet community recently:

- *Addressable Internet Multicast* (AIM), by Brian Levine, et al., attempts to provide explicit addressing of the multicast tree. The routers run a tree-walking algorithm to label all the branch points uniquely, and then make these labels available to end systems. This allows numerous interesting services or refinement of multicast services to be built. Of some particular interest would be the ability this service gives to end systems to do subcasting, which would be useful for some classes of reliable transport protocols.

- *Explicitly Requested Single-Source* (Express), by Hugh Holbrook et al., is aimed at optimizing multicast for a single source. The proposal includes additional features such as authentication and counting of receivers, which could be added to many other multicast protocols usefully. It is motivated by a perceived requirement from some ISPs for these additional features. Express makes use of an extended address (channel + group) to provide routing without global agreement on address assignment. A possible source of problem for AIM is the potential for unbounded growth in the size of identifiers for labeling subtree branch points.
- *Root Addressed Multicast Architecture* (RAMA), by Radia Perlman et al., is in some senses a generalization of Express type addressing, but it also requires bidirectional trees (CBT like, rather than current PIM-SM, although work on bidirectional PIM is under way too). The goal is to offer a single routing protocol for both intra- and interdomain. In fact, RAMA can be implemented by combining the address extensions proposed for Express, and two-level bidirectional PIM as an implementation of BGMP. RAMA and Express (and bidirectional PIM) require a mechanism for carrying additional information in multicast IP data packets.

  There are two critical problems for carrying this identifier that are difficult to solve in general: first, it takes new space in the IP packet, and this has to be accessed by both hosts and routers—that represents a deployment problem; secondly, in the general case, the extra field must be examined on the "fast path," in routers that have such a concept, and this takes valuable processing resources that may have to be taken away from some other forwarding task.
- *Connectionless Multicast* (CM) by Dirk Ooms, et al., is a proposal for small, very sparse groups to be implemented by carrying lists of IP unicast addresses in packets. The scheme is not simply a form of loose source routing, because it would make use of packet replication at appropriate branch points in the network. It may be well suited to IP telephony applications where a user starts with a unicast call, but then adds a third or fourth participant.
- The *L'Ecole Polytechnique Fédérale de Lausanne* (EPFL) work on *Distributed Core Multicast* (DCM) aims to address very large numbers of very small groups with mobile users, typical characteristics of mobile IP telephony users making conference or group calls.
- MIT has done some work on the use of wide-area "anycast" addresses for the core and Rendezvous Point. This results in a potential improvement in the availability of trees (and subtrees) for multicast delivery in the event of router or link outage. More importantly, it may be possible for a multicast group to survive network partitions (or lack of core reachability), a possibility that would make this an invaluable improvement to the service. It depends on the scalability of the wide-area anycast solution, which the MIT work shows is at least viable, and certainly worth more attention.

- *Yet Another Multicast* (YAM) routing protocol[30] was devised by Ken Carlberg of SAIC to address the possibility of forming different multicast trees based on some QoS metric—the idea is that IGMP is modified to provide a "one-to-many" join, and a receiver sends this with required performance parameters. Routers receiving the request over links that can provide this service respond. The receiver (sender of the one-to-many IGMP) selects the one to then commit the join to.
- *Quality of Service Sensitive Multicast Internet protoCol* (QoSMIC) is a development from YAM by Faloutsos[29] at Toronto, and slightly modifies the tree-building exercise.
- When multicast and *Multiprotocol Label Switching* (MPLS) are mentioned together, there is both confusion and surprise. MPLS can be used with multicast in two very different ways. The first method is by building multicast trees over MPLS traffic-engineered paths. Some multicast routing protocols already make use of unicast forwarding information for the construction of multicast trees. Using multicast traffic-engineered paths is simply an extension of this concept—with one caveat. Some multicast routing protocols use *Reverse Path Forwarding* (RPF) checks on incoming packets to prevent looping; this is accomplished by checking to see if the incoming interface is the "closest" to the source. With MPLS traffic engineering, RPF checks are difficult. A solution has not been presented at this time that addresses this problem.

  The second method for using multicast with MPLS is through the use of point-to-multipoint virtual circuits in much the same way as ATM point-to-multipoint virtual circuits. These are useful in cases where receivers are statically configured to a multicast address or multicast traffic is always to be delivered to a destination. Mapping dynamic memberships into a multipoint circuit has proven difficult, for example, with ATM. There are currently several Internet drafts that propose various solutions for MPLS and multicast[31].
- Several groups have been working on end system-only multicast schemes, probably most notably Carnegie-Mellon University[59].

Summary and Conclusions

In this article, we have looked at some of the newer ideas in the research and development community in the area of multicast. There is still a lot to be done to close the loop between network services, transport, and applications, but present research indicates that we will eventually achieve this goal.

References

[0] M. Handley and J. Crowcroft, "Internet Multicast Today," *The Internet Protocol Journal,* Vol. 2, No. 4, December 1999.

[1] A. Mankin, A. Romanow, S. Bradner, and V. Paxson, "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols," RFC 2357, June 1998.

[2] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," Proceedings of SIGCOMM '98, September 1998.

[3] Reliable Multicast Research Group:
http://www.east.isi.edu/RMRG/

[4] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, January 1996.

[5] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing, Scalable Reliable Multicast (SRM)," Proceedings of ACM SIGCOMM '95.

[6] M. Handley and J. Crowcroft, "Network Text Editor (NTE): A scalable shared text editor for the Mbone," Proceedings of ACM SIGCOMM '97, September 1997.

[7] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like Congestion Control for Layered Multicast Data Transfer," Proceedings of INFOCOM '98.

[8] M. Handley, S. Floyd, and B. Whetten, "Strawman specification for TCP friendly (reliable) multicast congestion control (TFMCC)," work in progress.

[9] S. R. Caceres, N. Duffield, J. Horowitz, D. Towsley, and T. Bu, "Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation," Proceedings of IEEE Infocom '99, March 1999.

[10] S. J. Cowley, "Of Timing, Turn-taking, and Conversations," *Journal of Psycholinguistic Research,* 1998, Vol. 27, No. 5, pp. 541–571.

[11] Jonathan Rosenberg and Henning Schulzrinne, "Timer Reconsideration for Enhanced RTP Scalability," Proceedings of the Conference on Computer Communications (IEEE Infocom), March/April 1998.

[12] B. Whetten, L. Vicisano, R. Kermode, M. Handley, S. Floyd, and M. Luby, "Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer," RFC 3048, January 2001.

[13] Handley, M. et al., "Rate Adjustment Protocol," Proceedings of Infocom 1999.

[14] Kouvelas, I. et al., "Self Organising Transcoders," Proceedings of NOSSDAV 1998.

[15] D-M. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithm hms for Congestion Avoidance," *Computer Networks and ISDN Systems,* Vol. 17, pp. 1–14, 1989.

[16] S. Floyd and K. Fall, "Router Mechanisms to Support End-to-End Congestion Control," Technical report,
ftp://ftp.ee.lbl.gov/papers/collapse.ps

[17] V. Jacobson, "Congestion Avoidance and Control," Proceedings of ACM SIGCOMM '88, August 1988, pp. 314–329.

[18] J. C. Lin and S. Paul, "RMTP: A Reliable Multicast Transport Protocol," Proceedings of IEEE INFOCOM '96, March 1996, pp. 1414–1424.

[19] M. Mathis, J. Semke, J. Mahdavi, and T. Ott,"The Macroscopic Behaviour of the TCP Congestion Avoidance Algorithm," ACM *Computer Communication Review,* Vol. 27 No. 3, July 1997.

[20] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast," Proceedings of SIGCOMM '96, August 1996, pp. 1–14.

[21] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modelling TCP Throughput: A Simple Model and Its Empirical Validation," Proceedings of SIGCOMM '98, September 1998.

[22] L. Rizzo and L. Vicisano, "A Reliable Multicast Data Distribution Protocol Based on Software FEC Techniques," The Fourth IEEE Workshop on the Architecture and Implementation of High Performance Communication Systems (HPCS '97), June 1997.

[23] Dan Rubenstein, Jim Kurose, and Don Towsley, "The Impact of Multicast Layering on Network Fairness," Proceedings of ACM SIGCOMM '99, August 1999.

[24] N. Shacham, "Multipoint Communication by Hierarchically Encoded Data," Proceedings of IEEE Infocom '92, 1992, pp. 2107–2114.

[25] Chris Greenhalgh, Steve Benford, Adrian Bullock, Nico Kuijpers, and Kurt Donkers, "Predicting Network Traffic for Collaborative Virtual Environments," *Computer Networks and ISDN Systems,* Vol. 30, 1998, pp. 1677–1685.

[26] Steve Deering, "Host Extensions for IP Multicasting," RFC 1112, August 1989.

[27] S. Deering, C. Partridge, and D. Waitzman, "Distance Vector Multicast Routing Protocol," RFC 1075, November 1988.

[28] Ken Miller, "Multicast File Transfer Protocol," White Paper, Starburst Technologies.

[29] Michalis Faloutsos, Anindo Banerjea, and Rajesh Pankaj, "QoSMIC: Quality of Service Sensitive Multicast Internet ProtoCol," ACM *Computer Communication Review,* Vol. 28, pp. 144–153, September 1998.

[30] K. Carlberg and J. Crowcroft, "Building Shared Trees Using a One-To-Many Joining Mechanism," ACM *Computer Communication Review,* Vol. 27, pp. 5–11, January 1997.

[31] D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul, and F. Ansari, "Framework for IP Multicast in MPLS," work in progress.

[32] K. Almeroth, K. Sarac, and L. Wei, "Supporting Multicast Management Using the Multicast Reachability Monitor (MRM) Protocol," UCSB CS Technical Report, May 2000.

[33] D. Thaler, D. Estrin, D. Meyer, et al., "Border Gateway Multicast Protocol (BGMP)," Proceedings of ACM SIGCOMM '98, 1998.

[34] B. Cain, T. Speakman, and D. Towsley, "Generic Router Assist Building Block," work in progress.

[35] B. Cain and D. Towsley, "Generic Multicast Transport Services (GMTS)," Proceedings of Networking 2000, Paris, France, May 2000.

[36] B. Fenner, "Domain Wide Multicast Group Membership Reports," work in progress.

[37] D. Farinacci et al., "Multicast Source Discovery Protocol," Internet Draft, January 2000, work in progress.

[38] B. Cain, "Connecting Multicast Domains," Internet Draft, work in progress, October 1999.

[39] D. Thaler, "Interoperability Rules for Multicast Routing Protocols," RFC 2715, October 1999.

[40] D. Estrin and D. Farrinacci, "Bi-directional Shared Trees in PIM-SM," work in progress.

[41] T. Speakman et al., "PGM Reliable Transport Protocol Specification," RFC 3208, December 2001.

[42] B. Cain, S. Deering, and A. Thyagarajan, "Internet Group Key Management Protocol, Version 3," work in progress.

[43] H. Holbrook and D. Cheriton, "IP Multicast Channels: Express Support for Large-scale Single-source Applications," Proceedings of SIGCOMM '99, September 1999.

[44] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment Issues for the IP Multicast Service and Architecture," IEEE Network Magazine, Special Issue on Multicasting, January/February 2000.

[45] J. Arkko, E. Carrera, F. Lindholm, M. Naslund, and K. Norrman, "MIKEY: Multimedia Internet KEYing," Internet Draft, work in progress, February 2002.

[46] M. Baugher, T. Hardjano, H. Harney, and B. Weis, "The Group Domain of Interpretation," Internet Draft, work in progress, February 2002.

[47] M. Baugher, R. Canetti, L. Dondeti, and F. Lindholm, "Group Key Management Architecture," Internet Draft, work in progress, February 2002.

[48] B. Briscoe, "MARKS: Zero Side Effect Multicast Key Management Using Arbitrarily Revealed Key Sequences," Proceedings of Networked Group Communication, November 1999.

[49] T. Hardjano, R. Canetti, M. Baugher, and P. Dinsmore, "Secure IP Multicast: Problem Areas, Framework, and Building Blocks," Internet Draft, work in progress, September 2000.

[50] T. Hardjano, H. Harney, P. McDaniel, A. Colgrove, and P. Dilmore, "Group Security Policy Token," Internet Draft, work in progress, November 2001.

[51] H. Harney, A. Schuett, and A. Colegrove, "GSAKMP Light," Internet Draft, work in progress, July 2001.

[52] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)," RFC 2408, November 1998.

[53] Luigi Rizzo, "pgmcc: A TCP-friendly Single-Rate Multicast Congestion Control Scheme," Proceedings of ACM SIGCOMM '2000, August 2000.

[54] Luby et al., "Wave and Equation Based Rate Control Using Multicast Round Trip Time," Proceedings of ACM SIGCOMM '2002, September 2002.

[55] A. Perrig, R. Canetti, B. Briscoe, D. Tygar, and D. Song, "TESLA: Multicast Source Authentication Transform," Internet Draft, work in progress, November 2000.

[56] D. M. Wallner, E. Harder, and R. C. Agee, "Key Management for Multicast: Issues and Architectures," RFC 2627, September 1998.

[57] F. Lo Presti, N.G. Duffield, J. Horowitz, and D. Towsley, "Multicast-Based Inference of Network-Internal Delay Distributions," **http://www.cs.umass.edu/pub/Lopr99TR9955.ps.Z**

[58] T. Henderson and S. Bhatti, "Protocol Independant Multicast Pricing," Proceedings of NOSSDAV 2001.

[59] Yang-hua Chu, Sanjay G. Rao, and Hui Zhang, "A Case for End System Multicast," Proceedings of ACM SIGMETRICS, June 2000, pp. 1–12.

[60] Multicast Address Allocation Working Group, **http://www.icir.org/malloc/**

[61] D. Meyer and P. Lothberg, "GLOP Addressing in 233/8," RFC 3180, September 2001.

[62] **http://www.icir.org/dccp/**

IAN BROWN holds a BSc from The University of Newcastle upon Tyne and a PhD from University College London. His research has focused on network security and active networking. He is a member of the ACM, IEEE, and is a contributor to the Internet Engineering Task Force, particularly in the area of authorized emergency communications. He has also worked extensively on the social implications of technology, and is a trustee of Privacy International and advisory board member of the Foundation for Information Policy Research. His e-mail address is:
I.Brown@cs.ucl.ac.uk

BRAD CAIN is a Senior Consulting Engineer at Storigen Systems, where he contributes to product development in the areas of networking and storage technology. Prior to joining Storigen, Cain was chief scientist at Cereva Networks, where he worked on system architecture and new product development. Cain also worked at Mirror Image Internet, one of the first commercial Content Delivery Networks (CDNs), where he helped architect their content distribution system. Cain is a contributor in the IETF and IRTF in the areas of IP multicast, IP routing, MPLS, and content networking. He has published numerous papers in the areas of routing and multicast and has more than 40 patents pending in the areas of multicast, security, routing, and router architecture. Cain holds a masters and bachelors in electrical engineering from the University of Delaware.
E-mail: **Brad.Cain@storigen.com**

JON CROWCROFT is the Marconi Professor of Networked Systems at the University of Cambridge. Prior to that he was professor of networked systems at University College London (UCL) in the Computer Science Department. He is a member of the ACM, a Fellow of the British Computer Society, a Fellow of the IEE, and a Fellow of the Royal Academy of Engineering, as well as a senior member of the IEEE. He is a member of the IAB, and was general chair for ACM SIGCOMM from 1995 to 1999. He is on the editorial team for the ACM/IEEE *Transactions on Networks and Computer Communications,* as well as on the program committee for ACM SIGCOMM and IEEE Infocomm. He has published five books—the latest is *Linux TCP/IP Implementation,* published by Wiley in 2001.
E-mail: **Jon.Crowcroft@cl.cam.ac.uk**

MARK HANDLEY received his BSc in Computer Science with Electronic Engineering from University College London in 1988 and his PhD from UCL in 1997. For his PhD he studied multicast-based multimedia conferencing systems, and was technical director of the European Union funded "MICE" and "MERCI" multimedia conferencing projects. After two years working for the University of Southern California's Information Sciences Institute (ISI), he moved to Berkeley to join the new ICSI Center for Internet Research (formerly known as ACIRI). Most of his work is in the areas of scalable multimedia conferencing systems, reliable multicast protocols, multicast routing and address allocation, and network simulation and visualization. He is co-chair of the IRTF Reliable Multicast Research Group, and he previously chaired the IETF Multiparty Multimedia Session Control working group. E-mail: **mjh@icir.org**

Zero Configuration Networking

by Edgar Danielyan, Danielyan Consulting

Zero configuration networking may sound like an oxymoron to many who spend most of their time setting up and mending networks. But don't decide on a career change yet—although zero configuration networks exist and work, they don't work always and everywhere. In this article I describe the current state of the affairs in zero configuration IP networking, introduce Zeroconf, the suite of zero configuration IP protocols, and tell what they do and how they work. This article is only a brief introduction to zero configuration networking and Zeroconf, so if you are really interested in all the details, refer to the sources listed in the References section at the end of this article.

The best introduction to Zeroconf is the one from the *Zeroconf Working Group* of the *Internet Engineering Task Force* (IETF)[1]:

> "The goal of the Zero Configuration Networking (Zeroconf) is to enable networking in the absence of configuration and administration. Zero configuration networking is required for environments where administration is impractical or impossible, such as in the home or small office, embedded systems 'plugged together' as in an automobile, or to allow impromptu networks as between the devices of strangers on a train."

Essentially, to reduce network configuration to zero (or near zero) in *Internet Protocol* (IP) networks, it is necessary, inter alia, to:

- Distribute IP addresses (without a *Dynamic Host Configuration Protocol* [DHCP] server),
- Provide name resolution (without a *Domain Name System* [DNS] server),
- Find and list services (without a directory service), and
- Distribute multicast IP addresses, if necessary (without a multicast server).

These and other requirements are defined in an Internet Draft titled "Requirements for Automatic Configuration of IP Hosts" by Aidan Williams[2]. This document does not define Zeroconf protocols themselves but instead spells out the requirements that should be met to achieve effective and useful zero configuration IP networking. One of the most important requirements for any Zeroconf protocol is that it should not interfere with other protocols and it must be able to exist on the same network with other non-Zeroconf protocols and devices. Another requirement is "no less" security—Zeroconf protocols should not be less secure than existing non-Zeroconf protocols—more on this later. Although IPv6 addresses some of the requirements of zero configuration networking (such as automatic allocation of link-local addresses), other requirements have yet to be met for both IPv4 and IPv6.

Zeroconf IETF Working Group

The Zeroconf Working Group of the IETF is chaired by Erik Guttman of Sun Microsystems and Stuart Cheshire from Apple Computer, with Thomas Narten (IBM) and Erik Nordmark (Sun) serving as area directors. It was chartered in September 1999 and had its first meeting at the 46th IETF in Washington, D.C., in November 1999. Those interested in the work of Zeroconf WG may find the mailing list archive of the working group at:

`http://www.merit.edu/mail.archives/zeroconf/`

Where and When to Use Zeroconf

For a correct understanding of the applicability and usefulness of Zeroconf it is necessary to keep in mind that it is a *link-local* technology. Link-local addressing and naming are meaningful only in a particular network; link-local addresses and names are not global and are not unique globally. In this case it means that Zeroconf is intended for use in small wired or wireless local-area networks in situations and places where zero configuration is necessary. It is appropriate to use Zeroconf in such networks when there is no possibility (or it is inappropriate) to set up a working IP network using the traditional technologies such as DNS and DHCP. Zeroconf is not appropriate and should not be used in many cases, for example in:

- Medium or large networks
- Networks where a high degree of security and control is required
- Large public access networks
- Networks with low bandwidth and high latency (such as some wireless networks)

When inappropriately used, Zeroconf may bring more problems and headaches than it solves. In contrast, examples of correct and appropriate use would include:

- Home and small office networks
- Ad hoc networks at meetings and conferences (especially wireless networks)
- Two devices needing to spontaneously share or exchange information

Likewise, Zeroconf advantages from one viewpoint may become annoying problems from another. Consider, for instance, the automatic distribution and configuration of link-local IP addresses. For a home network user this is a blessing—no longer do you have to spend time creating an addressing scheme and setting the IP addresses and netmasks on devices that should just work. But for an enterprise network (especially an incorrectly configured one), sudden appearance of nodes with (yet) unfamiliar and strange (this is not your regular **`10.*`** or **`192.168.*`**) IP addresses may result in more than surprise and added workload for the network administrator.

Continuing in this manner, Multicast DNS (mDNS) that ends the misery of having to remember and type **`ftp 10.20.30.1`** every time you need to transfer files from or to your PC named Bobo and replaces it with just **`ftp bobo`** may result in strange behavior on some networks. The bottom line? Zeroconf is not a one-size-fits-all solution; it wasn't designed to be one, and will not work as one.

Zeroconf and Security

Security should occupy an important place in the minds of all networking professionals, so an introduction to zero configuration networking would be incomplete without a mention of its security position. Security goals of Zeroconf are defined in section 4, Security Considerations, of "Requirements for Automatic Configuration of IP Hosts"[2]:

"Zeroconf protocols are intended to operate in a local scope, in networks containing one or more IP subnets, and potentially in parallel with standard configured network protocols. Application protocols running on networks employing zeroconf protocols will be subject to the same sets of security issues identified for standard configured networks. Examples are: denial of service due to the unauthenticated nature of IPv4 ARP and lack of confidentiality unless IPSec-ESP, TLS, or similar is used. However, networks employing zeroconf protocols do have different security characteristics, and the subsequent sections attempt to draw out some of the implications.

Security schemes usually rely on some sort of configuration. Security mechanisms for zeroconf network protocols should be designed in keeping with the spirit of zeroconf, thus making it easy for the user to exchange keys, set policy, etc. It is preferable that a single security mechanism be employed that will allow simple configuration of all the various security parameters that may be required. Generally speaking, security mechanisms in IETF protocols are mandatory to implement. A particular implementation might permit a network administrator to turn off a particular security mechanism operationally. However, implementations should be "secure out of the box" and have a safe default configuration.

Zeroconf protocols MUST NOT be any less secure than related current IETF-Standard protocols. This consideration overrides the goal of allowing systems to obtain configuration automatically. Security threats to be considered iclude both active attacks (e.g. denial of service) and passive attacks (e.g. eavesdropping). Protocols that require confidentiality and/or integrity should include integrated confidentiality and/or integrity mechanisms or should specify the use of existing standards-track security mechanisms (e.g. TLS (RFC 2246), ESP (RFC 1827), AH (RFC 2402) appropriate to the threat."

Although this document does not address each and every aspect of security issues with Zeroconf, it sets requirements for Zeroconf protocols. As is the case with traditional IPv4 and IPv6, use of such techniques as *IP Security Architecture* (IPSec) or *Transport Layer Security* (TLS) may be appropriate in some cases. However, the nonstatic (or one may say nondurable) nature of both IP addresses and names in Zeroconf environment may pose a problem for IPSec and TLS deployment.

Dynamic Configuration of IPv4 Link-Local Addresses

Generally speaking, the first requirement that should be fulfilled before any useful IP communication can occur are the IP addresses of sender and recipient. The IP addresses are usually either assigned and set manually or provided by some other means such as DHCP or the *Point-to-Point Protocol* (PPP). However, neither of these is possible in zero configuration networks. Therefore, an automatic mechanism for dynamic configuration of IP addresses without any manual intervention or dependence on third-party service (that is, DHCP) is necessary. This mechanism already exists in IPv6 but not in IPv4. In "Dynamic Configuration of IPv4 Link-Local Addresses"[3], Stuart Cheshire, Bernard Aboba, and Erik Guttman describe a method that may be used in IPv4 networks to automatically assign IPv4 addresses valid for local communication on a particular interface. A special network **`169.254/16`** is reserved with the *Internet Assigned Numbers Authority* (IANA) for this purpose. It is necessary to highlight that **`169.254/16`** addresses are reserved for link-local use only. The document also addresses such issues as support for multiple addresses and multiple interfaces, continuous address conflict detection, effects of joining previously not interconnected networks, and other considerations.

IPv4 Address Conflict Detection

Address conflicts in IP networks are annoying problems that (needlessly) take time and effort to detect and rectify, so a separate document on address conflict detection was deemed necessary. "IPv4 Address Conflict Detection"[4] by Stuart Cheshire presents two things: first, a way to prevent this unfortunate situation of conflicting IP addresses from happening, and second, a way to detect address conflicts if they do happen even after all the precautions. Both of these are accomplished using the *Address Resolution Protocol* (ARP). Interestingly, in the Security Considerations section of the document the author states:

> "The ARP protocol [RFC 826] is insecure. A malicious host may send fraudulent ARP packets on the network, interfering with the correct operation of other hosts. For example, it is easy for a host to answer all ARP requests with responses giving its own hardware address, thereby claiming ownership of every address on the network.
>
> This specification makes this existing ARP vulnerability no worse, and in some ways makes it better: Instead of failing silently with no indication why, hosts implementing this specification are required to either attempt to reconfigure automatically, or if not that, at least inform the human user of what is happening."

Although some may argue about the question of whether or not it is effective, appropriate, and useful to "inform the human user" in this case, this solution nevertheless follows the principle of at least not worsening the current security situation of an existing protocol.

Zeroconf Multicast Address Allocation Protocol

The *Zeroconf Multicast Address Allocation Protocol* (ZMAAP) defined in[5] specifies a method for peer-to-peer allocation of *multicast addresses without a multicast* (MADCAP) server in small zero configuration networks. The word "small" is important here because ZMAAP is not scalable beyond small networks (and is not designed to be).

Multicast DNS

"Performing DNS queries via IP Multicast"[6] by Stuart Cheshire suggests some very useful ideas on how to use mDNS with maximum benefit and minimum hassle in zero configuration networks. In my opinion, the best thing about this proposal is that it does not require any changes to the DNS protocol (messages, resource record types, etc.) itself. Instead it concentrates on the use of multicast for name resolution in environments where no DNS servers exist (and where one would not reasonably expect them to). The goal is to have a working name resolution service without name servers. The document proposes to use **`local.arpa`** (although the exact choice of this special domain is not the goal of this document) as the link-local domain (like the **`169.254/16`** network for dynamic allocation of IPv4 link-local addresses described earlier in this article). For reverse address resolution, **`254.169.in-addr.arpa`** is also link-local. The multicast address **`224.0.0.251`** that is used for mDNS queries is registered by the IANA for this purpose. No delegation is performed within mDNS domain **`local.arpa`**. There is also no *Start of Authority* (SOA) record for the mDNS domain because of the nature of zero configuration networks where it is intended to be used—in particular, there is no mailbox responsible for the zone. Likewise, zone transfers are not applicable with mDNS zones. To summarize, any local link has its own local and private **`local.arpa`** and **`254 254.169.in-addr.arpa`** zones, which have only link-local significance in the particular Zeroconf network.

DNS Service Discovery

Like the multicast DNS solution described previously, the *DNS Service Discovery* (DNS-SD)[7] does not require any changes to the existing DNS protocol; thus it is completely compatible with the existing DNS server and client software.

What DNS-SD proposes is a naming scheme for DNS *Resource Records* (RRs) to allow for service discovery using the existing DNS—either the traditional or multicast DNS described in the previous paragraph. DNS-SD uses the SRV and PTR resource records to provide the required functionality. To cite from [7]:

"Service discovery requires a central aggregation server. DNS already has one: It's called a DNS server.

Service discovery requires a service registration protocol. DNS already has one: It's called DNS Dynamic Update.

Service discovery requires a security model. DNS already has one: It's called DNSSEC.

Service discovery requires a query protocol. DNS already has one: It's called DNS."

It is necessary to note that DNS-SD is compatible with mDNS and vice versa, but neither requires the other one to function. However, it is practical to use mDNS for service discovery (using DNS-SD) to have a single protocol and interface and not have to implement another protocol just for service discovery.

Industry Support

Any new technology needs industry support to succeed, and Zeroconf is no exception. Several major vendors have announced plans to support or already support Zeroconf in their products, including Apple, Epson, Hewlett-Packard, Lexmark, Philips, Canon, Xerox, Sybase, and World-Book. One can expect that more companies will Zeroconf-enable their products as the technology itself matures and hopefully becomes standardized and widespread.

Rendezvous

Rendezvous is Apple Computer's implementation of Zeroconf in its Darwin 6 and Mac OS X 10.2 ("Jaguar") operating systems. Apple has stated its full support for the Zeroconf and intent to completely replace the aging AppleTalk with Zeroconf-enabled Macs, without sacrificing the ease of use and transparency to end users provided by AppleTalk networks. A good example of Zeroconf's use in OS X would be the iChat instant messaging (IM) client, which comes with the Version 10.2 of Mac OS X. It works not only with *AOL Instant Messenger* (AIM) and Mac networks but may also be used between Zeroconf-enabled Macs in a Zeroconf network.

Coupled with Apple's implementation of IEEE 802.11b ("WiFi") in ad hoc mode, it permits a wireless zero configuration network that just works without any configuration or additional hardware or software.

Apple has also made the source code for the mDNS Responder, a part of Rendezvous implementing mDNS, freely available through the Darwin Open Source Project. Mac OS X software developers are encouraged to use Zeroconf, and there are documentation and application examples to facilitate this. More information about Rendezvous and Zeroconf on Macs is available from Apple's Web sites[9].

Summary

With computers and computer networks becoming more and more complex and sophisticated, some people (including the author of this article) believe that care should be taken by those in the know not to create more problems than we solve using these computers and networks. Yes, we want more features—but we also need to remember that most users of these features do not have doctorates in computer science and (surprise, surprise) don't even wish to. Zero configuration networking would probably help in this regard, minimizing and even eliminating in some cases the need to configure and administer small networks. Let me conclude by quoting once more from the Zeroconf Working Group:

> "It is important to understand that the purpose of Zeroconf is not solely to make current personal computer networking easier to use, though this is certainly a useful benefit. The long-term goal of Zeroconf is to enable the creation of entirely new kinds of networked products, products that today would simply not be commercially viable because of the inconvenience and support costs involved in setting up, configuring, and maintaining a network to allow them to operate."

References

[1] Zeroconf Working Group, Internet Engineering Task Force (IETF): **http://www.ietf.org/html.charters/zeroconf-charter.html**

[2] Aidan Williams, "Requirements for Automatic Configuration of IP Hosts," **draft-ietf-zeroconf-reqts-12.txt**

[3] Stuart Cheshire, Bernard Aboba, and Erik Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses," **draft-ietf-zeroconf-ipv4-linklocal-07.txt**

[4] Stuart Cheshire, "IPv4 Address Conflict Detection," **draft-cheshire-ipv4-acd-02.txt**

[5] Octavian Catrina, Dave Thaler, Bernard Aboba, and Erik Guttman, "Zeroconf Multicast Address Allocation Protocol (ZMAAP)," **draft-ietf-zeroconf-zmaap-02.txt**

[6] Stuart Cheshire, "Performing DNS Queries via IP Multicast," **draft-cheshire-dnsext-multicastdns-00.txt**

[7] Stuart Cheshire, "Discovering Named Instances of Abstract Services Using DNS," **draft-cheshire-dnsext-nias-00.txt**

[8] Zeroconf: **http://www.zeroconf.org**

[9] Rendezvous: **http://developer.apple.com/macosx/rendezvous/** **http://www.apple.com/macosx/jaguar/rendezvous.html**

[10] Erik Guttman, "Autoconfiguration for IP Networking: Enabling Local Communication," *IEEE Internet Computing,* June 2001.

EDGAR DANIELYAN is a self-employed consultant, author, and editor specialising in UNIX, networking, and information security. In previous life he has been a cofounder of a national ISP and manager of a country TLD. He is currently working on his next book (*WLAN Security*) which is due to be published in 2003. His previous book, *Solaris 8 Security,* was published by New Riders Publishing in 2001. He is also a member of IEEE, IEEE Standards Association, IEEE Computer Society, ACM, USENIX, and the SAGE. He is online at **http://www.danielyan.com** and can be reached by e-mail at **edd@danielyan.com**

Book Reviews

Ruling the Root

Ruling the Root: Internet Governance and the Taming of Cyberspace, by Milton L. Mueller, ISBN 0-262-13412-8, The MIT Press, 2002, **http://mitpress.mit.edu**

> "WASHINGTON, Apr. 1 /Governance Newswire/ — The organizations that create street names, assign addresses, and assign telephone numbers have issued a joint announcement: Henceforth any conversation not conducted in Bahasa Malayu will result in termination of the relevant address or telephone number assignment."

The above bit of fiction is not pure silliness. Fear of equivalent, Internet-related excesses is the essence of Milton Mueller's book, *Ruling the Root*. The Syracuse University professor believes that administration of Internet addresses and domain names provides a fulcrum for overall Internet governance. He says they create a "political economy" vulnerable to serious abuse. Domain name administration is equated with control over Internet content, because, "a domain name record [is] very much like an Internet driver's license" as if it provides permission to use the Net, and even authorizes the locations one may visit.

Organization

The book covers both IP address and domain name administration. The material on IP addresses is thin, perhaps because it is a well-managed area without significant controversy. This is in marked contrast to the recent history of debate on *Domain Name System* (DNS) oversight. So it might have been instructive to see a comparison between the two administrative models, beyond simply noting that domain names can be interesting.

Discussion covers Internet technology, the history and politics of DNS and IP administrative management structure, and the intellectual property aspects of name assignment conflicts. Mueller suggests a three-layer hierarchy: technical, economic, and policy. What is missing from this "architecture" and from the entire book is any concern for the pragmatic details of administration and operation of these global, mission-critical services. Yet such tasks are difficult to perform well, as Network Solutions repeatedly demonstrated over the years, by losing registrations and corrupting critical data files; and the effects of problems are large.

When *Star Trek's* Captain Picard commands, "make it so," we know that he fully appreciates the challenges in implementing his directive. However, for *Ruling the Root,* policy development is not concerned with the operational complexities.

Not surprisingly, the book often demonstrates a misunderstanding of constraints inherent in DNS technology, although the tutorial on basic Internet technology is adequate, in spite of making the common error about the "T" in TCP/IP.[1]

Differing Opinions

Other reviewers of the book have called it well written, insightful, and nuanced. Indeed the discussion of history that is fully documented and involves simple, clear, objective facts is quite good. The rest of the time Mueller presents biased and unfounded descriptions of Internet governance, motives, and decisions, while failing to distinguish between what is fact and what is his opinion.

Ruling the Root sees adversaries, conspiracies, and threats, and permits no balancing sense of diverse collaboration, constructive criticism, or productive compromise. The technical community is somewhat less suspect, but is deprecated with the usual cliche about its naivete. So Mueller misses the essential point that techies designed, built, operated, and grew this robust, survivable, equitable system for global operations and service governance.

Professor Mueller's treatment of the dominant DNS registry, *Network Solutions* (NSI), now VeriSign, is curiously superficial and soft. NSI benefited spectacularly from the National Science Foundation's decision to permit charging for domain names, and from the policies and delays in the formation of the *Internet Corporation for Assigned Names and Numbers* (ICANN), as well as ICANN's distraction away from its intended registry oversight function and toward abstract debates about Internet governance. Yet the book does not consider NSI's role in ICANN-related political processes.

Mueller fails to understand the history of the organization that managed the DNS from its inception, the *Internet Assigned Numbers Authority* (IANA) and Jon Postel's role in running it. IANA is incorrectly represented as a simple operations arm of the U.S. Government. The grass-roots basis for its real legitimacy is missed. Its policy role is missed. Its collaborative processes are denied. For example, Mueller tells us that the description of IANA in RFC 1083, published in 1988 meant, "a new world was being defined by the RFC." In reality it was simply documenting established practice, as is typical for operations RFCs.

Validation

Mueller's substantiation of his analyses is also problematic. The book must be read with careful attention to the actual authority of each source. Goals and agendas are often misstated. For example, he characterizes the pre-ICANN *International Forum for the White Paper* (IFWP) as "the real arena for arriving at a decision [about the details of the new organization]." Its actual goal was simply to be a forum for discussion. Discussion, not decision-making.[2]

The book claims that the pre-ICANN *International Ad Hoc Committee* (IAHC) was formed "to develop and implement a blueprint for a global governance structure for the domain name system." In fact, the IAHC was formed for "specifying and implementing policies and procedures relating to iTLDs (international top-level domains, now called 'generic' TLDs, or gTLDs)."[3] He claims, "They had asserted that the root was theirs to dispose of." To the contrary, the IAHC was explicitly subordinate to IANA, and had nothing at all to do with management of the DNS root or any non-gTLD part of the DNS. Interestingly, the endnote Mueller offers as substantiation disproves his characterization.

Ruling the Root is loaded with endnotes—27 pages of small print. However, even the formal citations are problematic. Note #55 cites a newspaper article as a primary source, as if it were definitive proof the person discussed in the article held a specific opinion. Mueller's Note #45 claims to substantiate that, "Postel himself... admitted...it is unclear who actually controls the name space." Yet the note is for *Internet Architecture Board* (IAB) minutes. Attributing it to Postel was a fabrication.

Back-room, deal-making, conspiracy explanations are offered without substantiation. Of changes to *Internet Engineering Task Force* (IETF) management, Mueller states: "The most important reason the IETF didn't institute voting was that Jon Postel and several other senior figures vowed that they would refuse to run for office." Postel never made such a vow, and the process to effect these IETF changes did not experience any such attempts at influence. Of Postel's instructing some root servers to retrieve copies of the DNS root from a non-NSI master, Mueller claims that Postel was "apparently concerned about the direction U.S. policy was taking."

No substantiation is offered, because the claim is false. Postel and others were concerned about NSI's reaction to its own loss of control. The switch was intended to see what it would take to move NSI out of the hierarchy. These are not small matters of nuance. They show a pattern of misrepresentation.

The Author

Professor Mueller's credibility would have been aided by disclosing his own affiliations. The only ICANN constituency (the Non Commercial Domain Name Holders Constituency) claiming to represent the non-commercial world focuses on the civil society concerns that dominate the public debate about ICANN. Professor Mueller's discussion of the group is quite thin and does not disclose the fact that he held a dominant management position in it. In his criticism of dispute-resolution activities, he neglects to mention that he is a paid arbitration panelist.

An important book should be read because it has factual detail and thoughtful insight. *Ruling the Root* is, instead, important because it so thoroughly embodies the difficulties that have emerged in discussing Internet policy. Because so many people take *Ruling the Root* seriously, it should be read. However, the serious problems of the book encourage borrowing it, rather than buying a copy. Based on the pattern noted in this review, a thorough audit of those problems would be appropriate for the relevant Syracuse University academic ethics committee.

—*Dave Crocker*[4], *Brandenburg InternetWorking*
dcrocker@brandenburg.com

References

[1] The "T" stands for transmission, not transport or transfer.

[2] **http://web.archive.org/web/19981206105122/**
http://www.ifwp.org/

[3] **http://www.iahc.org/iahc-charter.html**

[4] Factual claims in the review that do not have citations are based on the reviewer's direct experience. Dave Crocker wrote the first Internet standard for domain name syntax (RFC 822). He also was the IETF area director for initial work on DNS security. More recently he was one of Jon Postel's appointees to the IAHC. He naively thought that its work should be conducted in the manner that had been typical for Internet administration. So the last few years of charged, global politicization have been an education. He must also note that he was once Jon Postel's officemate.

High-Speed Networks and Internets

High-Speed Networks and Internets: Performance and Quality of Service, 2nd ed., by William Stallings, ISBN 0-13-032221-0, Prentice Hall, 2002. **http://www.prenhall.com/stallings**

This thoroughly updated classic covers topics of traffic engineering, queuing, and traffic modeling. The book gives a complete look around the protocols of the next generation: *Resource Reservation Protocol* (RSVP), *Multiprotocol Label Switching* (MPLS), and *Real-Time Transport Protocol* (RTP). It gives the keys to understand the way Frame Relay, TCP, and ATM react to congestion and flow control. The book also deals with new trends and standards that will lead the telecommunications industry in the following years. A very useful book, from the same author of traditional titles such as: Data Communications, Cryptography, Computer Architecture, and many more.

Organization

High-Speed Networks is divided into seven parts. The first one discusses the basic background needed to understand the rest of the book. Following the introduction, the second chapter goes on with the classical: the *Open System Interconnection* (OSI) model and the TCP/IP suite.

Part II explains packet-switching technologies in detail. The forth chapter explains the architecture of Frame Relay, and the next one focuses on ATM, including its operation and the adaptation layers. Chapter 6 works on high speed LANs, covering Fast Ethernet and Gigabit Ethernet, with the different media supported by each.

The third part is one of the most important; chapter 7 presents an overview of probability and stochastic processes. Although it is a brief one, it is useful to make revision of some concepts. The next chapter works on queuing analysis, introducing the basic elements of a queuing model. It explains the topics with plenty of examples: M/M/1, multiserver queues, and networks of queues, presenting all the formulas. Chapter 9 is dedicated to self-similar traffic. As recent studies indicate, traffic on high speed networks does not have the characteristics needed for the queuing theory. It introduces and explains the concept of self-similarity. Then the author applies this concept to data traffic analysis and examines performance implications. Based on papers on this subject, Stallings explains this new approach to traffic modeling not analyzed before.

The forth part focuses on another main topic: congestion and traffic management. Chapter 10 explains the effects of congestion and the different ways to control and avoid it. In the following chapter the author discusses control mechanisms at the link level. He examines different ways used by protocols to handle flow control: *Stop and Wait, Sliding Window,* and *Go back N-ARQ*. An analysis of the performance gained by using *Automatic Repeat Request* (ARQ) techniques follows.

These chapters give a detailed description of the different ways that communications can be handled. Chapter 12 focuses on transport-level traffic management. It explains TCP flow control in detail, including the retransmission strategy. The way TCP avoids congestion is discussed thoroughly. The next chapter continues with congestion control in ATM networks. The framework for traffic control is explained in detail, with sections dedicated to *Available-Bit-Rate* (ABR) and *Guaranteed-Frame-Rate* (GFR) traffic management.

The next part of the book is about Internet routing. Chapter 14 presents the algorithms used to compute the minimum path, and introduces some elementary concepts in graph theory. Later the author concentrates on Interior routing protocols, analyzing the *Routing Information Protocol* (RIP) and *Open Shortest Path First* (OSPF), the most important ones. Next the book discusses exterior routing protocols and multicast. The author describes in a simple way these addressing schemes and the related protocols.

The following section is dedicated to *Quality of Service* (QoS) in IP networks. The first chapter discusses integrated services, with coverage of queuing disciplines such as *Weighted Fair Queuing* (WFQ). A review of the Differentiated Services architecture follows.

After discussing the concepts, the author examines the protocols that support QoS: RSVP, MPLS, and RTP. He explains the philosophy behind each protocol, its characteristics, and its implementation.

In the final part of the book, the author changes the subject to compression. In Chapter 19 he presents an overview of information theory, discussing typical areas such as entropy. The next chapter continues with loss-less compression, facsimile compression, and others. It discusses the Lempel-Ziv algorithm used in PKZIP. The final chapter reviews lossy compression, explaining the discrete cosine transform, a key component of the *Joint Photographics Expert Group* (JPEG) and *Motion Picture Experts Group* (MPEG) standards.

Two very interesting appendices end the book: one for Internet standards and the standardization process and the other one dedicated to sockets, containing source code. Although the book is not dedicated to programming, the inclusion of TCP sockets can be useful to understand its implementation.

A book worth reading

We are facing an essential book for networking professionals, designers, and engineers. It covers unusual topics such as self-similar traffic and data compression. It is the basement for the design of any high speed network. As Internet traffic continues to grow, the optimization of network resources becomes a critical topic. Also, more and more voice traffic is carried over packet networks, congestion being one of its worst enemies. The time-sensitive traffic needs attention, and this book provides the tools to manage it.

In addition to its solid coverage of topics, the book has plenty of bibliography and many links to the principal sites for each chapter. With no doubt this is a very useful book, from the well-known technical author William Stallings.

—Rodrigo J. Plaza, Iplan Networks, Argentina
rplaza@iplan.com.ar

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Letters to the Editor

ENUM

Ole,

As the co-chair of the ENUM work group in the IETF, I was delighted with Geoff Huston's article. (*The Internet Protocol Journal,* Volume 5, No. 2, June 2002, page 13).

I would like to point out and clarify several other issues raised by the Letters to the Editor published in the subsequent issue.

First, as a practical matter though the North American Numbering Plan uses a single country code "1," there will not be a single administration of ENUM within "1." The agreements between the IAB and the ITU on the administration of **e164.arpa** clearly indicate that these resources will be administered on a nation-state basis.
www.iab.org/DOCUMENTS/enum-pr.html
www.iab.org/DOCUMENTS/sg2-liaison-e164-sep-02.html

The United States, Canada, Bermuda, and the 18 countries of the NANP will be free to administer their numbering resources as they so choose through the use of 1 + NPA (area codes) zones within the root of **e164.arpa**.

Dr. Deleuze writes, "E.164 numbers are really telephone addresses. They are tied to telephone network topology and are surely not user friendly. There are no user-friendly names in the telephone system."

In fact, this is not exactly correct either. Since the advent of Number Portability by several national telephone administrations, including the United States, telephone numbers are no longer tied to the underlying network or routing structure of the PSTN. Actual routing of phone calls in the United States is done on Local Routing Numbers for all landline calls and, beginning in November of 2003, for wireless calls as well.

Phone numbers even now are essentially names, much like domain names in the Internet. In the United States, phone numbers can be taken or "ported" to any wireline service provider within proscribed geographic boundaries, in 2003 between wireless service providers and from wireline to wireless providers as well.

I partially take issue with Dr. Deleuze's thought that telephone numbers are not "user-friendly." Phone numbers are readily identifiable, easy to use, and are not tied to culture or language, problems we have not yet solved with domain names.

—Richard Shockey, NeuStar Inc.
rich.shockey@NeuStar.com

Visitor Networks

Dear Editor,

The September 2002 issue of IPJ featured a very interesting, comprehensive article on visitor networks. One aspect I found not mentioned, however, is the danger of users in such scenarios falling victim to fake visitor gateways. In public wireless hot spots, as they are increasingly being setup at numerous locations these days, attackers could employ their own mobile WLAN device to direct visitors trying to log on to the hot spot to their own fake login page, enabling them to easily collect their login details such as credit card information. Using encryption does not help here as long as the gateway does not need to authenticate itself to the customer's mobile device. The average user should not have a chance to realize whether he or she is connected to a legitimate or a fake login page—if he or she is aware of that potential danger at all. Given the fact that all such an attack would need, apart from readily available equipment such as a portable computer with a WLAN card, is some small piece of appropriate software and that it would be quite difficult to detect,that kind of threat unfortunately should be quite realistic in such environments.

—Dr. Georg Schwarz
Detecon International GmbH, Berlin, Germany
Georg.Schwarz@detecon.com

The author responds:

This is a good point that was not discussed in the article. There are actually at least three cases that visitors need to worry about. The first is, as you mentioned, that the service provider is not who they say they are. This can be dealt with by using SSL certificates assuming the visitor is conscious of the URL that he/she is being directed to and knows that it belongs to the real service provider. If the visitor has no idea who is a reasonable service provider, this is a different class of problem, very similar to what has happened with public telephones that accept standard calling and credit cards—someone makes a call, receives the service but then gets charged an outrageous rate. The third case is a man-in-the-middle attack or passive snooping where someone with a laptop as you describe is able to grab traffic and gather passwords.

Some basic advice to visitors is for services that require subscription, although possibly inconvenient, never subscribe on a potentially compromised connection. That way, only the service provider-assigned username and password is compromised, instead of more sensitive personal information related to the account. Connections using 802.1x authentication with EAP-TLS provide mutual authentication and are in the long run, a better solution than redirection of web pages. No matter what kind of security one has, inevitably there will be legally legitimate providers that will take advantage of visitors and in that case it's just "buyer beware."

—Dory Leifer
leifer@del.com

Wireless Security

Hi Ole,

Again, I found the latest issue of IPJ quite enlightening and useful. However, I do have one comment regarding the article by Greg Scholz on "An Architecture for Securing Wireless Networks." Although the use of source IP addresses to provide policy group membership on the firewall works in most cases, some client OSs and some IPSec VPN boxes allow the source address (even if it is the endpoint address of the tunnel, not the "real" address of the host) to be changed,provided the source address of the enciphered traffic does not change. This would allow users to change the policy group they belong to. A better solution is to use a VPN box that can associate groups of IPSec tunnels to VLANs. Then the firewall could be configured to allow policy group membership based on VLANs. This takes all determination of policy group membership off the client host and places it in the domain of trust of the VPN and firewall boxes.

—Chris Liljenstolpe
Cable and Wireless
chris@cw.net

Fragments

Upcoming Events

The IETF will meet in San Francisco, California, USA March 16–21, 2003. The IETF will also meet in Vienna, Austria, July 13–18, 2003 and in Minneapolis, Minnesota November 9–14, 2003.
See **http://www.ietf.org/meetings**

The next APRICOT (*Asia and Pacific Regional Internet Conference on Operational Technologies*) will be held in Taipei, Taiwan, February 19–28. See **http://www.apricot2003.net/**

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in Rio de Janeiro, Brazil, March 23–27, 2003, in Montreal, Canada, June 22–26, 2003, and in Carthage, Tunisia, December 1–5, 2003. See **http://www.icann.org**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

March 2003 Volume 6, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Even the most carefully designed and operated IP network is subject to any number of performance problems ranging from overloaded links and mis-configured routers to server failures. For these situations, the network manager has several diagnostic tools as options. Geoff Huston gives us an overview in an article entitled "Measuring IP Network Performance."

Voice over IP (VoIP) is an emerging application, as well as a rapidly growing market. Use of the corporate network or the Internet at large to carry telephone traffic has many advantages, not the least economic ones. A successful VoIP network must not only support IP-based telephones, but also provide a means of seamlessly integrating the IP-based network with traditional telephone networks. At the core of VoIP lies the *Session Initiation Protocol* (SIP) and a few related protocols. Bill Stallings describes SIP in our second article.

Book reviews published in *The Internet Protocol Journal* can rarely be characterized as "controversial." However, when the book in question deals with ICANN, it is perhaps not surprising that strong opinions emerge. Thus, following the review of *Ruling the Root* in our last issue, we received a letter from the author that is included in our "Letters to the Editor" section (along with a response from the book reviewer). I would like to take this opportunity to remind our readers that book reviews do represent the *opinion* of the reviewer and should be read in that light.

Our online subscription system has been up and running for a couple of months. Please give it a try at: **www.cisco.com/ipj**.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Measuring IP Network Performance

by Geoff Huston, Telstra

If you are involved in the operation of an IP network, a question you may hear is: "How *good* is your network?" Or, to put it another way, how can you measure and monitor the quality of the service that you are offering to your customers? And how can your customers monitor the quality of the service you provide to them?

These questions have been lurking behind many public and enterprise IP networks for many years now. With the increasing levels of deployment of various forms of high-speed (or broadband) services within today's Internet there is new impetus to find some usable answers that allow both providers and users to place some objective benchmarks against the service offerings. With the lift in access speed with broadband services, there is an associated expectation on the part of the end user or service customer about the performance of the Internet service. It should be "better" in some fashion, where "better" relates to the performance of the network and the service profile that is offered to network applications. And not only is there an expectation of "better" performance, it should be measurable. This article looks at network performance and explores its definition and measurement.

A Functional Definition of Network Performance

An informal functional approach to a definition of network performance is measuring the speed of the network. How fast is the network? Or, what is the elapsed time for a particular network transaction? Or, how quickly can I download a data file? This measurement of time for a network transaction to complete certainly relates to the speed of the network, and speed is a good network performance benchmark, but is speed everything?

When looking at the broad spectrum of performance, the answer is that speed is not everything. The ability of a network to support transactions that include the transfer of large volumes of data, as well as supporting a large number of simultaneous transactions, is also part of the overall picture of network load and hence of network performance. But large data sets is not everything in performance. Consideration should also be given to the class of network applications where the data is implicitly clocked according to some external clock source. Such real-time applications include interactive voice and video, and their performance requirements include the total delay between the end points, or latency, as well as the small-scale variation of this latency, or *jitter*. Such performance measurements also include the ratio of discarded packets to the total number of packets sent, or loss rate, as well as the extent to which a sequence of packets is reordered within the network, or even duplicated by the network. Taken together, this set of performance factors can be considered as a form of the amount of distortion of the original real-time signal.

Accordingly, a functional description of network performance encompasses a description of speed, capacity, and distortion of transactions that are carried across the network. This informal description of what

constitutes network performance certainly feels to be on the correct path, given that if one knew the latency, available bandwidth, loss, and jitter rates and packet reorder probability as a profile of network performance between two network end points, as well as the characteristics of the network transaction, it is possible to make a reasonable prediction relating to the performance of the transaction.

Taking this informal definition, the next step is to create a more rigorous framework for measuring performance. For any single network path between an entry and egress point, it is possible to measure the path latency, available peak bandwidth, loss rates, jitter profile, and reorder probability. But there is a difference between a description of the performance of a particular path across a network and the performance of the network as an aggregate entity. Given a set of per-path performance measurements, how can you construct a view of the performance of the network? A common methodology is to take a relatively complete set of path measurements across a network and then combine them to create an average metric. Although this accomplishes a useful reduction in the size of the data, there is also a loss of information. The average network performance measurements have little relationship to the performance of any individual path.

There are various ways to improve this loss of information, including weighting the individual path measurements by the amount of traffic passed along the path. Such techniques are indeed to ensure that paths that use far-flung network outliers that carry relatively low volumes of traffic have a much lower impact on the overall network performance metric than the major network transit paths.

Measuring Network Performance

Given these performance indicators, the next step is to determine how these indicators may be measured, and how the resulting measurements can be meaningfully interpreted. At this point it is useful to look at numerous popular network management and measurement tools and examine their ability to provide useful measurements. There are two basic approaches to this task; one is to collect management information from the active elements of the network using a management protocol, and from this information make some inferences about network performance. This can be termed a *passive approach* to performance measurement, in that the approach attempts to measure the performance of the network without disturbing its operation. The second approach is to use an active approach and inject test traffic into the network and measure its performance in some fashion, and relate the performance of the test traffic to the performance of the network in carrying the normal payload.

Measuring Performance with SNMP

In IP networks the ubiquitous network management tool is the *Simple Network Management Protocol* (SNMP). There is no doubt that SNMP can provide a wealth of data about the operational status of each management network element, but can it tell you anything about the overall network performance?

The operation of SNMP is a *polling* operation, where a management station directs periodic polls to various managed elements and collects the responses. These responses are used to update a view of the operating status of the network.

The most basic tool for measuring network performance is the periodic measurement of the interface byte counters. Such measurements can provide a picture of the current traffic levels on the network link, and when related to the total capacity of the link, the relative link loading level can be provided. As a performance indicator this relative link loading level can provide some indication of link performance, in that a relatively lightly loaded link (such as a load of 5 to 10 percent of total available capacity) would normally indicate a link that has no significant performance implications, whereas a link operating at 100 percent of total available capacity would likely be experiencing high levels of packet drop, queuing delay, and potentially a high jitter level. (Figure 1) In between these two extremes there are performance implications of increasing the load. Of course it should be noted that the characteristics of the link have a bearing on the interpretation of the load levels, and a low-latency 10-Gbps link operating at 90-percent load will have very significantly lower levels of performance degradation than a 2-Mbps high-latency link under the same 90-percent load. (Figure 2)

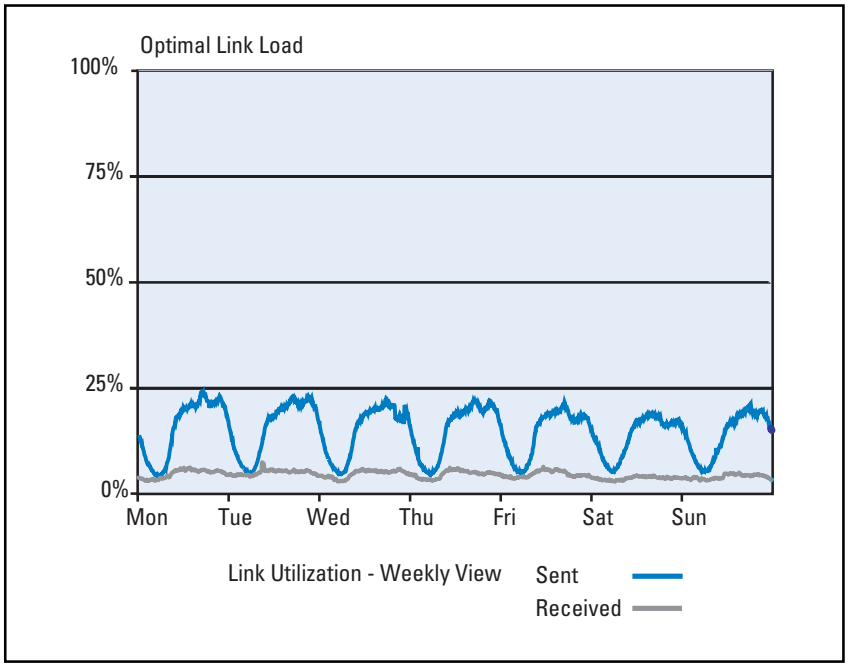


Figure 1a: Relative Link Loading – An Optimally Loaded Link

Figure 1b: Relative Link Loading – A Maximally Loaded Link

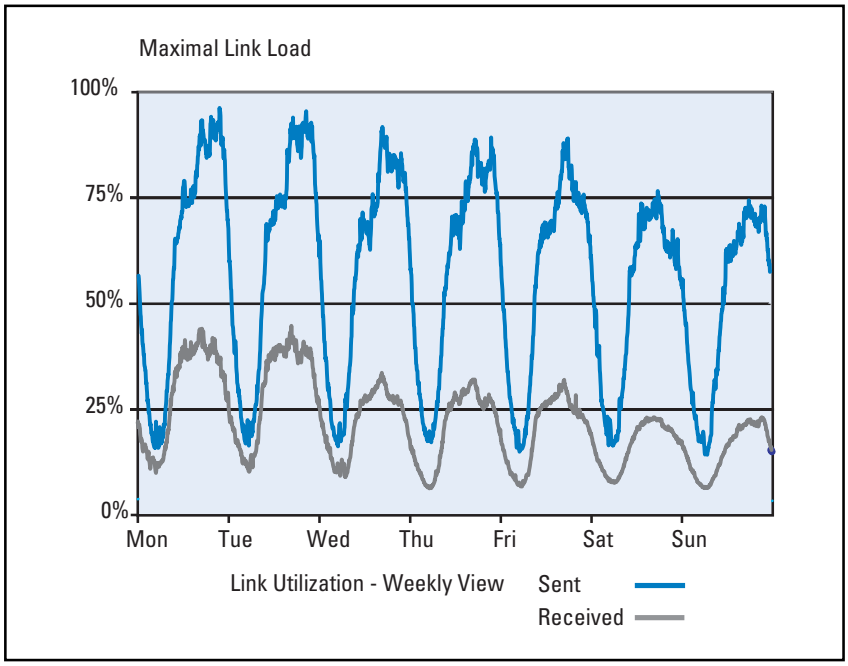


Figure 1c: Relative Link Loading – Highly Degraded Link

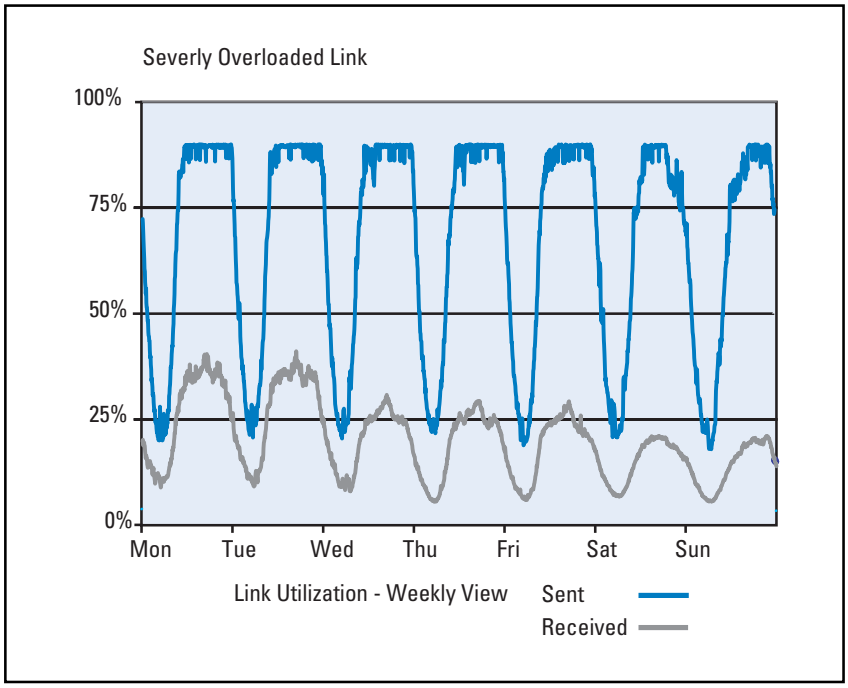
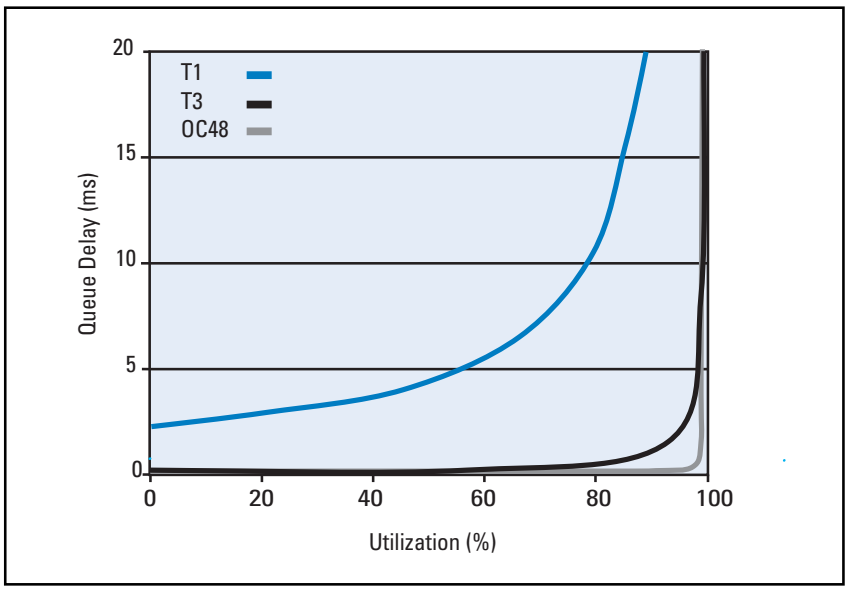


Figure 2: Queuing Delay Simulation (David Meyer, Sprint, November 2002)



Relative traffic load on each link can be complemented by measurement of performance-related SNMP counters. A management system can poll each active network element to retrieve the number of packets dropped for each interface, and the number of packets successfully forwarded. From these two data items, the relative drop proportion of packets can be calculated on an element-by-element and potentially a link-by-link basis, and a series of element measures can provide a per-path drop proportion by combining the individual packet-forwarding measurements for the interfaces on the path.

Because some count of relative packet drop rate can be gathered from each network element, with the additional input of the current forwarding state of the network it is possible to predict the path a packet will take through the network, and hence estimate the path probability of drop. However, this information is still well short of being a reliable measurement of service performance.

Queuing delay is somewhat more challenging to measure on an element-by-element basis using element polling with SNMP. In theory, the polling system could use a rapid sequence of polling the output queue length of a router and estimating the queuing delay based on an average packet size estimate, together with the knowledge of the available output capacity. Of course, such a measurement methodology assumes a simple *first-in, first-out* (FIFO) queuing discipline, a queue size that varies slowly over time, and slow link speeds. Such assumptions are rarely valid in today's IP networks. As the link speed increases, the queue size may oscillate with a relatively high frequency as a function of both the number and capacity of the input systems and of the capacity of the output system. In general, queuing delay is not easily measured using network element polling.

There is no ready way for a polling mechanism to detect and count the incidence of reordered packets. Packet reordering occurs in many situations, including the use of parallel switching fabrics within a single network element and the use of parallel links between routers.

IP routers are not typically designed to detect, let alone correct, packet reordering and because they do not detect this condition, they cannot report on the incidence of reordering via SNMP polling.

The generic approach of network management polling systems is that the polling agent, the network management station, is configured with an internal model of the network; status information, gathered through element polling, is integrated to the network model. The correlation of the status of the model to the status of the network itself is intended to be accurate enough to allow operational anomalies in the network to be recognized and flagged. The challenge is that a sequence of snapshots of element status values cannot readily be reconstructed into a comprehensive view of the performance of the network as an entire system, or even as a collection of edge-to-edge paths. Measurement techniques using polling and modeling can track the performance of the individual elements of the network, but they cannot track per-path service levels across the network. The network-element polling approach can indicate whether or not each network element is operating within the configured operational parameters, and alert the network operator when there are local anomalies to this condition. But such a view is best described as *network centric,* rather than service centric. An implicit assumption is that if the network is operating within the configured parameters, then all service-level commitments are being met. This assumption may not be well founded.

The complementary approach to performance instrumentation of network elements is active network probing. This requires the injection of marked packets into the data stream; collection of the packets at a later time; and correlation of the entry and exit packets to infer some information regarding delay, drop, and fragmentation conditions for the path traversed by the packet. The most common probe tools in the network today are *ping* and *traceroute*.

Measuring Performance with Ping

The best known, and most widely used active measurement tool is *ping*. Ping is a very simple tool: a sender generates an *Internet Control Message Protocol* (ICMP) echo request packet, and directs it to a target system. As the packet is sent, the sender starts a timer. The target system simply reverses the ICMP headers and sends the packet back to the sender as an ICMP echo reply. When the packet arrives at the original sender's system, the timer is halted and the elapsed time is reported. An example ping output is shown in Figure 3.

Figure 3: Example Ping Report

```
% ping www.iab.org
PING www.iab.org (132.151.6.25): 56 data bytes
64 bytes from 132.151.6.25: icmp_seq=0 ttl=44 time=254.409 ms
64 bytes from 132.151.6.25: icmp_seq=1 ttl=44 time=254.197 ms
64 bytes from 132.151.6.25: icmp_seq=2 ttl=44 time=255.238 ms
64 bytes from 132.151.6.25: icmp_seq=3 ttl=44 time=255.874 ms
--- www.iab.org ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 254.197/254.930/255.874/0.670 ms
```

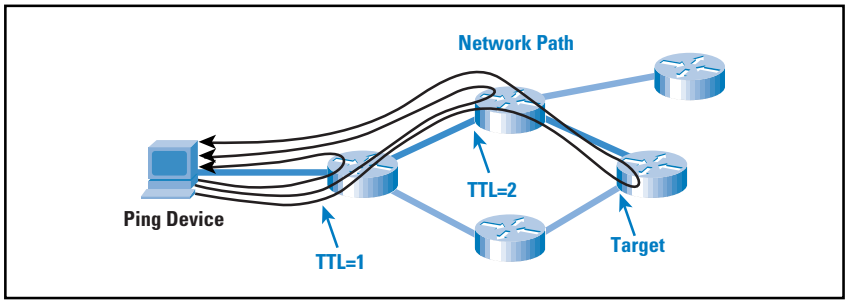
This simple active sampling technique can reveal a wealth of information. A ping response indicates that the target host is connected to the network, is reachable from the query agent, and is in a sufficiently functional state to respond to the ping packet. In itself, this response is useful information, indicating that a functional network path to the target host exists. Failure to respond is not so informative because it cannot be reliably inferred that the target host is not available. The ping packet, or perhaps its response, may have been discarded within the network because of transient congestion, or the network may not have a path to the target host, or the network may not have a path back to the ping sending host, or there may be some form of firewall in the end-to-end path that blocks the ICMP packet from being delivered.

However, if you can ping a remote IP address, then you can obtain numerous performance metrics. Beyond simple reachability, further information can be inferred by the ping approach with some basic extensions to our simple ping model. If a sequence of labeled ping packets is generated, the elapsed time for a response to be received for each packet can be recorded, along with the count of dropped packets, duplicated packets, and packets that have been reordered by the network. Careful interpretation of the response times and their variance can provide an indication of the load being experienced on the network path between the query agent and the target. Load will manifest a condition of increased delay and increased variance, due to the interaction of the router buffers with the traffic flows along the path elements as load increases. When a router buffer overflows, the router is forced to discard packets; and under such conditions, increased ping loss is observed. In addition to indications of network load, high erratic delay and loss within a sequence of ping packets may be symptomatic of routing instability with the network path oscillating between many path states.

A typical use of ping is to regularly test numerous paths to establish a baseline of path metrics. This enables a comparison of a specific ping result to these base metrics to give an indication of current path load within the network.

Of course, it is possible to interpret too much from ping results, particularly when pinging routers within a network. Many router architectures use fast switching paths for data packets, whereas the central processing unit of the router may be used to process ping requests. The ping response process may be given a low scheduling priority because router operations represent a more critical router function. It is possible that extended delays and loss, as reported by a ping test, may be related to the processor load or scheduling algorithm of the target router processor rather than to the condition of the network path. (Figure 4)

Figure 4: Ping Path



Ping sequences do not necessarily mimic packet flow behavior of applications. Typical TCP flow behavior is prone to cluster into bursts of packet transmissions on each epoch of the round-trip time. Routers may optimize their cache management, switching behavior, and queue management to take advantage of this behavior. Ping packets may not be clustered; instead, an evenly spaced pacing is used, meaning that the observed metrics of a sequence of ping packets may not exercise such router optimizations. Accordingly, the ping results may not necessarily reflect an anticipation of application performance along the same path. Also a ping test does not measure a simple path between two points. The ping test measures the time to send a packet to a target system and for the target to respond back to the sender. Ping is measuring a loop rather than a simple path.

With these caveats in mind, monitoring a network through regular ping tests along the major network paths can yield useful information regarding the status of the network service performance.

Many refinements to ping can extend its utility. Ping can use *loose source routing* to test the reachability of one host to another, directing the packet from the query host to the loose source routed host, then to the target host and back via the same path through the specified approach. However, many networks disable support for loose source routing, given that it can be exploited in some forms of security attacks. Consequently, the failure of a loose source routed ping may not be a conclusive indication of a network fault.

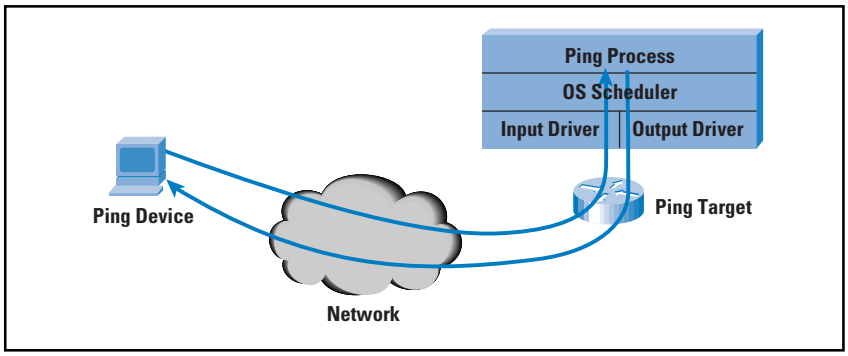
Ping also can be used in a rudimentary way to discover the provisioned capacity of network links. By varying the packet length and comparing the ping times of one router to the next-hop router on a path, the bandwidth of the link can be deduced with some degree of approximation required because of a background queue-induced level of network jitter.

A more sophisticated variation of ping is to pace the transmission of packets from the received packets, mimicking the behavior of the TCP flow control algorithms with *Slow Start* and subsequent congestion avoidance. *Treno* is such a tool. In Treno, the transmission of ping packets is managed by the TCP Reno flow-control algorithm, such that further ping packets are triggered by the reception of responses to earlier packets, and the triggering of further packets is managed by an implementation of the TCP control function. Such a tool can indicate available flow rate-managed capacity on a chosen path.

Path Discovery Using Traceroute

The second common ICMP-based network management tool, *traceroute*, devised by Van Jacobson, is based on the ICMP *Time Exceeded* message. Here, a sequence of *User Datagram Protocol* (UDP) packets are generated to the target host, each with an increased value of the *Time To Live* (TTL) field in the IP header. This generates a sequence of ICMP Time Exceeded messages sourced from the router where the TTL expired. These source addresses are those of the routers, in turn, on the path from the source to the destination. (Figure 5)

Figure 5: Traceroute Path



Like ping, traceroute measures the elapsed time between the packet transmission and the reception of the corresponding ICMP packet. In this way, the complete output of a traceroute execution exposes not only the elements of the path to the destination, but also the delay and loss characteristics of each partial path element. Traceroute also can be used with loose source route options to uncover the path between two remote hosts. The same caveats mentioned in the ping description relating to the relative paucity in deployment of support for loose source routing apply. An example of a traceroute report is shown in Figure 6.

Figure 6. Traceroute report

```
% traceroute www.cisco.com
traceroute to www.cisco.com (198.133.219.25), 64 hops max, 40 byte packets
1  dickson-gw1.Canberra.telstra.net (203.50.0.1)  0.272 ms  0.265 ms  0.270 ms
2  GigabitEthernet4-1.civ12.Canberra.telstra.net (203.50.8.1)  0.402 ms  0.272 ms  0.259 ms
3  GigabitEthernet3-1.civ-core2.Canberra.telstra.net (203.50.7.5)  0.214 ms  0.227 ms  0.193 ms
4  GigabitEthernet2-2.dkn-core1.Canberra.telstra.net (203.50.6.126)  0.459 ms  0.394 ms  0.385 ms
5  Pos4-0.ken-core4.Sydney.telstra.net (203.50.6.121)  3.806 ms  3.762  ms  3.770 ms
6  Pos2-0.pad-core4.Sydney.telstra.net (203.50.6.22) 3.907 ms  3.959 ms  3.913 ms
7  GigabitEthernet0-1.syd-core01.Sydney.net.reach.com (203.50.13.246)  3.898 ms  3.866 ms  3.977 ms
8  i-13-2.sjc-core01.net.reach.com (202.84.143.41)  191.361 ms  191.365 ms  191.341 ms
9  sl-st21-sj-6-1.sprintlink.net (144.223.242.1) 186.955 ms  186.851 ms  187.010 ms
10  sl-bb25-sj-5-1.sprintlink.net (144.232.20.73)  187.241 ms  187.337 ms  187.055 ms
11  sl-gw11-sj-10-0.sprintlink.net (144.232.3.134)  187.279 ms  186.898 ms  186.821 ms
12  sl-ciscopsn2-11-0-0.sprintlink.net (144.228.44.14)  187.572 ms  187.495 ms  187.620 ms
13  sjck-dirty-gw1.cisco.com (128.107.239.5)  184.533 ms  184.686 ms  184.694 ms
14  sjck-sdf-ciod-gw1.cisco.com (128.107.239.106)  184.676 ms  184.686 ms  184.644 ms

15  www.cisco.com (198.133.219.25)  185.017 ms 185.122 ms 185.019 ms
```

Notes:
1) There are interprovider handovers at hops 7, 9, and 13.
2) There is a sudden jump in response times at hop 8. The additional 182 ms of round-trip latency corresponds to a 36,000-km submarine cable path. This can be explained by the hop-7 to hop-8 segment, including a submarine cable path between Australia and the United States.

Traceroute is an excellent tool for reporting on the state of the routing system. It operates as an excellent "sanity check" of the match between the design intent of the routing system and the operational behavior of the network.

The caveat to keep in mind when interpreting traceroute output has to do with asymmetric routes within the network. Whereas the per-hop responses expose the routing path taken in the forward direction to the target host, the delay and loss metrics are measured across the forward and reverse paths for each step in the forward path. The reverse path is not explicitly visible to traceroute.

One-Way Measurements

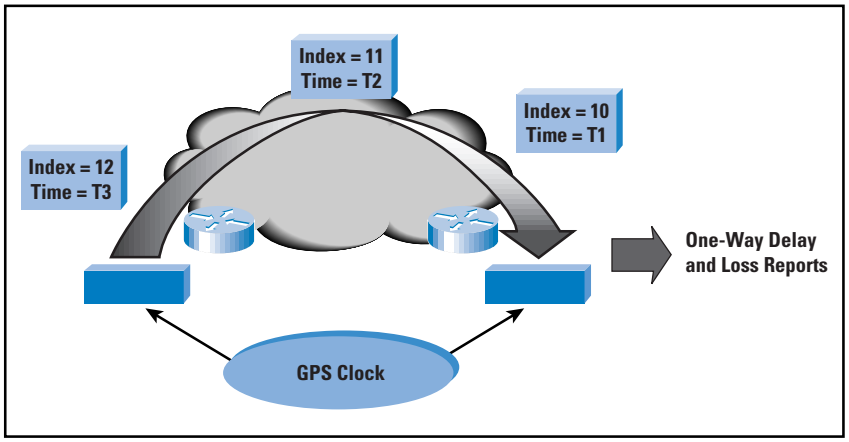
Round-trip probes, such as ping and traceroute, are suited to measuring the total network path between two ends of a transaction, but how can a network provider measure the characteristics of a component of the total end-to-end path? In such a case the network provider is interested in the performance of a set of unidirectional transit paths from an network ingress point to an egress point. There are now some techniques that perform a one-way delay and loss measurement, and they are suited to measuring the service parameters of individual transit paths across a network. A one-way approach does not use a single network management system, but relies on the deployment of probe senders and receivers using synchronized clocks.

The one-way methodology is relatively straightforward. The sender records the precise time a certain bit of the probe packet was transmitted into the network; the receiver records the precise time that same bit arrived at the receiver. Precisely synchronizing the clocks of the two systems is an interesting problem, and initial implementations of this approach have used *Global Positioning System* (GPS) satellite receivers as a synchronized clock source.

One of the noted problems with the use of GPS was that computers are generally located within machine rooms and a clear GPS signal is normally available only on a rooftop. Later implementations of this approach have used the clock associated with the *Code Division Multiple Access* (CDMA) mobile telephone network as a highly accurate, synchronized, distributed clock source, with the advantage that the time signal is usually available close to the measurement unit.

Consequent correlation of the sender's and receiver's data from repeated probes can reveal the one-way delay and loss patterns between sender and receiver. To correlate this to a service level requires the packets to travel along the same path as the service flow and with the same scheduling response from the network.

Figure 7: One-Way Measurements



Ping and traceroute are ubiquitous tools. Almost every device can support sending ping and traceroute probes, and, by default almost every device, including network routers, will respond to a ping or traceroute probe. One-way measurements are a different matter, and such measurements normally require the use of dedicated devices in order to undertake the clocking of the probes with the required level of precision (Figure 7).

Choosing the Right Time Base

Whether it is an active or passive measurement regime, the next basic decision is the time base to use for the measurements. Many applications are very sensitive to short-lived transient network conditions. This may take the form of a burst of packet loss, or a period of packet reordering, or a switch to a longer round trip time. TCP may react by halving its sending rate, or by entering an extended wait state while awaiting the retransmission timer to expire. In either case it will take numerous round trip time intervals for the transport session to recover, and this may impact the behavior of the application. On the other hand, a periodic network probe may miss the transient event altogether and report no abnormalities whatsoever.

IP networks have bursty traffic sources, and there is a marked self-similarity in the traffic patterns. This appears to be consistent over a wide range of networks, where large-capacity systems tend to observe large burst patterns and smaller systems also see bursts of a similar proportionate size. So the question is, what time interval for measurements can provide meaningful aggregation of information, while at the same time be sensitive enough to report on the outcomes of transient bursts within the network? Intuitively a measurement time base of hourly measurements is very insensitive to capturing transient bursts, whereas a time base of a millisecond would generate a massive amount of data, a scenario that would tend to smother the identification of abnormalities. Interestingly enough, the choice of a measurement base has little to do with the capacity of the links within a network, but it has a close relationship to the average routing trip time of the individual transport sessions that are active within the network.

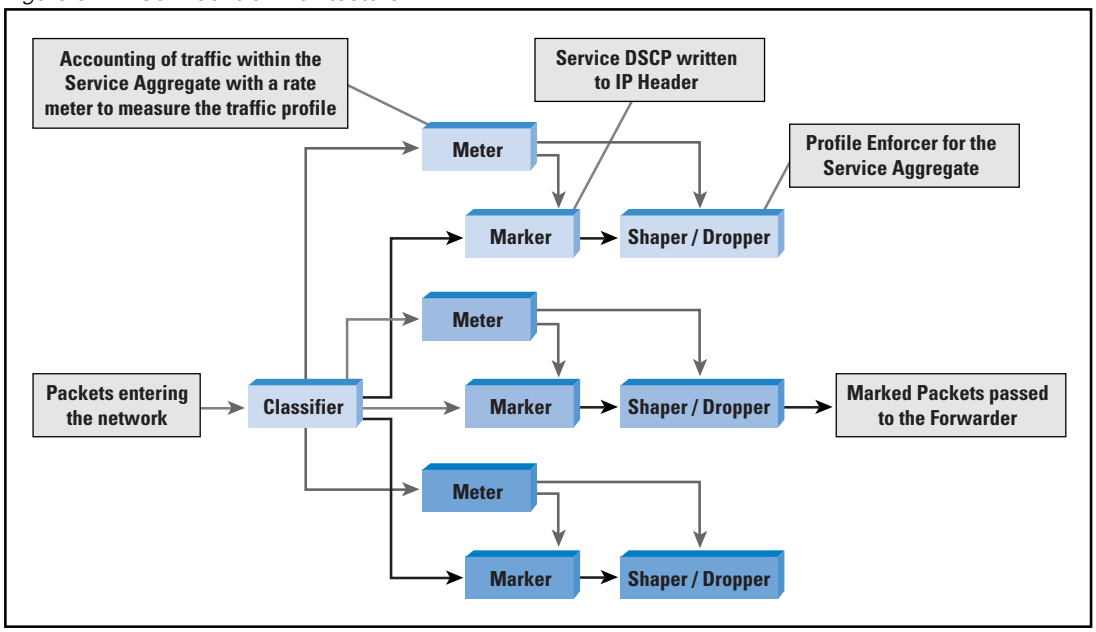
The profile of IP networks is one that is dominated by TCP traffic, and TCP traffic uses a transport control mechanism where the returning stream of *acknowledgement* (ACK) packets governs the actions of the sender. This implies that network-based distortion in the forward data path will not be signaled back to the sender for one complete round-trip time interval, and the consequent adaptation of the sender to the conditions of the network will take numerous additional round-trip times. The implication is that in order to capture a comprehensive view of network performance, a time base of 1 to 2 seconds is appropriate. However, for large networks, such a view generates a massive amount of data. It appears that many networks use a measurement time base of about 60 to 300 seconds, representing an acceptable compromise between sensitivity of the measurement system and the consequent volume of measurement data to analyze.

What About QoS Networks?

So far the assumption has been that the network operates with a single service level and that probes of the network operate at the same service level as the network payload. This is certainly a common situation, but the total picture is slightly broader. When the network provider attempts to create a premium response for certain classes of traffic, and where the customer is paying a premium tariff to use such a premium service, the question of performance becomes a matter of deep concern to both the provider and the customer. After all, the customer is now paying a premium for improved performance, so it would help all concerned if this could be clearly defined and measured.

Solutions exist in both the passive and active polling domains. In the case of SNMP there is a monitoring framework (or Management Information Base, MIB) relating to the *Differentiated Services* (DiffServ) model of *Quality of Service* (QoS), and also MIBs relating to the *Integrated Services* (IntServ) QoS model. For the DiffServ MIB, it is first necessary to define an abstract model of the operation of a DiffServ admission router, by looking at the major functional blocks of the router. The first of these blocks is the definition of the supported behavior aggregates provided by the network. Within the network path, the initial active path element is the traffic classification module, which can be modeled as a set of filters and an associated set of output streams. The output stream is passed to the traffic-conditioning elements, which are the traffic meters and the associated action elements. Many meter profiles can be used in the model: an average data rate, an exponential weighted moving average of one of numerous various traffic profiles that can be expressed by a set of token-bucket parameters using an average rate, a peak rate, and a burst size. More elaborate meter specifications can be constructed using a multilevel token-bucket specification. From the meter, the traffic is passed through an action filter, which may mark the packets and shape the traffic profile through queues or discard operations. Together, this sequence of components forms a *traffic conditioning block*. The traffic is then passed into a queue through the use of a queuing discipline that applies the desired service behavior. (Figure 8)

Figure 8: DIffServ Control Architecture



From this generic model it is possible to define instrumentation for SNMP polling, where each of these five components—the behavior aggregate, the classifier, the meter, profile actions, and the queuing discipline—correspond to a MIB table. With this structure it is possible to parameterize both the specific configuration of the DiffServ network element and its dynamic state. This MIB is intended to describe the configuration and operation of both edge and interior DiffServ network elements, the difference being that interior elements use just a behavior aggregate classifier and a queue manager within the management model, whereas the edge elements use all components of the model.

A comparable MIB is defined for the IntServ architecture and an additional MIB for the operation of guaranteed services. The IntServ MIB defines the per-element reservation table used to determine the current reservation state, an indication of whether or not the router can accept further flow reservations, and the reservation characteristics of each current flow. No performance polling parameters or accounting parameters are included in the MIB. The guaranteed services MIB adds to this definition with a per-interface definition of a backlog. This is a means of expressing *packet quantization delay*, a delay term, which is the packet propagation delay over the interface, and a slack term, which is the amount of slack in the reservation that can be used without redefining the reservation. Again, these are per-element status definitions, and they do not include performance or accounting data items.

The IntServ MIB is being further defined as a *Resource Reservation Protocol* (RSVP) MIB for the operation of IntServ network elements[14]. There are a larger number of objects within the MIB, including General Objects, Session Statistics Table, Session Sender Table, Reservation Requests Received Table, Reservation Requests Forwarded Table, RSVP Interface Attributes Table, and an RSVP Neighbor Table.

Interestingly, the MIB proposes a writeable RSVP reservation table to allow the network manager to manually create a reservation state that can be removed only through a comparable manual operation. The MIB enables a management system to poll the IntServ network element to retrieve the status of every active IntServ reserved flow and the operational characteristics of the flow, as seen by the network element.

In a QoS DiffServ environment, ping and traceroute pose some interesting engineering issues. Ping sends an ICMP packet. The network QoS admission filters may choose a different classification for these packets from that chosen for normal data-flow TCP or UDP protocol packets; as a result, the probe packet may be scheduled differently or even take a completely different path to the network. In an IntServ QoS network, the common classification condition for a flow is a combination of the IP header source and destination addresses and the TCP or UDP header source and destination port addresses. The ping probe packet cannot reproduce this complete flow description, and therefore cannot, by default, be inserted into the flow path that it is attempting to measure. With traceroute, the packet does have a UDP protocol address, but it uses a constant port address by default, causing a similar problem of attempting to be inserted to an IntServ flow. DiffServ encounters similar problems when attempting to pass the probe packet into the network via the DiffServ admission classification systems. Inside the network, it is possible to insert the probe packet into the network with the IP *Differentiated Services Code Point* (DSCP) field set to the DiffServ behavior aggregate that is being measured.

The measurement of delay and loss taken by ping and traceroute is a cumulative value of both the forward and return path delay and loss. When attempting to measure unidirectional flow-path behavior, such as an IntServ flow path, this measurement is of dubious value, given the level of uncertainty as to which part of the path, forward or reverse, contributed to the ping or traceroute delay and loss reports.

For one-way delay measurements, in DiffServ networks, this can be done within the network, setting the DSCP field to the value of the service aggregate being monitored. Of course, from the customer's perspective, the DiffServ network service profile includes the admission traffic-conditioning block, and the interior one-way measurements are only part of the delivered service. In the IntServ network, the packets have to be structured to take the same path as the elevated service flows; they are classified by each element as part of the collection of such elevated service flows for the purposes of scheduling.

Measuring Performance—The Client Perspective

From the client's perspective, the measurement choices are more limited. A client does not normally enjoy the ability to poll network elements within a provider's network. One way for a client to measure service quality is to instigate probing of the network path, whereby a sender can pass a probe packet into the network and measure the characteristics of the response. Of course, the problems of inserting probe packets into the service flow remain, as do the issues of unidirectional elevated service flows with bidirectional probes.

However, the client does have the advantage of being able to monitor and manipulate the characteristics of the service flow itself. For TCP sessions, the client can monitor the packet retransmission rate, the maximum burst capacity, the average throughput, the *round-trip time* (RTT), RTT variance, and misordered packets, by monitoring the state of the outbound data flow and relating it to the inbound ACK flow. For UDP sessions, there is no corresponding transport-level feedback information flow to the sender as a part of the transport protocol itself. The receiver can measure the service quality of the received datastream using information provided in the *Real-Time Protocol* (RTP) information feedback fields—if RTP is being used for real-time data or as an application-related tool for other application types. If sender and receiver work in concert, the receiver can generate periodic quality reports and pass these summaries back to the sender. Such applications can confirm whether an application is receiving a specified level of service. This approach treats the network like a black box; no attempt is made to identify the precise nature or source of events that disrupt the delivered service quality. There are no standardized approaches to this activity, but numerous analysis tools are available for host platforms that perform these measurements.

Though the client can measure and conform service quality on a per-application level of granularity, the second part of the client's motivation in measuring service quality is more difficult to address. The basic question is whether the service delivered in response to a premium service request is sufficiently differentiated from a best-effort service transaction. Without necessarily conducting the transaction a second time, the best approach is to use either one-way delay probes, for unidirectional traffic, or a bulk TCP capacity probe, to establish some indication of the relativity in performance. From a client perspective none of these are simple to set up, and the dilemma that the customer often faces is the basic question of whether the cost of operating the measurement setup is adequately offset by the value of the resulting answers.

Measuring Networks—Looking for Problems

So far we have been looking at the ways of measuring network performance as a general task. Of course degraded performance does not happen by accident (well, sometimes accidents do happen), and it makes the measurement task easier if you can identify precisely what it is that you are looking for. This approach requires identification of the various situations that can impact network performance and then set up network measurement and monitoring systems that are tuned to identify these situations.

Within this approach, the motives for network measurement are concerned with identification of traffic load patterns that cause uneven network load, monitoring, and verification of service-level agreements, detection of abnormal network load that may be a signature of an attack, forecasting and capacity planning, and routing stability.

The objective here is to create a stable and well-understood model of the operational characteristics of the network, and then analyze the situations that could disrupt this stable state and the implications in terms of delivered performance under such conditions.

Such an approach could be described in terms of opposites—instead of measuring network performance, the approach is measuring the network to identify the conditions that cause nonperformance at particular times within particular network paths. As a performance management technique, this approach has been very effective—rather than taking a larger amount of performance data and merging and averaging it into a relatively meaningless index, the approach is to isolate those circumstances where performance is compromised and report on these exceptions rather than on the remainder of the time.

Of course measuring what is "normal" may involve more than assembling a benchmark set of SNMP-derived polling data and a collection of latency, loss, and jitter profiles obtained from analysis of large volumes of ping data. One additional tool is the router itself. Because the router uses many IP packet header fields to switch each packet, one approach is to get the router to assemble and aggregate information about the characteristics of traffic that has been passed through the router, and send these aggregated reports to a network management station for further analysis. *NetFlow* is the most common tool to undertake this form of reporting. Like SNMP, NetFlow can report on the characteristics of traffic as it passes a point in the network. For measuring end-to-end performance of individual applications, NetFlow has the same limitations as SNMP. The analogy is one of standing on a street corner counting cars that go past and from that measurement attempting to derive the average time for a commuter to drive to or from work. However, the value of NetFlow is that in this context of performance measurement, it can be used to derive a picture of the baseline characteristics of the network, including identification of the endpoints of the traffic flows. Extending the car analogy further, NetFlow can provide an indication of the origins and ultimate destinations of the cars as they pass the monitoring point. This information is useful in terms of designing networks that are adequately configured to handle the transit traffic load. In addition, with careful analysis, NetFlow can be used to identify exceptional traffic conditions. The advantage here is that NetFlow data can be used to identify both the abnormal traffic load and also provide some indication of the endpoints of the abnormal flows. In this way, NetFlow can be deployed as both a baseline network traffic profile benchmarking tool and a performance exception diagnosis tool.

This approach of capturing the packet header information as the traffic passes a monitoring point in the network has been implemented in numerous ways, and NetFlow is not the only data-collection tool in this space. One interesting approach has been used by NeTraMet, an implementation of the *Internet Engineering Task Force's* (IETF's) *Realtime Traffic Flow Measurement* architecture for traffic flow measurement.

The feature here is a powerful ruleset within the tool that allows the flow collector to be configured to collect information about particular traffic flows and their characteristics. In the context of measuring performance, one of the abilities of the tool is to match the outbound data flow with the inbound acknowledgement stream, allowing an analyzer some ability to infer end-to-end performance of the application based on the collected information.

Where to Go from Here

It is clear that the picture is so far very incomplete. The active probe measurements require either some latitude of interpretation or dedicated instrumentation to take measurements with some necessary level of frequency and precision. The passive approach of probing the active switching elements of the network is constrained by a very basic model of the switching system, so that the collectable values provide only a very indirect relationship to the manner in which the switching element is generating queuing delays and traffic flow instability.

Perhaps what is also increasingly unclear is the relationship between performance and networks in any case. The last few years have seen a massive swing in public Internet platforms away from networks where some level of congestion and contention was anticipated to networks that are extensively overprovisioned, and there packet jitter and loss are simply not encountered. With the ever-decreasing cost of transmission bandwidth in many markets, this environment of abundant network capacity is now also finding its way into various enterprise network sectors. In such worlds of abundant supply and overengineering of networks, there is really little left to measure within the network. The entire question of performance then becomes a question phrased much closer to home: how well is your system tuned to make the most of its resources and those of the server? Often the entire issue with performance is a situation of abundant network resources, abundant local memory and processing resources, and poor tuning of the transport protocol stack. That is, of course, quite properly the subject of another article.

Further Reading

The Internet offers a wealth of material on the topic of network measurement, and the major exercise is undertaking some filtering to get a broad collection of material that encompasses a range of perspectives on this topic. The following sources were used to prepare this article, and are recommended as starting points for further exploration of this topic.

[1] *Internet Performance Survival Guide,* Geoff Huston, Wiley Computer Publishing, 2000.

[2] "IPPM Metrics for Measuring Connectivity," J. Mahdavi, V. Paxson, RFC 2678, September 1999.

[3] "A One-way Delay Metric for IPPM," G. Almes, S. Kalidinki, M. Zeukuaskas, RFC 2679, September 1999.

[4] "A One-way Packet Loss Metric for IPPM," G. Almes, S. Kalidinki, M. Zeukuaskas, RFC 2680, September 1999.

[5] The RIPE Test Traffic Measurement service at:
http://www.ripe.net/ripencc/mem-services/ttm/

[6] Treno, online at:
http://www.psc.edu/networking/treno_info.html

[7] "Trends in Measurement and Monitoring of Internet Backbones," session at the 26th North American Network Operators Group, hosted by D. Meyer,
http://www.nanog.org/mtg-0210/measurement.html,
October 2002.

[8] "Some thoughts on CoS and Backbone Networks," D. Meyer, presentation to the IEPREP Working Group, IETF-55,
http://www.maoz.com/~dmm/IETF55/ieprep/, November 2002.

[9] NetFlow resource page:
http://www.cisco.com/warp/public/732/Tech/nmp/netflow/netflow_techdoc.shtml

[10] Netramet, and many other interesting measurement tools are referenced in a resource page at: **http://www.caida.org/tools**

This area of research is active, and numerous activities are ongoing in the area of research group activities and workshops.

[11] The Internet Research Task Force has an Internet Measurement Research Group. Further details can be found at:
http://www.irtf.org/charters/imrg.html

[12] ACM SIGCOMM, the ACM Special Interest Group on Data Communications, sponsors an Internet Measurement Workshop. Proceeding of the November 2002 workshop can be found at:
http://www.acm.org/sigcomm/imw2002/

[13] The details of the 2003 Passive and Active Measurement Workshop can be found at: **http://www.pam2003.org**

[14] "RSVP Management Information Base using SMIv2," F. Baker, J. Krawczyk, A. Sastry, RFC 2206, September 1997.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Scientist in the Internet area for Telstra. He is also a member of the Internet Architecture Board, and is the Secretary of the APNIC Executive Committee. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@telstra.net**

The Session Initiation Protocol

by William Stallings

The *Session Initiation Protocol* (SIP), defined in RFC 3261[6], is an application level signaling protocol for setting up, modifying, and terminating real-time sessions between participants over an IP data network. SIP can support any type of single-media or multimedia session, including teleconferencing.

SIP is just one component in the set of protocols and services needed to support multimedia exchanges over the Internet. SIP is the signaling protocol that enables one party to place a call to another party and to negotiate the parameters of a multimedia session. The actual audio, video, or other multimedia content is exchanged between session participants using an appropriate transport protocol. In many cases, the transport protocol to use is the *Real-Time Transport Protocol* (RTP). Directory access and lookup protocols are also needed.

The key driving force behind SIP is to enable Internet telephony, also referred to as *Voice over IP* (VoIP). There is wide industry acceptance that SIP will be the standard IP signaling mechanism for voice and multimedia calling services. Further, as older *Private Branch Exchanges* (PBXs) and network switches are phased out, industry is moving toward a voice networking model that is SIP signaled, IP based, and packet switched, not only in the wide area but also on the customer premises[2, 3].

SIP supports five facets of establishing and terminating multimedia communications:

- *User location:* Users can move to other locations and access their telephony or other application features from remote locations.
- *User availability:* This step involves determination of the willingness of the called party to engage in communications.
- *User capabilities:* In this step, the media and media parameters to be used are determined.
- *Session setup:* Point-to-point and multiparty calls are set up, with agreed session parameters.
- *Session management:* This step includes transfer and termination of sessions, modifying session parameters, and invoking services.

SIP employs design elements developed for earlier protocols. SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a client request that invokes a particular method, or function, on the server and at least one response. SIP uses most of the header fields, encoding rules, and status codes of HTTP. This provides a readable text-based format for displaying information. SIP incorporates the use of a *Session Description Protocol* (SDP), which defines session content using a set of types similar to those used in *Multipurpose Internet Mail Extensions* (MIME).

SIP Components and Protocols

A system using SIP can be viewed as consisting of components defined on two dimensions: client/server and individual network elements. RFC 3261 defines client and server as follows:

- *Client:* A client is any network element that sends SIP requests and receives SIP responses. Clients may or may not interact directly with a human user. User agent clients and proxies are clients.
- *Server:* A server is a network element that receives requests in order to service them and sends back responses to those requests. Examples of servers are proxies, user agent servers, redirect servers, and registrars.

The individual elements of a standard SIP configuration include the following:

- *User Agent:* The user agent resides in every SIP end station. It acts in two roles:
  - User Agent Client (UAC): Issues SIP requests
  - User Agent Server (UAS): Receives SIP requests and generates a response that accepts, rejects, or redirects the request
- *Redirect Server:* The redirect server is used during session initiation to determine the address of the called device. The redirect server returns this information to the calling device, directing the UAC to contact an alternate *Universal Resource Identifier* (URI). A URI is a generic identifier used to name any resource on the Internet. The URL used for Web addresses is a type of URI. See RFC 2396[1] for more detail.
- *Proxy Server:* The proxy server is an intermediary entity that acts as both a server and a client for the purpose of making requests on behalf of other clients. A proxy server primarily plays the role of routing, meaning that its job is to ensure that a request is sent to another entity closer to the targeted user. Proxies are also useful for enforcing policy (for example, making sure a user is allowed to make a call). A proxy interprets, and, if necessary, rewrites specific parts of a request message before forwarding it.
- *Registrar:* A registrar is a server that accepts REGISTER requests and places the information it receives (the SIP address and associated IP address of the registering device) in those requests into the location service for the domain it handles.
- *Location Service:* A location service is used by a SIP redirect or proxy server to obtain information about a callee's possible location(s). For this purpose, the location service maintains a database of SIP-address/IP-address mappings.

The various servers are defined in RFC 3261 as logical devices. They may be implemented as separate servers configured on the Internet or they may be combined into a single application that resides in a physical server.

Figure 1: SIP Components and Protocols

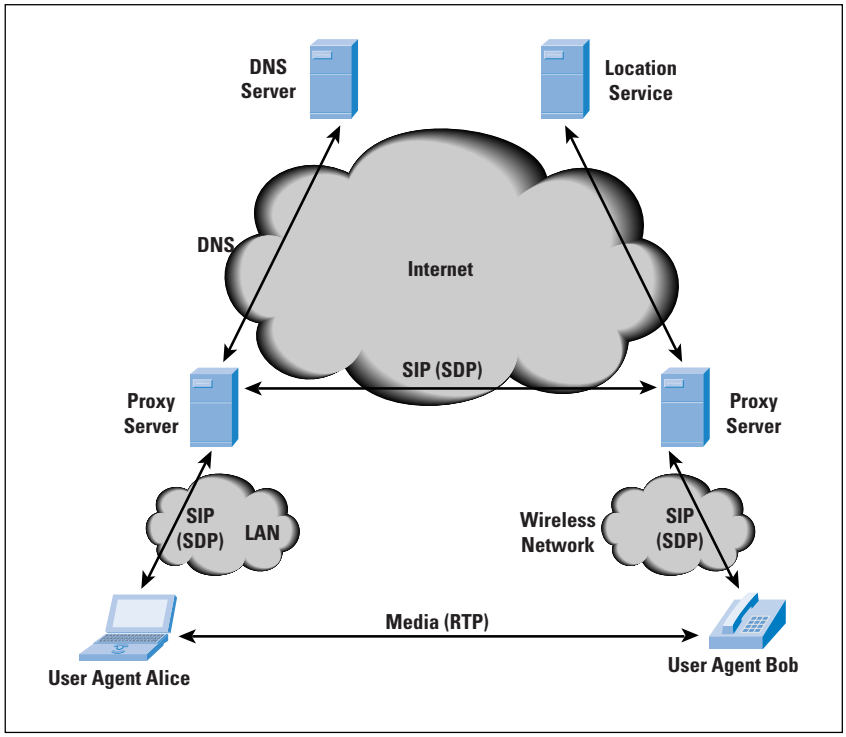


Figure 1 shows how some of the SIP components relate to one another and the protocols that are employed. A user agent acting as a client (in this case UAC Alice) uses SIP to set up a session with a user agent that acts as a server (in this case UAS Bob). The session initiation dialogue uses SIP and involves one or more proxy servers to forward requests and responses between the two user agents. The user agents also make use of the SDP, which is used to describe the media session.

The proxy servers may also act as redirect servers as needed. If redirection is done, a proxy server needs to consult the location service database, which may or may not be colocated with a proxy server. The communication between the proxy server and the location service is beyond the scope of the SIP standard. The *Domain Name System* (DNS) is also an important part of SIP operation. Typically, a UAC makes a request using the domain name of the UAS, rather than an IP address. A proxy server needs to consult a DNS server to find a proxy server for the target domain.

SIP often runs on top of the *User Datagram Protocol* (UDP) for performance reasons, and provides its own reliability mechanisms, but may also use TCP. If a secure, encrypted transport mechanism is desired, SIP messages may alternatively be carried over the *Transport Layer Security* (TLS) protocol.

Associated with SIP is the SDP, defined in RFC 2327[4]. SIP is used to invite one or more participants to a session, while the SDP-encoded body of the SIP message contains information about what media encodings (for example, voice, video) the parties can and will use. After this information is exchanged and acknowledged, all participants are aware of the participants' IP addresses, available transmission capacity, and media type. Then, data transmission begins, using an appropriate transport protocol. Typically, the RTP is used. Throughout the session, participants can make changes to session parameters, such as new media types or new parties to the session, using SIP messages.

SIP Universal Resource Indicators

A resource within a SIP configuration is identified by a URI. Examples of communications resources include the following:

- A user of an online service
- An appearance on a multiline phone
- A mailbox on a messaging system
- A telephone number at a gateway service
- A group (such as "sales" or "help desk") in an organization

SIP URIs have a format based on e-mail address formats, namely **`user@domain`**. There are two common schemes. An ordinary SIP URI is of the form:

```
sip:bob@biloxi.com
```

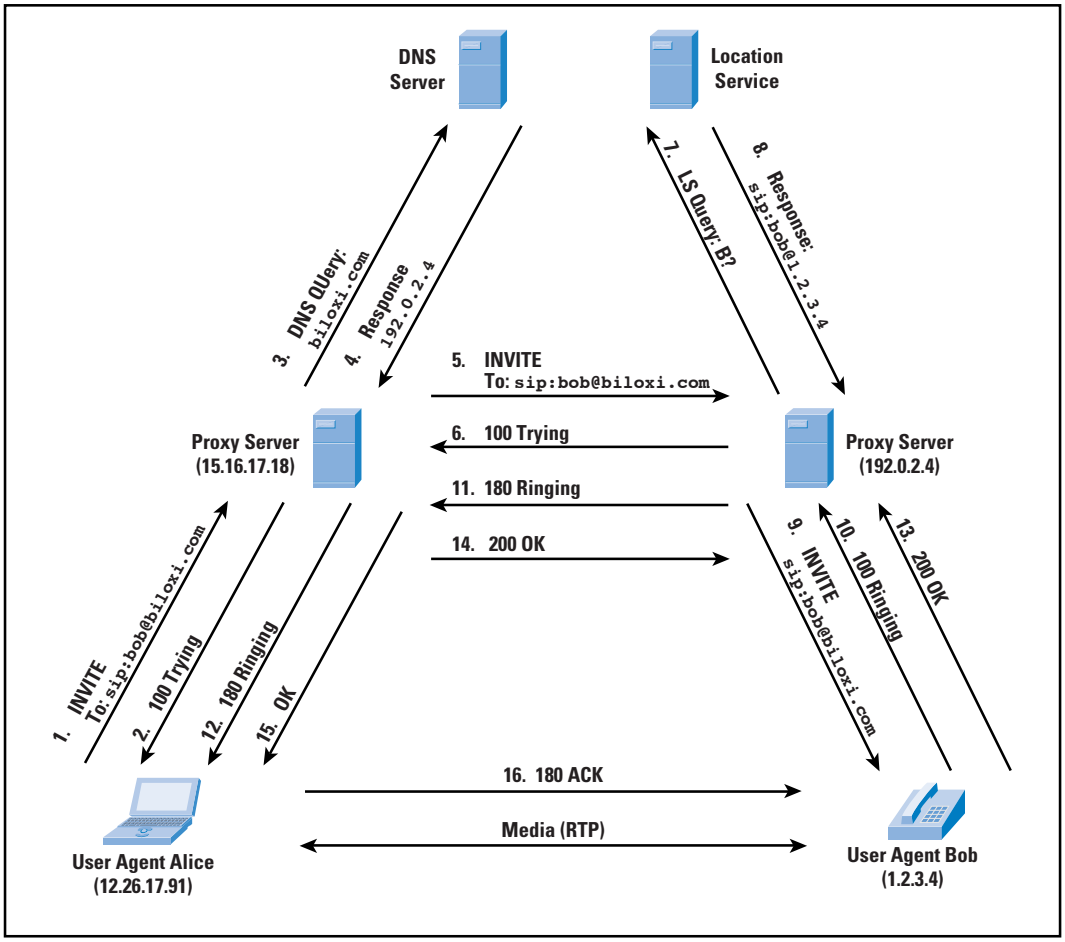
The URI may also include a password, port number, and related parameters. If secure transmission is required, "**`sip:`**" is replaced by "**`sips:.`**" In the latter case, SIP messages are transported over TLS.

Examples of Operation

The SIP specification is quite complex; the main document, RFC 3261, is 269 pages long. To give some feel for its operation, we present a few examples.

Figure 2 shows a successful attempt by user Alice to establish a session with user Bob, whose URI is **`bob@biloxi.com.`**[9] Alice's UAC is configured to communicate with a proxy server (the outbound server) in its domain and begins by sending an INVITE message to the proxy server that indicates its desire to invite Bob's UAS into a session (1); the server acknowledges the request (2). Although Bob's UAS is identified by its URI, the outbound proxy server needs to account for the possibility that Bob is not currently available or that Bob has moved. Accordingly, the outbound proxy server should forward the INVITE request to the proxy server that is responsible for the domain **`biloxi.com.`** The outbound proxy thus consults a local DNS server to obtain the IP address of the **`biloxi.com`** proxy server (3), by asking for the DNS SRV resource record that contains information on the proxy server for **`biloxi.com.`**

Figure 2: SIP Successful Call Setup

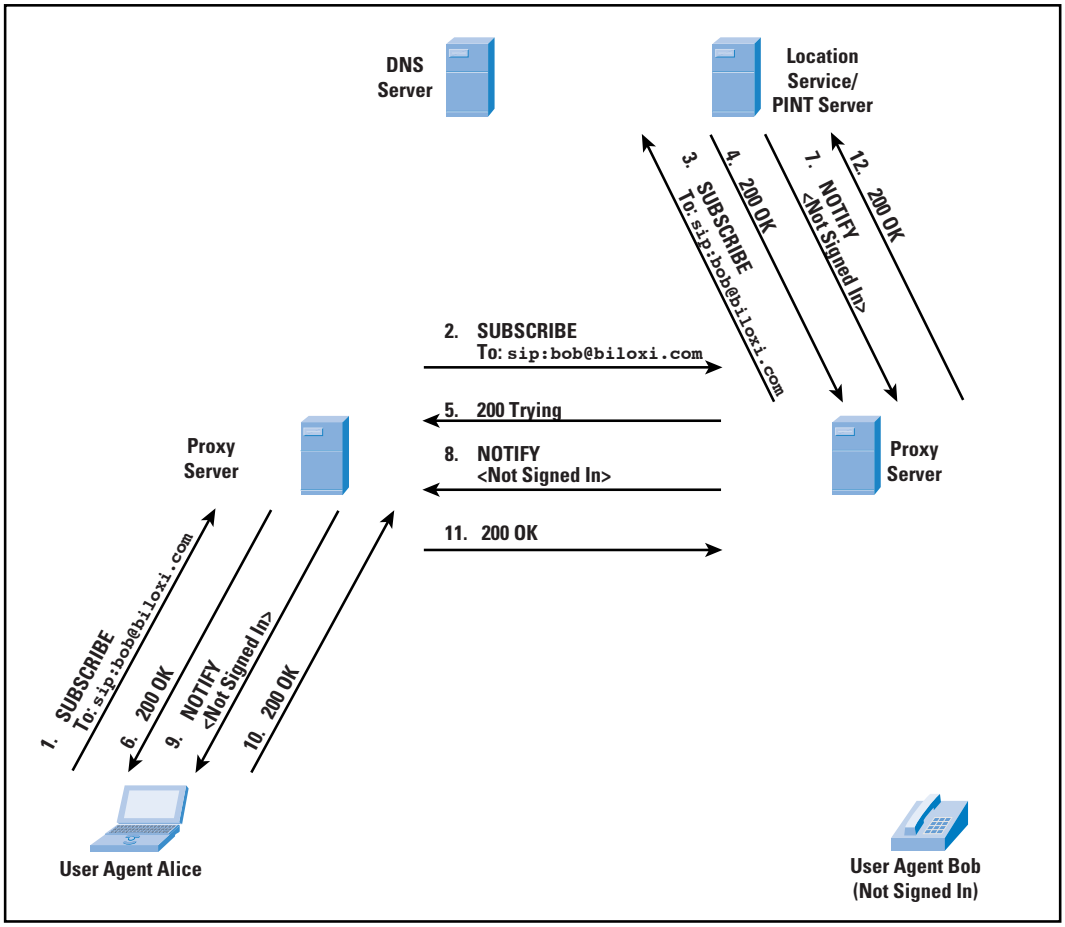


The DNS server responds (4) with the IP address of the **`biloxi.com`** proxy server (the inbound server). Alice's proxy server can now forward the INVITE message to the inbound proxy server (5), which acknowledges the message (6). The inbound proxy server now consults a location server to determine Bob's location Bob (7), and the location server responds with Bob's location, indicating that Bob is signed in, and therefore available for SIP messages (8).

The proxy server can now send the INVITE message on to Bob (9). A ringing response is sent from Bob back to Alice (10, 11, 12) while the UAS at Bob is alerting the local media application (for example, telephony). When the media application accepts the call, Bob's UAS sends back an OK response to Alice (13, 14, 15).

Finally, Alice's UAC sends an acknowledgement message to Bob's UAS to confirm the reception of the final response (16). In this example, the ACK is sent directly from Alice to Bob, bypassing the two proxies. This occurs because the endpoints have learned each other's address from the INVITE/200 (OK) exchange, which was not known when the initial INVITE was sent. The media session has now begun, and Alice and Bob can exchange data over one or more RTP connections.

Figure 3: SIP Presence Example

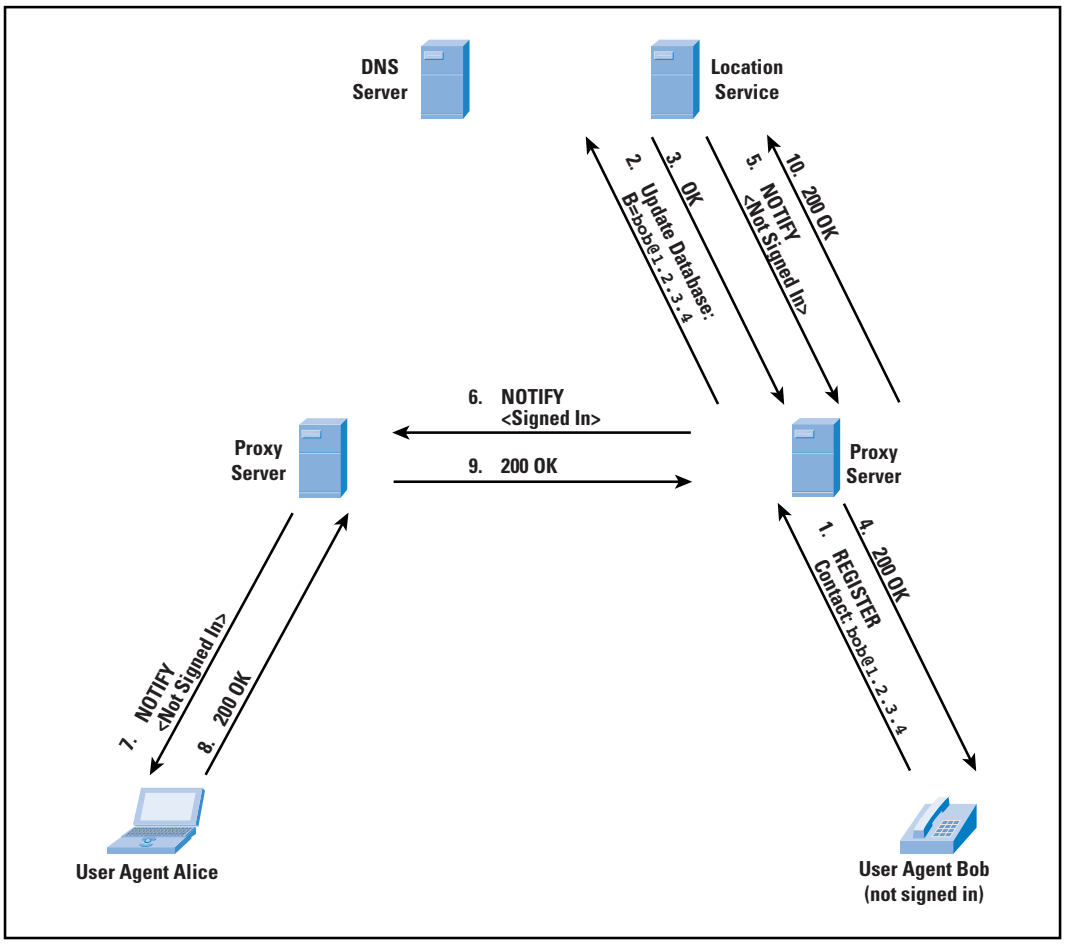


The next example (Figure 3) makes use of two message types that are not yet part of the SIP standard but that are documented in RFC 2848[5] and are likely to be incorporated in a later revision of SIP. These message types support telephony applications. Suppose that in the preceding example, Alice was informed that Bob was not available. Alice's UAC can then issue a SUBSCRIBE message (1), indicating that it wants to be informed when Bob is available.

This request is forwarded through the two proxies in our example to a PINT (*Public Switched Telephone Network [PSTN]-Internet Networking*) server (2, 3). A PINT server acts as a gateway between an IP network from which comes a request to place a telephone call and a telephone network that executes the call by connecting to the destination telephone. In this example, we assume that the PINT server logic is colocated with the location service. It could also be the case that Bob is attached to the Internet rather than a PSTN, in which case the equivalent of PINT logic is needed to handle SUBSCRIBE requests. In this example, we assume the latter and assume that the PINT functionality is implemented in the location service. In any case, the location service authorizes subscription by returning an OK message (4), which is passed back to Alice (5, 6). The location service then immediately sends a NOTIFY message with Bob's current status of not signed in (7, 8, 9), which Alice's UAC acknowledges (10, 11, 12).

Figure 4 continues the example of Figure 3. Bob signs on by sending a REGISTER message to the proxy in its domain (1). The proxy updates the database at the location service to reflect registration (2). The update is confirmed to the proxy (3), which confirms the registration to Bob (4). The PINT functionality learns of Bob's new status from the location server (here we assume that they are colocated) and sends a NOTIFY message containing Bob's new status (5), which is forwarded to Alice (6, 7). Alice's UAC acknowledges receipt of the notification (8, 9, 10).

Figure 4: SIP Registration and Notification Example



SIP Messages

As was mentioned, SIP is a text-based protocol with a syntax similar to that of HTTP. There are two different types of SIP messages, *requests* and *responses*. The format difference between the two types of messages is seen in the first line. The first line of a request has a method, defining the nature of the request and a Request-URI, indicating where the request should be sent. The first line of a response has a response code. All messages include a header, consisting of a number of lines, each line beginning with a header label. A message can also contain a body such as an SDP media description.

For SIP requests, RFC 3261 defines the following methods:

- *REGISTER:* Used by a user agent to notify a SIP configuration of its current IP address and the URLs for which it would like to receive calls
- *INVITE:* Used to establish a media session between user agents
- *ACK:* Confirms reliable message exchanges
- *CANCEL:* Terminates a pending request, but does not undo a completed call
- *BYE:* Terminates a session between two users in a conference
- *OPTIONS:* Solicits information about the capabilities of the callee, but does not set up a call

For example, the header of message (1) in Figure 2 might look like the following:

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP 12.26.17.91:5060
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com
From: Alice <sip:alice@atlanta.com;tag=1928301774
Call-ID: a84b4c76e66710@12.26.17.91
CSeq: 314159 INVITE
Contact: <sip:alice@atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```

The first line contains the method name (**INVITE**), a SIP URI, and the version number of SIP that is used. The lines that follow are a list of header fields. This example contains the minimum required set.

The **Via** headers show the path the request has taken in the SIP configuration (source and intervening proxies), and are used to route responses back along the same path. As the INVITE message leaves, there is only the header inserted by Alice. The line contains the IP address (**12.26.17.91**), port number (**5060**), and transport protocol (**UDP**) that Alice wants Bob to use in his response.

The **Max-Forwards** header limits the number of hops a request can make on the way to its destination. It consists of an integer that is decremented by one by each proxy that forwards the request. If the **Max-Forwards** value reaches 0 before the request reaches its destination, it is rejected with a 483 (**Too Many Hops**) error response.

The **To** header field contains a display name (Bob) and a SIP or SIPS URI (**sip:bob@biloxi.com**) toward which the request was originally directed. The **From** header field also contains a display name (Alice) and a SIP or SIPS URI (**sip:alice@atlanta.com**) that indicate the originator of the request. This header field also has a **tag** parameter that contains a random string (**1928301774**) that was added to the URI by the UAC. It is used to identify the session.

The **`Call-ID`** header field contains a globally unique identifier for this call, generated by the combination of a random string and the host name or IP address. The combination of the **`To`** tag, **`From`** tag, and **`Call-ID`** completely defines a peer-to-peer SIP relationship between Alice and Bob and is referred to as a dialog.

The **`CSeq`** or *Command Sequence* header field contains an integer and a method name. The CSeq number is initialized at the start of a call (**`314159`** in this example), incremented for each new request within a dialog, and is a traditional sequence number. The CSeq is used to distinguish a retransmission from a new request.

The **`Contact`** header field contains a SIP URI for direct communication between user agents. Whereas the **`Via`** header field tells other elements where to send the response, the **`Contact`** header field tells other elements where to send future requests for this dialog.

The **`Content-Type`** header field indicates the type of the message body. The **`Content-Length`** header field gives the length in octets of the message body.

The SIP response types defined in RFC 3261 are in the following categories:

- *Provisional* (1xx): The request was received and is being processed.
- *Success* (2xx): The action was successfully received, understood, and accepted.
- *Redirection* (3xx): Further action needs to be taken in order to complete the request.
- *Client Error* (4xx): The request contains bad syntax or cannot be fulfilled at this server.
- *Server Error* (5xx): The server failed to fulfill an apparently valid request.
- *Global Failure* (6xx): The request cannot be fulfilled at any server.

For example, the header of message (13) in Figure 2 might look like the following:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
Via: SIP/2.0/UDP 12.26.17.91:5060
To: Bob <sip:bob@biloxi.com;tag=a6c85cf
From: Alice <sip:alice@atlanta.com;tag=1928301774
Call-ID: a84b4c76e66710@12.26.17.91
CSeq: 314159 INVITE
Contact: <sip:bob@biloxi.com>
Content-Type: application/sdp
Content-Length: 131
```

The first line contains the version number of SIP that is used and the response code and name. The lines that follow are a list of header fields. The **`Via`**, **`To`**, **`From`**, **`Call-ID`**, and **`CSeq`** header fields are copied from the INVITE request. (There are three **`Via`** header field values—one added by Alice's SIP UAC, one added by the **`atlanta.com`** proxy, and one added by the **`biloxi.com`** proxy.) Bob's SIP phone has added a **`tag`** parameter to the **`To`** header field. This tag is incorporated by both endpoints into the dialog and is included in all future requests and responses in this call.

Session Description Protocol

The *Session Description Protocol* (SDP), defined in RFC 2327, describes the content of sessions, including telephony, Internet radio, and multimedia applications. SDP includes information about[8]:

- *Media streams:* A session can include multiple streams of differing content. SDP currently defines audio, video, data, control, and application as stream types, similar to the MIME types used for Internet mail.
- *Addresses:* SDP indicates the destination addresses, which may be a multicast address, for a media stream.
- *Ports:* For each stream, the UDP port numbers for sending and receiving are specified.
- *Payload types:* For each media stream type in use (for example, telephony), the payload type indicates the media formats that can be used during the session.
- *Start and stop times:* These apply to broadcast sessions, for example, a television or radio program. The start, stop, and repeat times of the session are indicated.
- *Originator:* For broadcast sessions, the originator is specified, with contact information. This may be useful if a receiver encounters technical difficulties.

Although SDP provides the capability to describe multimedia content, it lacks the mechanisms by which two parties agree on the parameters to be used. RFC 3264[7] remedies this lack by defining a simple offer/answer model, by which two parties exchange SDP messages to reach agreement on the nature of the multimedia content to be transmitted.

References

[1] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, August 1998.

[2] S. Borthick, "SIP Services: Slowly Rolling Forward," *Business Communications Review,* June 2002.

[3] S. Borthick, "SIP for the Enterprise: Work in Progress," *Business Communications Review,* February 2003.

[4] M. Handley and V. Jacobson, "SDP: Session Description Protocol," RFC 2327, April 1998.

[5] S. Petrack and L. Conroy, "The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services," RFC 2848, June 2000.

[6] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, June 2002.

[7] J. Rosenberg and H. Schulzrinne, "An Offer/Answer Model with the Session Description Protocol," RFC 3264, June 2002.

[8] H. Schulzrinne and J. Rosenberg, "The Session Initiation Protocol: Providing Advanced Telephony Access Across the Internet," *Bell Labs Technical Journal,* October-December 1998.

[9] Figures 2 through 4 are adapted from ones developed by Professor H. Charles Baker of Southern Methodist University.

WILLIAM STALLINGS is a consultant, lecturer, and author of over a dozen books on data communications and computer networking. He also maintains a computer science resource site for CS students and professionals at **WilliamStallings.com/StudentSupport.html**. He has a PhD in computer science from M.I.T. His latest book is *Computer Networks, with Internet Protocols and Technology* (Prentice Hall, 2003). His home in cyberspace is **WilliamStallings.com** and he can be reached at **ws@shore.net**

Letters to the Editor

Ruling the Root

Ole,

As a matter of principle, I don't mind having my book *Ruling the Root* reviewed by David Crocker. Mr. Crocker was a significant figure in some of the key events covered in the book. His assessment and opinion of the book had the potential to be quite interesting.

One can only be disappointed with the results, however. The review reveals an inability to rise above partisan sniping and engage rationally with an different view. That, as a matter of policy, is why serious journals don't publish unsolicited reviews of books. Unsolicited reviewers tend to fall into one of two types: unabashed promoters with a personal interest in the success of the book, or people with an axe to grind trying to shoot down a perceived enemy.

I offer a rebuttal only because I think it is vital that the Internet technical community, the presumed readers of *The Internet Protocol Journal,* achieve a higher standard in their discussion of Internet-related policy issues.

Ruling the Root is a serious attempt to analyze the intersection of technology and policy. It offers a way of understanding that intersection based on theories of institutions and property rights. I know that this intersection irritates many engineers, who often harbor a wish that it would go away. By now we should know that it won't. Technical systems raise political issues. Technical people, economists, lawyers, and policy analysts, therefore, must be able to engage in rational dialogue about institutional issues, even when the discussion comes uncomfortably close to home. If we can't, the world is in big trouble.

The review completely misses this big picture. It begins with an attempt to belittle the policy significance of domain name management by inventing a mythical decree that all street names have to be in an obscure language. Crocker's attempt at humor falls flat, given today's headlines. Virtually the same day his review was published a German registrar was ordered to take a domain name away from a Web site with objectionable content. Not too long after, an ICANN Task Force published a WHOIS policy proposal that allows domain names to be shut down after 15 days if someone challenges the accuracy of the contact information, raising issues of privacy and harassment. ICANN regulates the prices of registries and entry into the market for domain name services. No one, not even ICANN itself these days, pretends that domain name administration is an exclusively technical matter.

Instead of engaging on those terms, the review concentrated on factual nitpicking. Take this one: "...the book does not consider NSI's role in ICANN-related political processes." This is an astoundingly inaccurate statement. The index of the book under "Network Solutions" contains 33 listings under 5 separate headings.

The book analyzes at length NSI's origins and ownership changes, its opposition to the IAHC and gTLD-MoU, its implied threat to establish a new root, and its policy conflicts with ICANN and the U.S. Department of Commerce.

Crocker claims that I "[characterize] the pre-ICANN *International Forum for the White Paper* (IFWP) as 'the real arena for arriving at a decision [about the details of the new organization].'" His use of a sentence fragment covers up what appears to be a deliberate distortion. I really wrote that some people viewed the IFWP in that way, while others, notably Joe Sims, Jon Postel, and the Information Technology Association of America, did not; see pages 176–178. I wrote at length about how that basic lack of agreement between adherents of IFWP and followers of IANA over legitimacy led to lasting conflict over ICANN's formation.

Crocker was one of Jon Postel's appointees to the *International Ad Hoc Committee* (IAHC). The review takes issue with my characterization of the IAHC, but unfortunately only to maintain Crocker's fictional self-conceptions. He denies that the IAHC ever claimed that "the root was theirs to dispose of." He also denies that IAHC was intended to be the seed of an alternative DNS governance structure. He's wrong on both counts. There is a voluminous record on this question, comprising contemporary news accounts, e-mail list archives, and my own recorded interviews with principal figures such as Don Heath.

Crocker's assertion that IAHC was "explicitly subordinate to IANA" is rather disingenuous, because IANA's U.S. government funding was ending and IAHC was explicitly perceived by Postel and ISOC as a mechanism for continuing its funding. So IAHC was intended to be the governance and support structure for IANA, just as ICANN now is. Indeed, today's ICANN has many features in common with the IAHC proposal, such as the shared registry concept, the slant toward intellectual property interests, the treatment of TLDs as "public resources," and a compulsory and uniform dispute resolution procedure.

What is really at issue here? It is this: Crocker cannot accept the simple fact that a political battle was under way for control of the root, and Postel/IAHC, as well as NSI and the U.S. government, were contenders for that control. Crocker's review challenges the claim in the book that Postel's root redirection exercise in January 1998 was "apparently" based on "concerns about the direction U.S. policy was taking." This judgment was based on interviews with people who were involved with Postel's effort. Of course I cannot read Postel's mind, but neither can David Crocker. My interpretation of why Postel acted is based on the timing and on evidence drawn from first-hand participants. Crocker offers an alternative interpretation, plausible but based on nothing but his own assertion. There is plenty of room for legitimate debate about historical interpretations. Such debate is useful, however, only if it is aimed at discovering the truth.

Regarding the status of IANA, I am sure we will never agree. I see it fundamentally as a DARPA contractor subject to U.S. governmental authority; Crocker views it in almost mystical terms as the embodiment of the Internet community. He says nothing about who paid the bills. Yet, we are not as far apart on the facts as he wants to make it seem. Contrary to the review, the book does document in great detail how a new community for Internet standards development grew up around the old DARPA-funded cadre of Postel, Cerf, and the IAB, and created its own standards of legitimacy and process. My book doesn't dispute Postel's tremendous respect and legitimacy among the technical community. But when it comes to institutionalizing control and ownership of the name and address roots of the Internet, whoever pays the piper calls the tune. And Postel's ability to perform the IANA functions was supported by U.S. government money from day one.

Hence, it was unrealistic to expect Postel to be exempt from governmental authority after domain names became resources of economic value and produced legal and political conflict over that value. Nor is it correct to imply, as Crocker does, that knowledge of the operational details of a technology automatically confers wisdom as to the correct public policies that should be adopted when that happens. Of course, policy decisions must respect technical facts and technical constraints. It is this relationship between technical system, technical community, and the worlds of business, law, and government that is cntral to the story told by *Ruling the Root*.

Crocker's final stab at discrediting the book involves some rather spurious charges of ethics problems. "In his criticism of dispute-resolution activities, he neglects to mention that he is a paid arbitration panelist," he writes. Crocker here refers to the fact that I was one of the few nonlawyers allowed by WIPO to serve as one of three judges in domain name—trademark disputes brought under ICANN's UDRP. The "pay" he refers to is a $500 or $750 honorarium for each case. I do about ten cases a year. I fail to see any conflict of interest or ethical problem here. Crocker implies that my meager remuneration for assuring that justice is done in UDRP cases somehow corrupts me, but he knows perfectly well that I am an opponent of the UDRP and would happily stop receiving those honoraria if the darn thing went away. Besides, no one is in a better position to understand what is right and what is wrong with UDRP than someone who is involved in the actual cases. I do not even understand what his concern is about the noncommercial DNSO constituency. I deal with it in one sentence in the book, and most of my activity in a "management capacity" (i.e., as an elective representative) came after the book manuscript was written.

—Milton Mueller, Syracuse University
Mueller@syr.edu

The author of the book review responds:

Professor Mueller's response discusses his goals of the book and his opinions of my review, to which he is, of course, entitled. He characterizes *Ruling the Root* as an academic consideration of the policy issues pertaining to the Domain Name Service, which he casts as global Internet administrative services. Note that the tag line to the title of his book, however, casts it more even more generally as "Internet governance." Academic and policy work need to be conducted carefully. Unfortunately, Professor Mueller confuses the issues, rather than elucidating them.

The opening, mythical decree of the review was carefully constructed to make the perspective of the book on communication system administrative policy clear: Professor Mueller confuses an administrative agency, such as ICANN or its telephonic equivalent, with a national government such as Germany. He also confuses control over administrative information, such as names and addresses associated with registrations, with primary content, such as a Web page.

Professor Mueller defends his writing about the IFWP as merely reporting the view of others, rather than being his own advocacy. However, his reporting is highly selective and results in his confusing the difference between tension that was *within* the IFWP process, versus *between* IFWP and IANA. His casting the issue as being with IANA is contrary to the formal documentation of IFWP, and contrary to the style and content of its process. IFWP was not designed, nor was it conducted, as a decision-making body.

Professor Mueller confuses the actions and intent of the IAHC with those of IANA (and ISOC). He claims to have extensive substantiation for his assessment of the IAHC. Yet none that is relevant to this confusion appears in his book or his letter. This omission is in spite of the fact that his view is at odds with the formal charter for the IAHC, the group's published report, and the direct record of the group's actions.

The review cites IANA's community-based authority. Professor Mueller confuses this with a rejection of the importance of funding, which it was not. He further confuses the IETF technical standards specification process with the operations administrative work of IANA. He continues to misunderstand the role of operational expertise in policy planning for critical infrastructure services, and he ignores the particular 15-year history of successful administrative policy activities provided by operations geeks, for DNS and IP addresses.

Lastly, given the minor points that Professor Mueller chose to address in his response, it is curious that he fails to respond to the primary ethics point raised in the review, namely his pattern of erroneous or absent citations that substantially undermine many of the assertions of his book.

—Dave Crocker, Brandenburg InternetWorking
dcrocker@brandenburg.com

Zero Configuration

IPJ,

I recently read Edgar Danielyan's article on Zero Configuration Networking in the December 2002 issue of IPJ. As is always the case, as a journalist Edgar is entitled to hold and express his own opinions, so as I began the article I didn't know whether to expect glowing praise of Zeroconf, or a savage attack. Thankfully I needn't have worried. I found an excellent and well-balanced article.

I have two brief comments to make.

1. Since Edgar wrote his article, the old expired Internet Drafts have been updated. The drafts Edgar worked from discussed names ending in local.arpa. The actual shipping version of Mac OS X 10.2 ("Jaguar") uses names ending in just local. to designate link-local names, (link-local names are locally assigned, unique only within the local link, not required to be globally unique).
2. Edgar expressed the opinion that Zeroconf is only useful on small networks, not large networks.

While Edgar is correct that Zeroconf per se is aimed at solving the "small network" problem, discovering your local peers is useful no matter how big the network. At the recent IETF meeting in San Francisco, there was a large network with full connectivity to the Internet, including IPv6, yet the printers were still advertised using Rendezvous, and for Mac users those printers showed up automatically in the "Printer" popup menu in the print dialogs, with zero configuration.

There is also the issue that Rendezvous (the Apple product) will go beyond just what is required for Zeroconf (the IETF Working Group). Service Discovery, on which Rendezvous is based, doesn't have to be used only with link-local multicast DNS. It can also be used with conventional unicast DNS. For a preview of what the future might hold, you can browse to find an example list of printers at my house. Type: **`nslookup -q=ptr _ipp._tcp.stuartcheshire.org`**

Thanks for publishing a great article.

—Stuart Cheshire, Apple Computer, Inc.
`cheshire@apple.com`

List of Acronyms

DARPA	*Defense Advanced Projects Agency*
DNS	*Domain Name System*
DNSO	*Domain Name Supporting Organization*
gTLD-MoU	*generic Top Level Domain-Memorandum of Understanding*
IAHC	*International Ad Hoc Committee*
IANA	*Internet Assigned Numbers Authority*
ICANN	*Internet Corporation for Assigned Names and Numbers*
IETF	*Internet Engineering Task Force*
IFWP	*International Forum for the White Paper*
ISOC	*Internet Society*
UDRP	*Uniform Domain Name Dispute Resolution Policy*
WIPO	*World Intellectual Property Organization*

Book Review

Troubleshooting Campus Networks

Troubleshooting Campus Networks: Practical Analysis of Cisco and LAN Protocols, by Priscilla Oppenheimer and Joseph Bardwell, Wiley, 2002

It is perhaps rare that a book review would encompass the acknowledgements. A break from tradition here is warranted, though, because both authors reveal up front what every prospective reader should know when faced with a purchase decision: Is this work drawn merely from professional circumstance on the part of the author or does it embody a passion held by the author? Judge for yourself. How often do the words "love," "wonderful," and "protocol analysis" congregate?

Coauthors Priscilla Oppenheimer and Joseph Bardwell consider the spectrum of protocols and technologies likely to be encountered in a campus environment. A campus network, it is said by the authors, is any one that spans buildings (whether or not in an educational setting). Of course, bricks and mortar are functionally transparent to most modern technologies, and thus the definition of campus could easily be narrowed to any collection of departments or perhaps even any collection of LANs. A contrast is simply being made against the larger metropolitan or wide-area arena.

Although this book does include substantial theory and background for context, it is not yet another rehash of how things *ought* to behave in the vacuum of a lab environment (indeed, the authors occasionally express surprise at their own observations). Neither is it a step-by-step troubleshooting checklist for novice network administrators. To generalize the format, a thorough decomposition of the whole into its many parts follows an introductory discussion of the subject protocol or technology. It is next released into the wild and is quietly observed. Some conclusions are then drawn (some by the authors, some by the reader) regarding appropriate and inappropriate behavior. Lastly, possible courses of action in response to poor or abnormal performance or behavior are considered. This, again, is merely a generalization. The authors take great care to keep the discussion interesting and relevant, often doing so by sharing real-world experiences.

Organization

The six pages that comprise chapter 1 seek to set a stage, define a scope, and target an audience. The reviewer would add only that those of us who trade in wide-area networks also stand to gain a great deal from the experience.

If chapter 2 were packaged for individual sale, it would find its way under the Christmas tree of every colleague, customer, and boss this reviewer has ever encountered. Those readers familiar with Ms. Oppenheimer's acclaimed *Top-Down Network Design*[1] may be surprised to find the expression "bottom-up" in any of her work. It is, however, cornerstone not only to the chapter, but also to the remainder of the book.

This seemingly obvious approach to trouble-shooting and analysis could not possibly be emphasized enough according to this reviewer's professional observation.

Chapters 3, 5, and 6 delve into campus datalink layer technologies, protocols, and architectures, including Ethernet, *Spanning-Tree Protocol* (STP), and *Virtual Local-Area Networks* (VLANs). Yawn? The reviewer challenges the reader to finish these three chapters without learning something of considerable value. The Ethernet discussion, for example, breaks from the traditional approach where a cursory review of frame types, cable types, and topologies is deemed sufficient. Where Ethernet came from, where it is going, how it is encoded and presented to the physical layer (and why), and how to interpret frame size distribution using *Remote Monitoring* (RMON) or a protocol analyzer are but a few of the topics considered. Extensive use of protocol analyzer capture files casts new light on STP and VLANs.

Chapter 4 additionally addresses a Layer 2 technology (IEEE 802.11 wireless LANs) but warrants honorable mention. Rare is the *radio frequency* (RF) engineer who possesses a full appreciation for the heretofore all-digital, all-wired campus realm. Perhaps less common would be the network administrator with a capacity to do much other than tune in an FM radio station on a digital set. The authors masterfully string together all the relevant RF concepts, at exactly the right level of detail, to allow for a solid fundamental comprehension of 802.11 networks, technologies, architectures, and deployment. This chapter also would do superbly for anyone with a generic interest in RF units of measurement.

Chapter 7 advances the discussion up to the network layer. Although this may seem common knowledge for readers of a publication such as the IPJ, it is written from the perspective of seasoned protocol analysts. It is worth your time.

Chapter 8 persists at Layer 3 with a thorough discussion of relevant routing protocols. It is again worth noting the emphasis on analysis versus simple textbook theory. It, too, is worthy of your investment.

Chapter 9 rounds out the protocol stack, beginning with an emphasis on Layer 4 protocols *Transmission Control Protocol* (TCP) and *User Datagram Protocol* (UDP). One of the highlights found here is a thorough lesson on TCP window size analysis. Could there perhaps be a little more to this seemingly intuitive concept than you at first thought? The chapter closes following an in-depth consideration of application layer protocols such as the *File Transfer Protocol* (FTP), *Hypertext Transfer Protocol* (HTTP), and the *Domain Name System* (DNS). The fundamental mechanics of these protocols and how they interact with their lower-layer counterparts make for a good page-turner.

Chapters 10, 11, and 12 are dedicated to troubleshooting and analysis of *Internetwork Packet Exchange* (IPX), AppleTalk, and Windows networking, respectively. The latter is arguably the more relevant. The other two are nonetheless interesting and left the reviewer longing for a decent AppleTalk trace file with which to recreate.

Chapter 13, WAN Troubleshooting for LAN Engineers, covers the obvious wide-area technologies and architectures, such as *Integrated Services Digital Network* (ISDN), Frame Relay, and *Synchronous Optical Network* (SONET) in about as much detail as the typical LAN engineer or administrator is likely to tolerate. The subject of WAN analysis warrants a volume or two on its own in any case and thus would have been out of place if explored in much greater detail.

Conclusion

The reading of Troubleshooting Campus Networks is not to be approached as a spectator sport. Although the protocol analyzer screen captures are aplenty, and they suitably complement the lessons, merely thumbing the pages would be an opportunity missed. This reviewer chose a free, open-source protocol analyzer (readily available on the Internet) as a reading companion. Although likely far less capable, particularly in terms of graphing, than the oft-referenced Wildpackets EtherPeek product, it nevertheless affords the reader a Layer 2 through 7 window into a living, breathing network.

It bears mentioning that although "Cisco" appears in the subtitle, vendor neutrality is, on the whole, maintained. The Cisco sanctioned troubleshooting methodology is given brief mention in chapter 2. Coverage of the Cisco proprietary *Interior Gateway Routing Protocol* (IGRP), the *Enhanced IGRP* (EIGRP), and the Cisco Discovery Protocol is included, as is coverage of Cisco's "enhancements" to STP. Lastly, where appropriate, Cisco IOS® "show" and "debug" output is included alongside protocol analyzer screen captures. None of this coverage appears to be included in the spirit of product promotion (bear in mind that this is not a Cisco Press title and that neither author is presently employed by Cisco Systems). Rather, it seems simply to be an acknowledgement that the target audience might very well include candidates for Cisco's professional and expert-level certification programs (and rightly so).

It is probably anticlimactic that the reviewer would offer a strong buy recommendation for those with an interest in the fundamental interworkings of campus protocols and technologies. The authors' enthusiasm for packet capture and analysis is infectious. Mr. Bardwell, in fact, is apparently so infatuated that he is at times moved to poetry. This could well be one for the ages.

—Scott Vermillion, IT Artisans Group
scott@itartisans-group.com

References

[1] *Top-Down Network Design,* Priscilla Oppenheimer, ISBN 1578700698, Cisco Press, 1998.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

June 2003 Volume 6, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

Articles in *The Internet Protocol Journal* broadly fall into three categories. First, we have articles that explain well-established technologies or operational practices. Second, we offer tutorials on new or emerging protocols and systems, not yet deployed but on the horizon. Finally, IPJ brings you insights, lessons learned and opinions on aspects of networking that have not completely lived up to their promises. In this issue, you will find a mixture of all three.

Our first article is an example from the "nuts-and-bolts" category. The *Border Gateway Protocol* (BGP) is one of the core routing protocols that is widely used in the Internet and has been around for a long time. Kris Foster explains how the *BGP Community* attribute can be used in service provider networks.

Efforts to provide cellular telephones with Internet access systems have produced mixed results. Japan has been leading the way in this area with widespread deployment of iMode devices or variants thereof. Having used such a system I must say I am both impressed and somewhat frustrated. It is wonderful to receive e-mail while on a busy Tokyo train, but accessing the Internet on a tiny screen (typically a 2-inch display with a resolution of 120 x 160 pixels) is not particularly rewarding. Not to mention the bandwidth limitations inherent with this technology. Another system, the *Wireless Application Protocol* (WAP) has been implemented in most countries that offer *Global System for Mobile Communications* (GSM) cell phone service. WAP is the subject of our second article. Edgar Danielyan describes the WAP architecture and looks at some of the lessons learned from its deployment.

The push for deployment of *IP Version 6* (IPv6) is taking place on several fronts and we cover some of them in this issue. In the IETF, a recently formed group has been chartered to help design transition strategies from IPv4 to IPv6. We have a short overview of this effort starting on page 20. Additionally, both the U.S. and Japanese governments are promoting the use of IPv6 in various ways. The U.S. Department of Defense has recently adopted IPv6 as one of its official protocols. In Japan the "IPv6 Appli-Contest 2003" is underway in an effort to encourage development of software and applications for IPv6. See "Fragments," page 37–38 for further details.

Of course, not everyone is convinced that IPv6 is such a good idea, and with that in mind we bring you an opinion piece as well as a Letter to the Editor on this topic.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Application of BGP Communities

by Kris Foster, TELUS

The *Border Gateway Protocol* (BGP) is the glue that binds networks and their individual policies together. Several attributes are passed along and possibly modified with each individual prefix, one of which is the *community* attribute. BGP communities are described poorly in most texts. The problem is not in explaining how they fit into the protocol, but in how to apply these to the real world. In this article I describe how they can be applied within a service provider network and between service provider networks. However, communities are not limited to service providers and can be applied creatively in enterprise networks.

The density of interconnection among service providers, and the various business agreements or political policies, means that controlling who can talk to whom over your network can become difficult. At a basic level there are two types of agreements between service providers: transit/customer and peers.

- Customers pay to receive every prefix from a transit provider.
- Customers advertise only the prefixes they own (along with their customers' prefixes) to the transit provider.
- Peers agree to send only their customers' prefixes to each other, and not other peers' prefixes.

Several methods are available to implement these policies. They can include prefix filters, *Autonomous System* (AS) path filters, and communities. With only prefix and AS path filters, service providers must ensure that as a new customer or peer is added, the prefixes and *AS Numbers* (ASNs) associated with the customer (and potentially *their* customers) are added to the filters on all of the BGP edge routers. This can be automated with scripts, possibly in combination with a route registry database. Very small service providers may be able to manage such a scheme, but as they grow and customer churn begins, this can quickly get out of control. The more time network operators spend in router configurations, the greater likelihood of human error. Communities provide an elegant solution for these problems.

The BGP Community Attribute

Within an AS, all BGP-speaking routers run *Internal BGP* (iBGP) in a full mesh to prevent routing loops (route reflectors can be used to relax this rule). This means that every BGP-speaking router passes its prefixes to each of its iBGP neighbors. ASs that are adjacent typically run eBGP on directly connected routers. All BGP routers share their prefixes—that is, the network number, network mask, and BGP attributes with each other—allowing each to run its own best-path selection algorithm. As a prefix is passed between ASs, an attribute called the AS-PATH is updated with the corresponding ASN. The AS-PATH is used to prevent routing loops between eBGP neighbors.

A community is a BGP attribute that may be added to each prefix. Communities are transitive optional attributes[1], meaning BGP implementations do not have to recognize the attribute and at the network operator's discretion carry it through an AS or pass it on to another AS. The community attribute can be thought of as simply a flat, 32-bit value that can be applied to any set of prefixes. It can be read as a 32-bit value or split into two portions, the first 2 bytes representing an ASN and the last 2 bytes as a value with a predetermined meaning. The format of the community attribute is shown in Figure 1.

The values **`0x00000000`** through **`0x0000FFFF`** and **`0xFFFF0000`** through **`0xFFFFFFFF`** are reserved. Most modern router software displays communities as **`ASN:VALUE`**. In this format the communities **`1:0`** through **`65534:65535`** are available for use. The convention is to use the ASN of your own network as the leading 16 bits for your internal communities and communities that you accept from and send to your customers.

Three communities are defined in RFC 1997[2] and are standard within BGP implementations: NO-EXPORT (**`0xFFFFFF01`**), NO-ADVERTISE (**`0xFFFFFF02`**), and NO-ADVERTISE-SUBCONFED (**`0xFFFFFF03`**). Additionally, NO-PEER (**`0xFFFFFF04`**) has been proposed in an Internet Draft[3].

NO-EXPORT is commonly used within an AS to instruct routers not to export a prefix to eBGP neighbors. For instance, subnets of a larger block can be advertised to influence external AS best-path selection, and those not required for this traffic engineering purpose may be tagged NO-EXPORT to prevent them from being leaked to the Internet (and thus contributing to unnecessary global routing table growth). If a neighboring AS accepts this community, it can be used to selectively leak more specifics for traffic engineering but limit their propagation to just one AS.

NO-ADVERTISE instructs a BGP-speaking router not to send the tagged prefix to any other neighbor, including other iBGP routers.

NO-ADVERTISE-SUBCONFED is used to prevent a prefix from being advertised to other members within a *confederation*. A confederation can be thought of as a single AS, broken down into sub-ASs. The use of confederations within service provider networks is rare or nonexistent, so they are not considered here.

Finally, NO-PEER is used in situations where traffic engineering control over a more specific prefix is required, but to constrain its propagation only to transit providers and not peers. That is, the prefix is advertised from AS to AS provided there is a transit/customer relationship, unlike NO-EXPORT, which restricts propagation of the prefix to only the adjacent AS. Because peers of the various upstream providers will not see this prefix, the larger prefix encompassing the more specific one is used for routing, thereby conserving an extra entry for some in the global routing table. At this time the community is not recognized by major vendors and requires manual implementation.

Adding Depth: The Extended Community

The current community attribute is getting an upgrade with a new transitive-optional attribute (Type 16) called the *Extended Community*[4]. Missing from regular communities was any real form of structure. The current Internet Draft defines the Extended Community as an 8-octet value as shown in Figure 1. The first octet specifies the type (and optionally the second value can specify a subtype). This value dictates the structure given to the remaining octets.

The Type field gives the community some immediate flexibility. The first is the use of bit 0 to represent whether the community is registered with the *Internet Assigned Numbers Authority* (IANA) or if it is specified by the *Internet Engineering Task Force* (IETF). The second bit gives the Extended Community a coarse scope, either *Transitive*, meaning it may be passed between ASs, or *Non-Transitive*, meaning it should be carried only within the local AS.

The Internet Draft also specifies numerous types available for use as templates.

The *Route Target Community* is already in popular use within *Multiprotocol Label Switching Virtual Private Networks* (MPLS VPNs). The Route Target Community identifies a set of routers that may receive this prefix. In the MPLS VPN context, this is necessary to limit the resources required to support individual VPN services; only routers that are part of the individual VPN need to hear about the routes within the VPN.

The *Link Bandwidth Community* gives the network operator additional control in influencing the best path selection. As prefixes are learned from eBGP neighbors, the local neighbor applies this community to specify in bytes per second the bandwidth of the link. It is a *Non-Transitive Community,* so its scope is limited to the local AS.

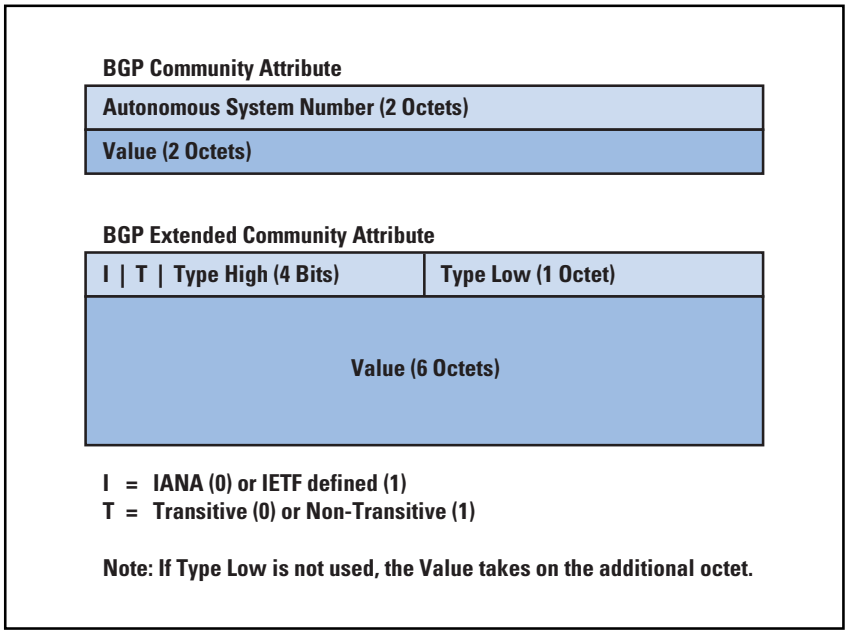


Figure 1: Community Formats

Intra-Autonomous System Communities

Policy control using communities within an AS can go farther than this, and their true value is evidenced when they are used to create new and complex policies. If we take our example of the three basic types of neighbor relationships, customers of a transit provider will want to send their customers' prefixes but not their peers' prefixes. To distinguish between a customer's prefix, a peer's prefix, and a transit provider's prefix, we can add a community to each as we learn it from the neighbor.

When advertising a prefix to a customer, peer, or transit provider, simply match all prefixes carrying the communities associated with the correct policy. As shown in Figure 2, all prefixes received from customers are tagged with **53:100**, peers are tagged with **53:200**, and transit is tagged with **53:300**. Our basic definition of a customer is someone who expects to receive all prefixes, so each customer-facing BGP session is preconfigured to send all prefixes matching **53:100**, **53:200**, and **53:300**. Again, from our definition of a peer being someone who wants to see only our customers, we would preconfigure all of our peers' BGP sessions to send only prefixes tagged with **53:100**.

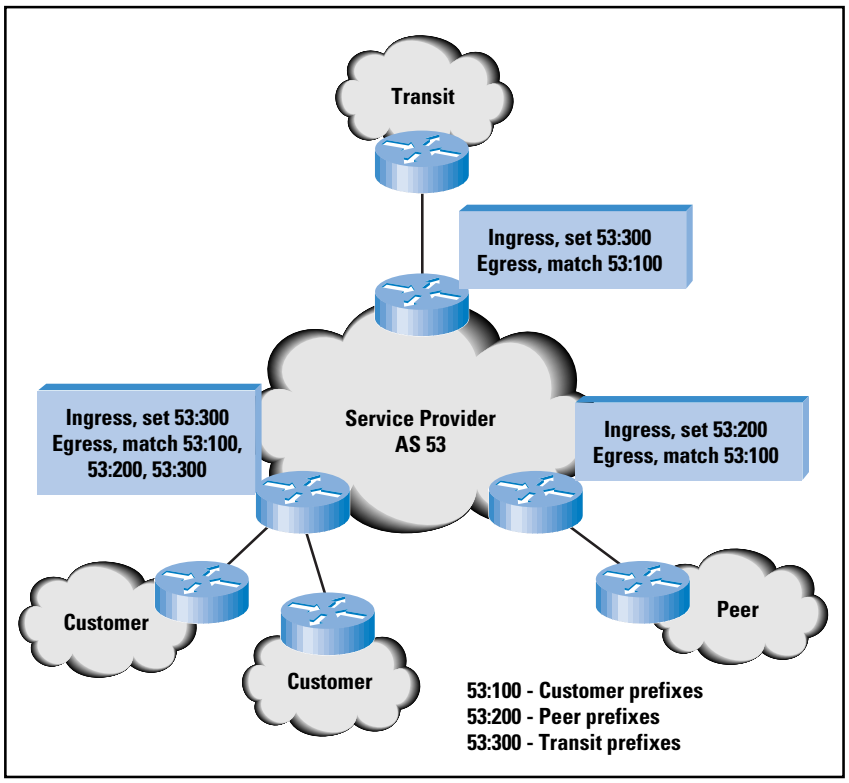


Figure 2: Internal Use of Communities for Applying a Basic Service Provider Policy

We can extend this community coding and turn it into a useful troubleshooting tool by adding more information such as where the route was learned geographically. Codes could be assigned per continent, country, state/province, city, or central office.

During redistribution from an Interior Gateway Protocol, a community can be used to specify the original protocol (for example, *Intermediate System-to-Intermediate System* [IS-IS], *Open Shortest Path First*

[OPSF], or *Routing Information Protocol* [RIP]). These can be used to quickly determine where a prefix came from without tracing it back to the point of its origination.

It is possible to assign these additional properties in two different ways (or a combination). A single community value may represent a single meaning, such as **`53:100`**, meaning a customer-learned prefix. We could then add additional communities such as **`53:1`** to mean a prefix learned on the east coast, **`53:2`** to mean central, and **`53:3`** to mean west coast. Alternatively, a single community could represent both a customer and a prefix learned on the west coast by tagging with the single tag **`53:103`**. To support these complex values, most vendors allow for pattern matching of specific values, ranges of values, and logical operators such as OR and NOT, in the form of regular expressions. Using regular expressions and complex communities can help to make a router configuration more economical and easier to read.

Inter-Autonomous System Communities

We have some options for Inter-AS traffic engineering: we can prepend additional AS numbers onto a prefix path, use *Multi-Exit Discriminators* (if the provider supports this), announce more specific prefixes or not announce prefixes at all, modify the origin type, or use communities designed by the other service provider. Communities are clean and consistent with regard to the method of signaling to an adjacent AS how each prefix should be treated.

Of most concern to downstream customers is controlling their primary and backup circuits. Small service providers and enterprises may negotiate different rates on different circuits. Customers purchasing transit with a commitment to send a high amount of traffic with a lower cost per megabit on one circuit, and on a second circuit purchase transit with a very low commitment but at a higher cost per megabit can save some money, assuming they use only the second circuit during outages on the first. Two simple communities can be used to effectively influence a service provider into using the appropriate primary and backup circuits: one value to lower and another to raise the preference of specific prefixes during the transit provider's best-path selection.

An example of adjusting Local Preference with communities can be found in RFC 1998, "An Application of the BGP Community Attribute in Multi-home Routing"[5].

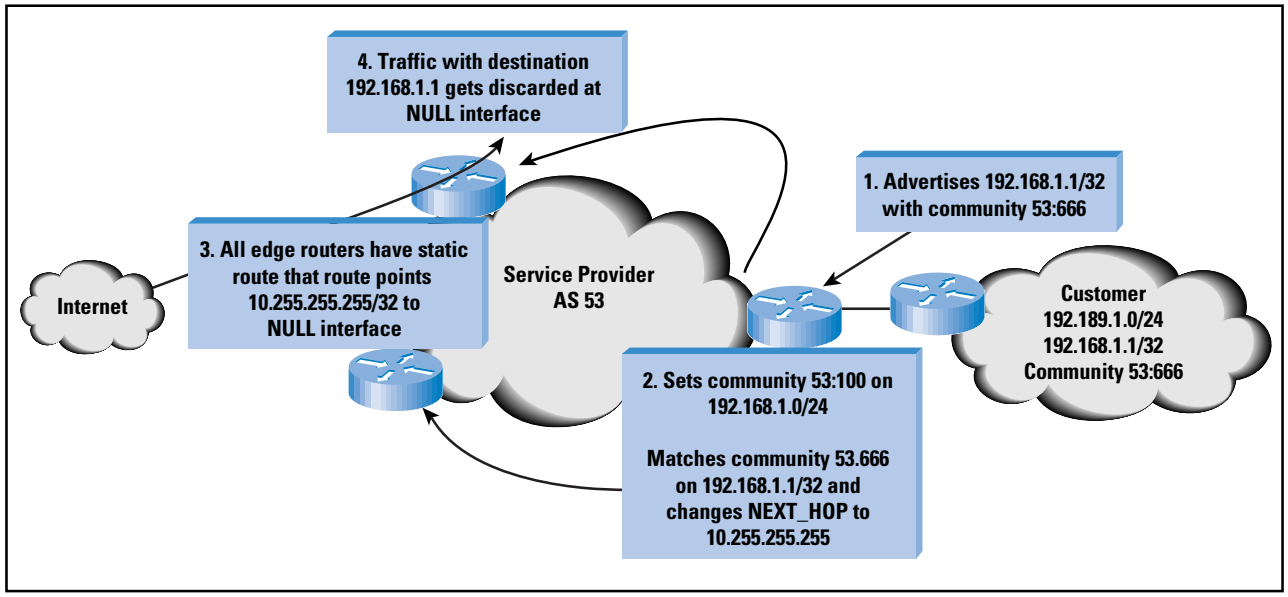
Some other traffic engineering signaling possibilities include:

- Force the adjacent AS to prepend its ASN a certain number of times to a prefix sent to customers or peers.
- Force the other side to selectively advertise a prefix to specific neighbors.
- Request that the neighbor drop all traffic to a prefix.

The last example may seem a little strange; if you are paying someone to deliver traffic, you expect to receive that traffic. Here is where communities can play a role in network security. *Denial-of-Service* (DoS) attacks may take out an entire customer's service, but the attack may be

focused on one or several hosts and not an entire network, as illustrated in Figure 3, allowing customers to tag individual host routes (a subnet consisting of a single address), the customer can signal to the provider to drop all traffic (black hole) for that specific address. To achieve this, the provider selects a single IP address and routes all traffic destined for it to the NULL interfaces on every BGP-speaking router. When a customer signals for a prefix to be blackholed, the service provider replaces the NEXT_HOP information in the BGP advertisement (which under normal circumstances is the edge router IP address) with the specific address that all other routers have statically routed to the NULL interface. When a packet arrives destined for the host under attack, the edge router performs a routing table lookup to find the BGP prefix; using the NEXT_HOP, it then performs a recursive lookup and ultimately sends the packet out the NULL interface. It is important to use other techniques such as prefix lists to prevent a third party from exploiting this technique to disrupt service for others in the Internet.

Figure 3: Customer-Initiated Black Hole to Defend Against a DoS Attack



A service provider may elect to send communities to its customers, leaving it up to the customers to decide for themselves which communities to act on. For a customer who is dual-homed to the same service provider in multiple states or countries, it may be helpful to know where a prefix was originated. A customer could use this community to prefer a connection in New York instead of a Los Angeles connection for European traffic. A single composite metric composed of all relevant geographical information is best, because this gives customers maximum flexibility in choosing the values that are meaningful to them.

Tagging the type of prefix may help other networks to selectively filter more specific addresses. Adding a community specifying if a block is a more specific part of a *Classless Inter-Domain Routing* (CIDR) block being advertised, the CIDR block itself, or if it is a more specific block but the CIDR block is not being advertised, can help the downstream network avoid incorrect filtering.

Example: Network A announces

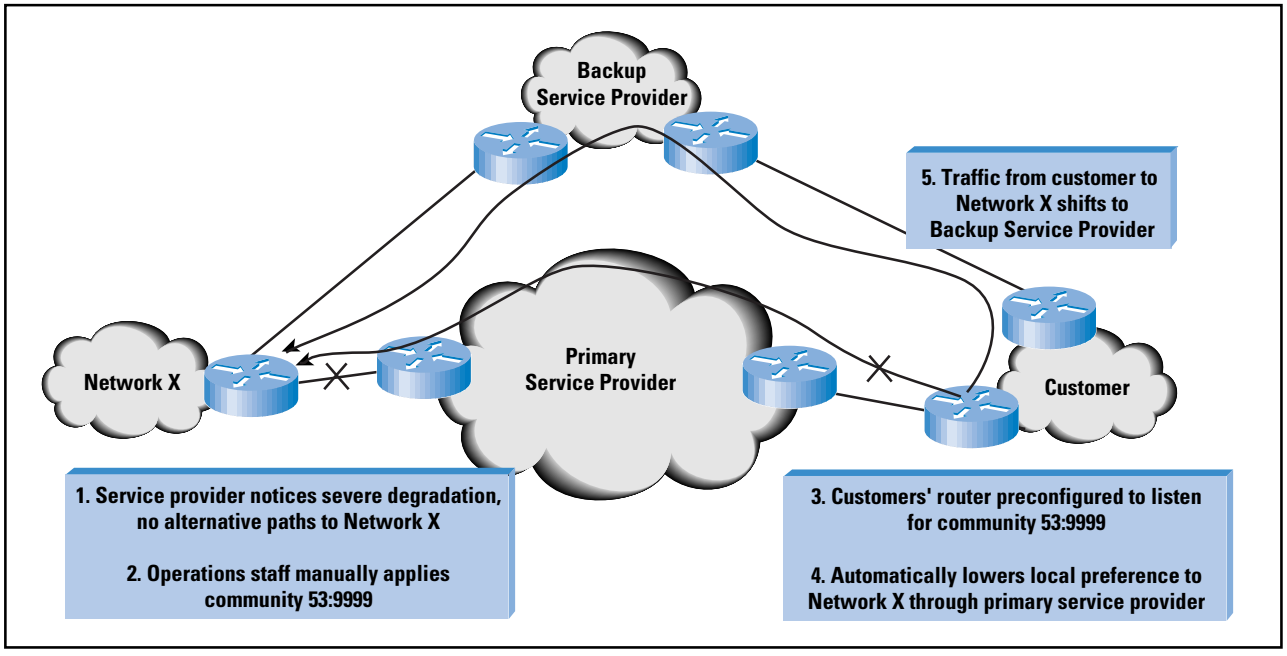
142.77.0.0/16 with a tag of **1:77**
142.77.1.0/24 with a tag of **1:88**
150.3.12.0/24 with a tag of **1:99**

1:77 means it is a CIDR block
1:88 means it is a more specific block within a CIDR block
1:99 means that the full CIDR block is not being announced

Network B then has the option of accepting the more specific **142.77.1.0/24**. It also knows that it must accept **150.3.12.0/24** because there is no other route to this network.

In extreme cases providers may find that a portion of their network has become severely degraded. Planned with customers in advance, the upstream provider manually sets a specific community on prefixes associated with the degradation to indicate that this path should be avoided. This could be helpful during natural disasters, fiber cuts, or other unanticipated network outages/degradation. The downstream customers' inbound filters would then match this community and lower the preference on the prefixes tagged with it, causing them to automatically shift traffic to an alternative source if it is available. The degradation signalling process can be seen in Figure 4.

Figure 4: Provided Initiated Signalling of Severe Route Degradation



Design Recommendations

The following are some suggestions if you are just starting out with using communities in your own network. Even the smallest network can benefit from starting early with a clean community design.

- Choose a set of internal communities that best reflects the topology and characteristics of your network. For external communities some service providers offer none, others offer only enough to allow for the tagging of primary and backup circuits, and others provide a seemingly endless list.
- Keep the set simple. Adding additional complexity typically requires changes to all the BGP-speaking edge routers. Router configurations can quickly grow to enormous proportions to accommodate the numerous community combinations. Troubleshooting a routing mess with a complex community structure can be difficult for those on the graveyard shift.
- Avoid transiting communities received from neighboring ASs blindly through your network. This could be abused intentionally or unintentionally to influence traffic to use your costly transit over settlement-free peering and revenue-generating customer circuits. Problems can be created farther out in the Internet and can be very difficult to locate. Depending on the support of your router software, you may be able to selectively add and remove communities, or failing that, you may need to remove all communities and re-add what is acceptable.
- Document your communities internally and externally. Your customers will appreciate the additional control, and your operations team will have an easier time troubleshooting.

Summary

Communities add power to BGP, changing it from a routing protocol to a tool for signaling and policy enforcement. If deployed correctly and consistently, communities can help make a network scale, easier to operate, easier to troubleshoot, and can give its customers what they want.

References

[1] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, March 1995.

[2] R. Chandra, P. Traina, and T. Li, "BGP Communities Attribute," RFC 1997, August 1996.

[3] G. Huston, "NOPEER Community for BGP Route Scope Control," Internet Draft, May 2003.

[4] S. Sangli, D. Tappan, and Y. Rekhter, "BGP Extended Communities Attribute," Internet Draft, May 2002.

[5] E. Chen and T. Bates, "An Application of the BGP Community Attribute in Multi-home Routing," RFC 1998, August 1996.

KRIS FOSTER, CCIE® #7749, currently lives in Calgary, Alberta, and spends his time in TELUS' IP backbone. His industry affiliations include the Association for Computing Machinery (ACM), the Internet Society (ISOC), and the North American Network Operators Group (NANOG). He can be reached at **kris.foster@telus.com**

WAP: Broken Promises or Wrong Expectations?

by Edgar Danielyan, Danielyan Consulting LLP

The *Wireless Application Protocol* (WAP) was once hailed as the ultimate mobile Internet solution that would revolutionize how we use the Internet and mobile phones. As you may already know, it didn't. What is to blame? Is it bad technology, wrong time, or greedy network operators? Actually, is there a reason to blame anyone? This article introduces WAP with its related technologies and tries to answer these questions. Although WAP is available on a variety of wireless mobile networks, such as those employing *Code Division Multiple Access* (CDMA) IS-95, *Time Division Multiple Access* (TDMA) IS-136, *International Mobile Telecommunications* (IMT-2000), *Universal Mobile Telecommunication System* (UMTS), and *Wideband Code Division Multiple Access* (W-CDMA), in addition to GSM/GPRS this article covers WAP over GSM/GPRS networks only.

A Case for WAP

Before looking at WAP itself, let's first recall what sparked its idea and development. As we all know, most if not all *second-generation* (2G) mobile phones and networks suffer from numerous limitations that make it impossible or impractical to use standard Internet protocols and technologies on today's mobile phones. The most visible of these limitations include the following:

- Low bandwidth (usually 9.6 kbps)
- High network latency
- Small, mostly monochrome displays
- Numeric keypads
- Slow processors
- Limited memory

All these limitations meant that it was necessary to develop an alternative suite of protocols and technologies that would work on these mobiles phones but still provide functionality comparable to the standard Internet technologies used on wired networks and desktops. WAP was developed to address these issues[1].

WAP Forum and Open Mobile Alliance

The *WAP Forum* is the industry organization behind WAP and its associated protocols and technologies. In 2002, the WAP Forum and the Open Mobile Architecture Initiative merged, creating the *Open Mobile Alliance* (OMA), which will continue work on WAP 2 and develop new mobile and wireless solutions. Nearly 200 of the world's top network operators, vendors, and content providers are members of the Open Mobile Alliance[2]. Other organizations such as the *Location Interoperability Forum* (LIF)[3], *Multimedia Messaging (MMS) Interoperability Group* (MMS-IOP)[4], *SyncML Initiative*[5], and *Wireless Village Initiative*[6] have announced their support for the new organization.

Global System for Mobile Communications

GSM, or *Global System for Mobile Communications,* is used by more than 700 million people across 190 countries[7]. In less than ten years after its introduction, GSM became the most popular and widely used digital mobile wireless communications standard in the world. GSM networks use TDMA technology and are fully digital, employing a unique voice codec known as *GSM codec* to provide relatively good voice quality using narrow bandwidth (usually 9.6 kbps). However, GSM is not as secure as many may think. Although it does use encryption and smartcard technology, this didn't result in strong security. As a result, it is possible to intercept and decrypt GSM communications, fake short text messages (*Short Message Service* [SMS]), and clone *Subscriber Identification Modules* (SIMs), miniature smartcards used to identify subscribers to the GSM network. GSM security is not the subject of this article, but it deserves attention and I hope to cover it in a separate article in this journal.

Wireless Application Environment

Before proceeding further, we should clarify one point. The term "WAP" is usually used to refer to the entire suite of protocols and technologies that are actually called the *Wireless Application Environment* (WAE)[8]. However, "WAP" is used everywhere to refer to WAE (which includes WAP). Because WAP is the commonly used term, we shall continue to use it as well.

Wireless Application Protocol

WAP protocols were expected to satisfy the following criteria in order to implement the objectives set by the WAP Forum:

- Independent of wireless network standard (bearer technology)
- Open to all
- Will be proposed to the appropriate standards bodies
- Applications scale across transport options
- Applications scale across device types
- Extensible to new networks and transports

The objectives of the WAP as defined by the WAP Forum follow:

- To bring Internet content and advanced data services to digital cellular phones and other wireless terminals
- To create a global wireless protocol specification that will work across differing wireless network technologies
- To enable the creation of content and applications that scale across a very wide range of bearer networks and device types
- To embrace and extend existing standards and technology wherever appropriate

Two major versions of WAP exist—Versions 1 and 2. WAP Version 2 is backward compatible with WAP Version 1 and tends to be more integrated with the newest Internet and Web standards than WAP 1. Although WAP uses many technologies and concepts from the Internet and Web worlds, because of their inherent limitations, WAP devices are unable to directly access Web resources on the Internet[9]. To do so, they must use a WAP gateway. The following table shows the relationship between the WAP client device, WAP gateway, and Web servers on the Internet, with their protocol layers side by side:

Web Client	WAP Gateway	Web Server
WSP	WSP/HTTP	HTTP
WTP	WTP/HTTP	HTTP
WTLS	WTLS/SSL/TLS	SSL/TLS
WDP	WDP/TCP/UDP	TCP/UDP
Bearer	Bearer/IP	IP

The table shows that the main function of the WAP gateway is to translate between WAP and Web/Internet protocols, conventions, and encodings. In some cases the WAP gateway and the Web server may be the same system, eliminating the need for a separate WAP gateway and possibly improving performance—however, for this setup to work the combined WAP/Web server has to be integrated into the mobile/wireless network provider's infrastructure. In practice, network operators provide the WAP gateway services and content providers offer WAP content on separate Web servers configured for WAP access (any standards-compliant Web server can do this).

Wireless Session Protocol

The *Wireless Session Protocol* (WSP) is the WAP session-layer protocol for remote operations between a wireless (WAP) client and proxies, gateways, and servers[10]. It functions above the *Wireless Transaction Protocol* (WTP) and the *Wireless Datagram Protocol* (WDP), and optionally, the *Wireless Transport Layer Security* (WTLS). The WSP provides a way for an organized exchange of data between client/server applications in a wireless environment. It provides such features as establishment and release of sessions between client and server; agreement on common functionality by way of negotiation; and exchange of data between client and server using compact encoding. WSP defines two subprotocols—a connection-oriented session service protocol over WTP and a connectionless service protocol over the WDP.

Wireless Transaction Protocol

WTP runs on top of the WDP and optionally, the WTLS protocol, and provides the request/response protocol used by WAP browsers to request and receive content[11]. WTP is a reliable transaction-oriented protocol specially designed for wireless networks—in WTP there are no connection setup or release phases.

Reliability in WTP is achieved using transaction IDs, retransmissions, acknowledgments, and removal of duplicates.

Wireless Datagram Protocol

WDP is the transport protocol of WAP[12]. It operates directly above the bearer technology (such as GSM CSD or GPRS) and directly below WTP described previously. WDP provides a consistent, bearer-independent interface for the upper-level protocols to the transport service provided by WDP. In addition to the GSM *Circuit Switched Data* (CSD) and the *General Packet Radio Service* (GPRS), WDP supports the following wireless bearer technologies:

GSM SMS	IDEN Packet Data
GSM USSD	FLEX
GSM Cell Broadcast	REFLEX
ANSI-I36	PHS CSD
CDPD	DataTAC
CDMA CSD	TETRA Short Data Service
CDMA Packet Data	TETRA Packed Data
CDMA SMS	DECT SMS
PDC Circut Switched Data	DECT Connection-oriented Service
PDC CSD	DECT Packed Switched Service
PDC Pacet Data	Mobitex
IDEN CSD	

When used over GSM CSD, WDP actually uses the *User Datagram Protocol* (UDP) in the following way:

Layer 4: UDP
Layer 3: Internet Protocol (IP)
Layer 2: Point-to-Point Protocol (PPP)
Layer 1: GSM CSD

When used over the GPRS, PPP at Layer 2 is not necessary, because GPRS works at Layers 1 and 2:

Layer 4: UDP
Layer 3: IP
Layers 1 and 2: GSM and GPRS

In all cases when IP is supported over a given bearer, UDP is used by WDP—actually, UDP is the WDP in these cases.

Wireless Control Message Protocol

Not surprisingly, *Wireless Control Message Protocol* (WCMP) resembles and corresponds to the *Internet Control Message Protocol* (ICMP) of TCP/IP networks[13]. WCMP is used by WDP nodes to report errors and provide network information and diagnostics. However, WCMP is not necessary and is not used with bearers that support IP—the function of WCMP in these circumstances is carried out by ICMP. In particular, this is the case with GSM CSD and GPRS bearers.

Wireless Transport Layer Security

WTLS is the transport layer security protocol of the WAE that provides privacy, integrity, and authentication services[14]. It is heavily influenced by the *Transport Level Security* (TLS) protocol Version 1 and includes additional support for optimized handshake, connectionless transport, and dynamic key refresh. WTLS, like other WAP protocols, is optimized for low-bandwidth, high-latency wireless networks and supports server and client certificates for mutual authentication. WTLS includes the following three subprotocols:

- Cipher protocol
- Alert protocol
- Handshake protocol

The following cryptographic algorithms are used by the Wireless TLS protocol:

- RSA
- SHA-1
- Diffie-Hellman (DH)
- Elliptic Curve Diffie-Hellman (EC-DH)
- DSA
- Elliptic Curve DSA (EC-DSA)
- MD5
- RC5
- DES
- IDEA

WTLS is tightly linked to and works in conjunction with the *Wireless Public Key Infrastructure* (WPKI).

Wireless Public Key Infrastructure

WPKI tries to reuse the existing *Public Key Infrastructure* (PKI) standards as much as practical to provide an adequate PKI framework for the WAE. Both X.509 and WTLS certificates can be used by WTLS[15].

Wireless Markup Language Version 1

The *Wireless Markup Language* (WML) Version 1[16] is used in WAP/WAE 1 and supported in WAE 2. Unlike usual HTML, it is a strict application of the *Extensible Markup Language* (XML), specially designed for use on narrowband devices. Also unlike HTML, WML has a metaphor of *decks* and *cards*. A deck contains one or more cards, and cards in turn contain one or more screens of user interaction. This metaphor helps increase efficiency on low-speed, high-latency wireless networks by bundling several screens into a single WML file (deck). WML supports all basic text display options, such as *italic*, **boldface**, and underlined text, as well as inter-card and inter-deck navigation using hyperlinks. The most apparent difference between HTML and WML noted by HTML developers is the fact that WML is a strict markup language and does not tolerate even seemingly little errors—an incorrectly written WML file will not display at all. Some would say this is an overkill but it is not—this feature of WML is important because compiled versions of WML files are sent to WAP clients by the WAP gateway instead of the source WML text files. This compiled bytecode is known as *WMLC*, and it considerably lessens the time it takes to download a WML document.

WML Version 2

WML version 2 is based on XHTML Basic with additional modules for support of features specific to wireless devices—this extended XHTML is called *XHTML Mobile Profile* (XHTML-MP)[17]. WML Version 2 is backward compatible with WML Version 1, so devices able to display WML 2 will also display WML 1 content. Use of XHTML shows that WAP in Version 2 is moving toward even closer integration with Internet and Web standards.

WMLScript

WMLScript is a lightweight scripting language based on ECMAScript, which is in turn based on JavaScript[18]. It is well integrated with WML and has a defined set of standard libraries, including support for cryptographic functions. Like WML, WMLScript files are also compiled into bytecode and only then sent to the requesting WAP device. Another difference between JavaScript and WMLScript is that WMLScript content is not embedded in WML pages but instead is requested separately—the necessary WMLScript functions are only referenced in WML pages. The main use of WMLScript is the client-side validation of user input—accepting only valid input is more crucial for WAP than for Web applications because of the low-speed and usually expensive nature of WAP transport.

Wireless bitmaps

The *Wireless Bitmaps* (WBMP) file format (`.wbmp`) is used by WAP devices to transmit and display small and simple monochrome bitmap images[19].

GSM CSD

CSD is the traditional data service provided by GSM networks. Also known as a *data call service*, it provides either a 9.6- or 14.4-kbps dialup facility and is supported by all GSM networks. Data calls are possible both from and to a GSM network. When used as a bearer for WAP, it serves at the physical layer of the *Open System Interconnection* (OSI) model, with PPP used in the usual way.

High-Speed Circuit Switched Data

The *High-Speed Circuit Switched Data* (HSCSD) service is similar in nature to CSD, but provides 28.8 or 43.2 kbps of bandwidth. It is not as widespread as the regular CSD, nor it is as asked-for as GPRS.

General Packet Radio Service

GPRS is an always-on, higher-speed alternative to the CSD service of GSM networks. It solves two of the most annoying issues of GSM data users—connection delay (the time it takes to set up a data call before data may be sent or received) and the bandwidth limitation, increasing the supported data rates to 48 kbps, with theoretical maximum of 171.2 kbps. Because GPRS is a connectionless packet service, GPRS terminals are always connected and may send and receive IP packets at any time. This makes possible applications such as instant messaging previously impossible or impractical with GSM CSD. Eight time slots are available for GPRS in GSM networks, but only five may be used simultaneously. The GPRS class supported by the GPRS terminal dictates what data rates are possible:

Class 2:	Uplink 8–12 kbps, downlink 16–24 kbps
Class 4:	Uplink 8–12 kbps, downlink 24–36 kbps
Class 6:	Uplink 16–24 kbps, downlink 24–36 kbps, or Uplink 24–36 kbps, downlink 16–24 kbps
Class 8:	Uplink 8–12 kbps, downlink 32–40 kbps
Class 10:	Uplink 8–12 kbps, downlink 32–48 kbps, or Uplink 16–24 kbps, downlink 24–36 kbps
Class 12:	Uplink 8–12 kbps, downlink 32-48 kbps, or Uplink 16–24 kbps, downlink 24–36 kbps, or Uplink 24–36 kbps, downlink 16–24 kbps, or Uplink 32–48 kbps, downlink 8–12 kbps

In addition to the classes of GPRS service, there are three classes of GPRS terminals:

- Class A terminals can be connected to GSM and GPRS services simultaneously.
- Class B terminals can be connected to both GSM and GPRS services, but can use only one service at a time.
- Class C terminals can be connected to either GSM or GPRS services but the user has to switch between two modes of operation.

When used as a bearer for WAP, GPRS works at the physical and data link layers of the OSI reference model. Because GPRS is connectionless and always on, there is no need for PPP—so IP works directly over GPRS.

So Why Aren't We Happy with WAP?

Many surveys of customer opinion show that the end users of WAP are not as happy as WAP developers and content providers wanted them to be. WAP service and content providers discovered that sign-up and usage rates of WAP services have not reached two-thirds of the total customer base once predicted. In short, WAP didn't change the world, and people still use their mobile phones mainly to talk to each other and send a text message or two. If you have used WAP, you probably know the reasons: the data transfer rate is slow, screens are small, charges are high, and it is tiring to type even a short URL or an e-mail message using the ten keys of a phone.

But wait a moment—are these limitations of WAP or the handsets and networks they use? Remember, WAP was required to work on devices with many limitations? So it does. Is WAP to blame that these devices have these limitations? No, that wouldn't be just. But of course it is not only the today's technology restrictions that stood in the way of the widespread usage and popularity of WAP. Scarcity of WAP content and services also contributed to this. Relatively high charges for WAP/data usage by network operators didn't help either, so the combination of these issues resulted in the situation we have today—most networks support WAP but most users don't use it anyway.

Is the technology dead, as some think? Definitely not—there are millions of WAP handsets and most wireless users will not have 3G for the foreseeable future because of both technical and economic issues, so the only available solution for these users is WAP. On the other side, 3G networks and handsets are coming and will be upon us sooner or later (they are already available in some countries), and only time will show whether tomorrow's WAP will be more popular or less relevant when 3G finally arrives. And, of course, fundamental limits of mobile phones—screen sizes, power consumption, and input methods—will still remain relevant. Other issues, such as the time it takes to set up a CSD connection, are solved by newer technologies such as GPRS, and are not really faults of WAP. You may say that if GRPS is available why would you need WAP? Why not run trusted IP? Well, this is true if you are using GPRS with a laptop or a palmtop computer, but a large majority of mobile phones don't have the resources necessary to run IP, UDP, TCP, HTTP/HTTPS, POP, and SMTP—so even if GPRS is available but your equipment cannot run the full TCP/IP suite, your only choice is still WAP.

Although WAP is clearly not as popular as its proponents and developers hoped, it is still used and developed, and handsets that support only WAP are still sold. But the hype and excitement built up by the media and the industry didn't match the reality, and it is these unrealistic expectations that have broken the promise of WAP.

Additional Acronyms

DataTAC:	*Motorola wireless data system*
DECT:	*Digital Enhanced Cordless Technology*
DES:	*Data Encryption Standard*
DSA:	*Digital Signature Algorithm*
FLEX:	*Motorola one-way paging system*
IDEA:	*International Data Encryption Algorithm*
IDEN:	*Integrated Dispatch Enhanced Network*
MD5:	*Message Digest 5*
PDC:	*Pacific Digital Cellular System*
RC5:	*Rivest Cipher 5*
REFLEX:	*Motorola two-way paging system*
SHA-1:	*Secure Hash Algorithm 1*
TETRA:	*TErrestrial Trunked RAdio* Nokia open digital professional mobile radio standard
USSD:	*Unstructured Supplementary Service Data*

For Further Reading

[1] WAP Forum: **http://www.wapforum.org**

[2] Open Mobile Alliance: **http://www.openmobilealliance.org**

[3] Location Interoperability Forum:
http://www.openmobilealliance.org/lif

[4] MMS Interoperability Group (MMS-IOP):
http://www.openmobilealliance.org

[5] SyncML: **http://www.openmobilealliance.org/syncml**

[6] Wireless Village: **http://wireless-village.org**

[7] Global System for Mobile Communications (GSM):
http://www.etsi.org, http://www.gsmworld.com

[8] Wireless Application Environment (WAE) Version 2.0:
http://www.wapforum.org

[9] Wireless Application Protocol Architecture Specification:
http://www.wapforum.org

[10] Wireless Session Protocol Specification: **http://www.wapforum.org**

[11] Wireless Transaction Protocol Specification:
http://www.wapforum.org

[12] Wireless Datagram Protocol Specification:
http://www.wapforum.org

[13] Wireless Control Message Protocol Specification:
http://www.wapforum.org

[14] Wireless Transport Layer Security Specification:
http://www.wapforum.org

[15] Wireless Public Key Infrastructure Architecture Specification:
http://www.wapforum.org

[16] Wireless Markup Language Version 1 Specification:
http://www.wapforum.org

[17] Wireless Markup Language Version 2 Specification:
http://www.wapforum.org

[18] WMLScript Specification: **http://www.wapforum.org**

[19] Wireless Bitmap Specification: **http://www.wapforum.org**

EDGAR DANIELYAN, CISSP, CCNP Security, CCDP®, SCNA, TICSA, CIWCI Security is the principal partner at Danielyan Consulting LLP (**www.danielyan.com**), an information security consultancy in London and Yerevan. He is a published author and editor specialising in UNIX, networking, and information security, having been a cofounder of a national ISP and manager of a country TLD. His book, *Solaris 8 Security*, was published by New Riders Publishing in English and by Pearson Education in Japanese. He is a member of IEEE, IEEE Standards Association, IEEE Computer Society, ACM, ISACA, USENIX, and the SAGE. E-mail: **edd@danielyan.com**

The IETF IPv6 Operations Group and the Development of a Framework for Deployment of IPv6 into IPv4 Networks

by Bob Fink,
Margaret Wasserman, Wind River,
Jun-ichiro Itojun Hagino, IIJ

During 2002, the *Internet Engineering Task Force* (IETF) determined that it was best to focus the introduction of IPv6 into the IPv4 Internet by developing deployment scenarios before further development of transition mechanisms without any clearly identified framework for their place in an IPv6 deployment.

Previously the IPv6 Transition working group of the IETF, called *ngtrans* (for IP next-generation transition), was chartered to develop mechanisms and tools to support an IPv6 transition. This work initially focused, in 1995–1996, on the development of the original IPv6 standards, and it led to the basic Transition Mechanism RFC 1933[1] and later RFC 2893[2] that defined dual IPv4 and IPv6 protocol stack operation as well as IPv6-over-IPv4 tunnels.

Subsequent attempts to define a framework for transition in 1998–1999 were not successful because there did not appear to be a single vision for a transition to IPv6. Indeed the focus became one of how to have IPv4 and IPv6 coexist for a long period of time, because most felt that a full transition could take well over 10–15 years, with many believing that it would never completely obsolete IPv4. This led to the development of many transition mechanisms and tools, some of which might possibly be more useful than others, that never fit into a coherent framework for operation of a *dual protocol,* that is, IPv4 and IPv6, network.

v6ops

Thus in 2002 the ngtrans working group was disbanded, and the IPv6 Operations working group, *v6ops,* created. The v6ops working group was chartered to:

- Solicit input from network operators and users to identify operational or security issues with the IPv4/IPv6 Internet, and determine solutions or workarounds to those issues. This includes identifying standards work that is needed in other IETF working groups or areas and working with those groups or areas to begin appropriate work. These issues will be documented in Informational or *Best Current Practice* (BCP) RFCs, or in Internet-Drafts. For example, important pieces of the Internet infrastructure such as the *Domain Name System* (DNS), the *Simple Mail Transfer Protocol* (SMTP), and the *Session Initiation Protocol* (SIP) have specific operational issues when they operate in a shared IPv4/IPv6 network. The v6ops working group will cooperate with the relevant areas and working groups to document those issues, and find protocol or operational solutions to those problems.

- Provide feedback to the IPv6 working group regarding portions of the IPv6 specifications that cause, or are likely to cause, operational or security concerns, and work with the IPv6 working group to resolve those concerns. This feedback will be published in Internet-Drafts or RFCs.
- Publish Informational RFCs that help application developers (within and outside the IETF) understand how to develop IP version-independent applications. Work with the Applications area, and other areas, to ensure that these documents answer the real-world concerns of application developers. This includes helping to identify IPv4 dependencies in existing IETF application protocols and working with other areas or groups within the IETF to resolve them.
- Publish informational or BCP RFCs that identify potential security risks in the operation of shared IPv4/IPv6 networks, and document operational practices to eliminate or mitigate those risks. This work will be done in cooperation with the Security area and other relevant areas or working groups.
- Publish Informational or BCP RFCs that identify and analyze solutions for deploying IPv6 within common network environments, such as *Internet Service Provider* (ISP) networks (including core, *Hybrid Fiber-Coaxial* [HFC] or cable, DSL, and dialup networks), enterprise networks, unmanaged networks (home or small office), and cellular networks. These documents should serve as useful guides to network operators and users on how to deploy IPv6 within their existing IPv4 networks, as well as in new network installations.
- Identify open operational or security issues with the deployment scenarios documented in the previous bullet point and fully document those open issues in Internet-Drafts or informational RFCs. Try to find workarounds or solutions to basic, IP-level operational or security issues that can be solved using widely applicable transition mechanisms, such as dual-stack, tunneling, or translation. If the satisfactory resolution of an operational or security issue requires the standardization of a new, widely applicable transition mechanism that does not properly fit into any other IETF working group or area, the v6ops working group will standardize a transition mechanism to meet that need.
- Assume responsibility for advancing the basic IPv6 transition mechanism RFCs along the standards track, if their applicability to common deployment scenarios is demonstrated.

v6ops has started by creating four efforts to define transition scenarios and subsequently to analyze them for potential solutions to the deployment scenarios. These four efforts follow:

- *Third Generation Partnership Project* (3GPP) defined packet networks, that is, *General Packet Radio Service* (GPRS) that would need IP Version 6 deployment into the IPv4 Internet.

- "Unmanaged networks," which typically correspond to home networks or small office networks.
- ISP networks, including core, HFC or coaxial, DSL, dialup, public wireless, broadband Ethernet, and Internet exchange points.
- Enterprise networks, which are networks that have multiple links and a router connection to an ISP, and are actively managed by a network operations entity.

During 2003 and 2004 it is expected that these deployment scenario efforts will lead to further analysis and identification of deployment solutions and development of appropriate mechanisms to support them.

In addition to this work, serious efforts are under way to engage the entire IETF standards process in the identification and development of appropriate solutions for an IPv6 deployment. One such effort is the *IPv4 Survey* project, which has reviewed the entire IETF RFC catalog of standards to identify what work might need to be done and to disseminate this information to the appropriate area within the IETF.

As progress is made in v6ops, follow-up articles in IPJ will inform you of these efforts.

For Further Reading

[1] "Transition Mechanisms for IPv6 Hosts and Routers," R. Gilligan and E. Nordmark, RFC 1933, April 1996.

[2] "Transition Mechanisms for IPv6 Hosts and Routers," R. Gilligan and E. Nordmark, RFC 2893, August 2000.

[3] v6ops IETF information:
http://www.ietf.org/html.charters/v6ops-charter.html

[4] v6ops Web site:
http://www.6bone.net/v6ops/http://www.6bone.net/v6ops/

ROBERT FINK is a retired U.S. national laboratory network researcher working with the IPv6 Forum. He is currently a co-chair of the IETF v6ops (IPv6 Operations) working group, and leads the 6bone project. You can reach him at: **bob@thefinks.com**

MARGARET WASSERMAN is a Principal Technologist at Wind River. She is currently a co-chair of the IETF IPv6 and v6ops working groups. You can reach her at: **mrw@windriver.com**

JUN-ICHIRO ITOJUN HAGINO is a network researcher with IIJ Research Laboratory. He is currently a co-chair of the IETF v6ops working group and a member of the IETF IAB. You can reach him at **itojun@iijlab.net**

Opinion: The Mythology of IP Version 6

by Geoff Huston, Telstra

Disclaimer: This is an opinion piece and, therefore, the author takes some liberties in making his points. I hope you as the reader take this in the spirit in which it is intended—a gentle poke at ourselves that sometimes we oversell ourselves and our technology.

In January 1983, the *Advanced Research Projects Agency Network* (ARPANET) experienced a "flag day," and the Network Control Protocol, NCP, was turned off, and TCP/IP was turned on. Although there are, no doubt, some who would like to see a similar flag day where the world turns off its use of IPv4 and switches over to IPv6, such a scenario is a wild-eyed fantasy. Obviously, the Internet is now way too big for coordinated flag days. The transition of IPv6 into a mainstream deployed technology for the global Internet will take some years, and for many there is still a lingering doubt that will happen at all.

Let's look more closely at how IPv6 came about, and then look at IPv6 itself in some detail to try to separate the myth from the underlying reality about the timeline for the deployment of IPv6. Maybe then we can suggest some answers to these questions.

IPv6

The effort that has led to the specification of IPv6 is by no means a recently started initiative. A workshop hosted by the then *Internet Activities Board* (IAB) in January 1991 identified the two major scaling issues for the Internet: a sharply increasing rate of consumption of address space and a similar, unconstrained growth of the interdomain routing table. The conclusion reached at the time was that "if we assume that the Internet architecture will continue in use indefinitely, then we need additional [address] flexibility."

These issues were considered later that year by the *Internet Engineering Task Force* (IETF) with the establishment of the ROAD (*ROuting and ADdressing*) effort. This effort was intended to examine the issues associated with the scaling of IP routing and addressing, looking at the rate of consumption of addresses and the rate of growth of the interdomain routing table. The ultimate objective was to propose some measures to mitigate the worst of the effects of these growth trends. Given the exponential consumption rates then at play, the prospect of exhaustion of the IPv4 Class B space within two or three years was a very real one at the time. The major outcome of the IETF ROAD effort was the recommendation to deprecate the implicit network/host boundaries that were associated with the Class A, B, and C address blocks. In their place the IETF proposed the adoption of an address and routing architecture where the network/host boundary was explicitly configured for each network, and proposed that this boundary could be altered such that two or more network address blocks may be aggregated into a common, single block.

Side Note:

Some would argue that although CIDR was important, it was not the only reason why IPv4 has been able to defy the earlier predictions of its imminent demise. Dynamic *Network Address Translation*, or NAT, allows a network to use a local private address pool to uniquely number its devices, and then translate these private addresses into public addresses to support transactions involving local and external end points. This way, a small pool of public addresses, or even a single address, is used to service a very much larger local private network. It is difficult to estimate the number of devices that are positioned behind NATs, but a highly conservative estimate would see the Internet being at least three times as large as the directly visible part of the Internet.

Side Note:

At an IETF plenary session from that time, the OSI protocol suite was termed the "Road-kill of the Information Superhighway." It was not completely clear that the presenter made the comment in jest!

This approach was termed *Classless Interdomain Routing*, or CIDR. This was a short-term measure that was intended to buy some time, and it was acknowledged that it did not address the major issue of defining a longer-term, scalable network architecture. But as a short-term measure it has been amazingly successful, given that almost ten years and one Internet boom later, the CIDR address and routing architecture for IPv4 is still holding out.

The IAB, by then renamed the Internet *Architecture* Board, considered the ROAD progress in June 1992, still with its eye on the longer-term strategy for Internet growth. The board's proposal was that the starting point for the development of the next version of IP would be *Connectionless Network Layer Protocol* (CLNP). This protocol was an element of the *Open System Interconnection* (OSI) protocol suite, with CLNP being defined by the ISO 8473 standard. It used a variable-length address architecture, where network level addresses could be up to 160 bits long. RFC 1347 contained an initial description of how CLNP could be used for this purpose within the IPv4 TCP/IP architecture and with the existing Internet applications. For the IAB this was a bold step, and considering that the IETF community at the time regarded the OSI protocol suite as a very inferior competitor to its own efforts with IP, it could even be termed a highly courageous step. Predictably, one month later in July 1992, at the IETF meeting this IAB proposal was not well received.

The IETF outcome was not just a restatement of architectural direction for IP, but a sweeping redefinition of the respective roles and membership of the various IETF bodies, including that of the IAB.

Of course such a structural change in the composition, roles, and responsibilities of the bodies that collectively make up the IETF could be regarded as upheaval without definite progress. But perhaps this is an unkind view, because the IAB position also pushed the IETF into a strenuous burst of technical activity. The IETF immediately embarked on an effort to undertake a fundamental revision of the Internet Protocol that was intended to result in a protocol that had highly efficient scaling properties in both addressing and routing. There was no shortage of protocols offered to the IETF during 1992 and 1993, including the fancifully named TUBA, as well as PIP, SIPP and NAT.

This effort was part of a process intended to understand the necessary attributes of such a next-generation protocol.

The IETF formed an *Internet Protocol Next Generation (IPng) Directorate* in 1994, and canvassed various industry sectors to understand the broad dimensions of the requirements of such a protocol. This group selected the IPv6 Protocol from a set of proposals, largely basing its selection on the so-called "Simple Internet Protocol," or SIP proposal. The essential characteristic of the protocol was that of an evolutionary refinement of the Version 4 protocol, rather than a revolutionary departure from Version 4 to an entirely different architectural approach.

Side Note:

IPv6 has had a variety of names—the original IAB documents refer to IP Version 7, working on the assumption that the protocol numbers 5 and 6 were already in use in research networks. It was renamed IPng, for "next generation."

The final word from the *Internet Assigned Numbers Authority* (IANA) was that protocol number 6 was unused, and the final specification was named Version 6 of the Internet Protocol.

The major strength of IPv6 is the use of fixed-length, 128-bit address fields. Other packet header changes include the dropping of the fragmentation control fields from the IP header, dropping the header checksum and length, and altering the structure of packet options within the header and adding a flow label. But it is the extended address length that is the critical change with IPv6. A 128-bit address field allows an addressable range of 2 to the 128th power, and 2 to the power of 128 is an exceptionally large number. On the other hand, if we are talking about a world that is currently capable of manufacturing more than a billion silicon chips every year, and recognizing that even a one in one thousand address utilization rate would be a real achievement, then maybe it is not all that large a number after all. There is no doubt that such a protocol has the ability to encompass a network that spans billions of devices, which is a network attribute that is looking more and more necessary in the coming years.

Its not just the larger address fields per se, but also the ability for IPv6 to offer an answer to the address scarcity workarounds being used in IPv4 that is of value here. The side effect of these larger address fields is that there is then no forced need to use NAT as a means of increasing the address scaling factor. NAT has always presented operational issues to both the network and the application. NAT distorts the implicit binding of IP address and IP identity and allows only certain types of application interaction to occur across the NAT boundary. Because the "interior" to "exterior" address binding is dynamic, the only forms of applications that can traverse a NAT are those that are initiated on the "inside" of the NAT boundary. The exterior cannot initiate a transaction with an interior end point simply because it has no way of addressing this remote device. IPv6 allows all devices to be uniquely addressed from a single address pool, allowing for coherent end-to-end packet delivery by the network. This in turn allows for the deployment of end-to-end security tools for authentication and encryption and also allows for true peer-to-peer applications.

IPv6, as a protocol architecture, is not a radical departure from the architecture of IPv4. The same datagram delivery model is used, with the same minimal set of assumptions about the underlying network capabilities, and the same decoupling of the routing and forwarding capabilities. The use of an address field in the IP header to contain the semantics of both location and identity was not altered in any fundamental way. The changes made by IPv6 could be seen as conservative set of decisions, based on falling back to the IPv4 protocol model for guidance, on the principle that IPv4 is an operating proof of concept for this architectural approach.

In such a light, IPv6 can be seen as an attempt to regain the advantage of the original IP network architecture: that of a simple and uniform network service that allows maximal flexibility for the operation of the end-to-end application.

It is often the case that complex architectures scale very poorly, and from this perspective the core of IPv6 appears to be a readily scalable architecture.

The Mythology of IPv6

Good as all this is, these attributes alone have not been enough so far to propel IPv6 into broad-scale deployment, and consequently there has been considerable enthusiasm to discover additional reasons to deploy IPv6. Unfortunately, most of these reasons fall into the category of myth, and in looking at IPv6 it is probably a good idea, as well as fair sport, to expose some of these myths as well.

"IPv6 Is More Secure"

A common claim is that IPv6 is more "secure" than IPv4. It is more accurate to indicate that IPv6 is no more or less secure than IPv4. Both IPv4 and IPv6 offer the potential to undertake secure transactions across the network, and both protocols are potentially highly capable in attempting to undertake highly secure transactions. Yes, the IPv6 specification includes as mandatory support for *Authentication and Encapsulating Security Payload* extension headers, but no, there is no "mandatory to use" sticker associated with these extension headers, and, like IPv4 *IP Security* (IPSec), it is left to the application and the user to determine whether to deploy security measures at the network transport level. So, to claim that IPv6 is somehow implicitly superior to IPv4 is an overly enthusiastic claim that falls into the category of "IPv6 myth."

Now I should qualify this, because there is a distinction between the protocol and its environment of deployment. In the case of IPv4, this protocol capability is compromised in many environments in the face of various forms of deployed active middleware such as NAT. It's too early to tell with IPv6, but the line of argument is that NAT-based active middleware has been deployed as a means of address extension, and in a IPv6 world such devices are no longer necessary, and will not be deployed. So perhaps one could say that IPv6 enables a path toward widespread peer-to-peer authentication and transport security at the protocol level, but whether the deployment models faithfully follow along such a path remains an open question.

"IPv6 Is Required for Mobility"

It is also claimed that only IPv6 supports mobility. If one is talking about a world of tens of billions of mobile devices, then the larger IPv6 address fields are entirely appropriate for such large-scale deployments. IPv6 includes a developing concept of stateless autoconfiguration and *Neighbor Discovery* mechanisms.

But if the claim is more about the technology to support mobility than the number of mobile devices, then this claim also falls short. The key issue with mobility is that mobility at a network layer requires the network to separate the functions of providing a unique identity for each connected device, and identifying the location within the network for each device.

As a device "moves" within the network its identity remains constant while its location is changing. IPv4 overloaded the semantics of an address to include both identity and locality within an address, and IPv6 did not alter this architectural decision. In this respect, IPv4 and IPv6 offer the same levels of support for mobility. Both protocols require an additional header field to support a decoupled network identity, commonly referred to as the "home address," and then concentrate on the manner of the way in which the home agent maintains a trustable and accurate copy of the mobile node or current location of the network. This topic remains the subject of activity within the IETF in both IPv4 and IPv6.

"IPv6 Is Better for Wireless Networks"

Mobility is often associated with wireless, and again there has been the claim that somehow IPv6 is better suited for wireless environments than IPv4. Again this is well in the realm of myth.

Wireless environments differ from wireline environments in numerous ways. One of the more critical differences is that a wireless environment may experience bursts of significant levels of bit error corruption, which in turn will lead to periods of non-congestion-based packet loss within the network. A TCP transport session is prone to interpreting such packet loss as being the outcome of network level congestion. The TCP response is not only retransmission of the corrupted packets, but also an unnecessary reduction of the sending rate at the same time. Neither IPv4 nor IPv6 have explicit signaling mechanisms to detect corruption-based packet loss, and in this respect the protocols are similarly equipped, or ill-equipped as in this case, to optimize the carriage efficiency and performance of a wireless communications subnet.

"IPv6 Offers Better QoS"

Another consistent assertion is that IPv6 offers "bundled" support for differentiated *Quality of Service* (QoS), whereas IPv4 does not. The justification for this claim often points to the 20-bit flow label in the IPv6 header as some kind of instant solution to QoS. This claim conveniently omits to note that the flow identification field in the IPv6 header still has no practical application in large-scale network environments. Both IPv4 and IPv6 support an 8-bit traffic class field, which includes the same 6-bit field for differentiated service code points, and both protocols offer the same fields to an *Integrated Services* packet classifier. From this perspective, QoS deployment issues are neither helped nor hindered by the use of IPv4 or IPv6. Here, again, it is a case of nothing has changed.

"Only IPv6 Supports Auto-Configuration"

Another common claim is that only IPv6 offers "plug-and-play" auto-configuration. Again this is an overenthusiastic statement, given the widespread use of the *Dynamic Host Configuration Protocol* (DHCP) in IPv4 networks these days. Both protocol environments support some level of "plug-and-play" auto-configuration capability, and in this respect the situation is pretty much the same for both IPv4 and IPv6.

"IPv6 Solves Routing Scaling"

It would be good if IPv6 included some novel approach that solved, or even mitigated to some extent, the routing scaling issues. Unfortunately, this is simply not the case, and the same techniques of address aggregation using provider hierarchies apply as much to IPv6 as they do to IPv4. The complexity of routing is an expression of the product of the topology of the network, the policies used by routing entities, and the dynamic behavior of the network—not the protocol being routed. The larger address space does little to improve on capability to structure the address space in order to decrease the routing load. In this respect IPv6 does not make IP routing any easier, nor any more scalable.

"IPv6 Provides Better Support for Rapid Prefix Renumbering"

If provider-based addressing is to remain an aspect of the deployed IPv6 network, then one way to undertake provider switching for multi-homed end networks is to allow rapid renumbering of a network common prefix. Again, it has been claimed that IPv6 offers the capability to undertake rapid renumbering within a network to switch to a new common address prefix. Again IPv6 performs no differently from IPv4 in this regard. As long as "rapid" refers to a period of hours or days, then yes, IPv4 and IPv6 both support "rapid" local renumbering. For a shorter time frame for "rapid," such as a few seconds or even a few milliseconds, this is not really the case.

"IPv6 Provides Better Support for Multihomed Sites"

This leads on to the more general claim that IPv6 supports multi-homing and dynamic provider selection. Again this is an optimistic claim, and the reality is a little more tempered. Multihoming is relatively easy if you are allowed to globally announce the network address prefix without recourse to any form of provider-based address aggregation. But this is a case of achieving a local objective at a common cost of the scalability of the entire global routing system, and this is not a supportable cost. The objective here is to support some form of multihoming of local networks where any incremental routing load is strictly limited in its radius of propagation. This remains an active area of consideration for the IETF and clear answers, in IPv4 or IPv6, are not available at present. So at best this claim is premature, and more likely the claim will again fall into the category of myth rather than firm reality.

"IPv4 Has Run Out of Addresses"

Again, this is in the category of myth rather than reality. Of the total IPv4 space, some 6 percent is reserved and another 6 percent is used for multicast. Forty-one percent of the space has already been allocated, and the remaining 37 percent (or some 1.5 billion addresses) is yet to be allocated. Prior to 1994, some 36 percent of the address space had been allocated. Since that time, and this includes the entire Internet boom period, a further 15 percent of the available address space was allocated. With a continuation of current policies it would appear that IPv4 address space will be available for many years yet.

So Why IPv6 Anyway ?

The general observation is that IPv6 is not a "feature-based" revision of IPv4—there is no outstanding capability of IPv6 that does not have a fully functional counterpart in IPv4. Nor is there a pressing urgency to deploy IPv6 because we are about to run out of available IPv4 address space in the next few months or even years within what we regard as the "conventional" Internet.

It would appear that the real drivers for network evolution lurk in the device world. We are seeing the various wireless technologies, ranging from Bluetooth for personal networking through the increasingly pervasive IEEE 802.11 "hot-spot" networking to the expectations arising from various forms of *third-generation* (3G) large radius services being combined with consumer devices, control systems, identification systems, and various other forms of embedded dedicated function devices. The silicon industry achieves its greatest advantage through sheer volume of production, and it is in the combination of Internet utility with the production volumes of the silicon industry that we will see demands for networking that encompasses tens, if not hundreds, of billions of devices. This is the world where IPv6 can and will come into its own, and I suspect that it is in this device and utility mode of communications that we will see the fundamental drivers that will lead to widespread deployment of IPv6 support networks.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Scientist in the Internet area for Telstra. He is also the Executive Director of the Internet Architecture Board, and is a member of the APNIC Executive Committee. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@telstra.net**

Letters to the Editor

SIP Typos

Dear Mr. Stallings, and Mr. Jacobsen,

The *Session Initiation Protocol* article by Mr. Stallings in the *Internet Protocol Journal,* Volume 6, Number 1, March 2003, provides an excellent tutorial on the subject, IMHO.

The article does an extraordinary job at presenting what is quite a complicated protocol (SIP) in simple terms. However, there seem to be some typographical errors in the article, which I wanted to bring to your attention:

- In Figure 2, message number 10 should be "180 Ringing" as opposed to "100 Ringing."
- In Figure 2, the line under message number 14 should be pointing in the opposite direction (that is *from* Bob's proxy *to* Alice's proxy).
- In Figure 2, message number 16 should read only "ACK" not "180 ACK."
- In Figure 2, message number 15 should perhaps read as "200 OK" as opposed to just "OK"
- In Figure 3, message number 5 should read "200 OK" as opposed to "200 Trying"
- Figure 4 message number 5 and 7 should perhaps read as "NOTIFY <Signed In>" as opposed to "<Not Signed-In>"
- Figure 4 "User Agent Bob" should be labelled as "(signed in)" as opposed to "(not signed in)"
- There are missing closing angular brackets in the SIP INVITE message listing on page 27:

  To: Bob **<sip:bob@biloxi.com>**
  From: Alice **<sip:alice@atlanta.com>;tag=...**
- There are missing closing angular brackets in the SIP 200 OK message listing on page 28:

  To: Bob **<sip:bob@biloxi.com>;tag=....**
  From: Alice **<sip:alice@atlanta.com>;tag=...**

Sincerely,

—Rajnish Jain, Excel Switching Corp.
rajnishjain@xl.com

The author responds:

Rajnish,

Thanks for the comments. I am embarrassed that so many errors slipped through, even though I and several reviewers for Ole checked the paper.

—Bill Stallings
ws@shore.net

DoD and IPv6

Dear Geoff,

After reading your article, I couldn't help but notice the U.S. Department of Defense's announcement concerning their intentions to adopt IPv6 in the coming years (see "Fragments," page 38). Given that you've made some strong statements about the value of IPv6 in your article, would you care to offer some views about this announcement?

—Ole

Dear Editor,

As I said in the article, the true value of IP v6 lies in the massive amount of coherent address space that allows literally billions of devices to be uniquely addressed. Address uniqueness is a strong value proposition when you want an identifier space to cover a very large deployment space. As an example of this, one of the two properties of the original Digital-Intel-Xerox Ethernet II specification that remains in today's 10 Gigabit Ethernet specification is unique MAC addresses. All of that highly innovative CSMA/CD thinking that at the time we thought was the fundamental property of Ethernet has been dispensed with.

The general observation is that any communications systems requires any party to be able to uniquely identify any other party in order to initiate a private communication session. If you cannot perform that most basic of communications functions, then you simply do not have a functional peer-to-peer communications network.

But doesn't that mean that the stories of IPv4 address exhaustion have some substance? With the large amount of addressable devices hidden behind NATs, and the associated move to using domain names as the underlying identifier space for many communications applications, the pressure on consumption of IPv4 address space has been reduced considerably. This has implied that in a world of human-driven screens and keyboards we see some considerable lifetime left in the admittedly comfortable world of IPv4 as we know it. To support this model we've actually moved away from the IP address as the unique identifier token for many applications, and substituted an application model that is driven from domain names. As an example, consider the virtual hosting mechanism as implemented in Apache Web servers to see this shift in communications identifiers from address to domain name. And both as consumers of the technology and as an industry we can live with this for some time yet, because we appear to concentrate our use IP addresses as a routing and forwarding framework and increasingly use the DNS as the identifier realm of an application.

But our world is a world where the device is subservient to the user, and the applications we associate with the Internet of today are applications that are essentially human pastimes, such as e-mail, Web browsing, or high-value automated transactions, such as those commonly bracketed into the e-commerce area. And we've now established a highly valuable global industry upon these foundations.

But in so doing we should recognize the emergence of a second set of communications realms populated by uniquely identified devices that number in their billions, where the inter-device traffic is not human mediated, and the value of the device transactions are, on an individual transactions value level, far lower than the value of the human-driven realm of IPv4. In other words, in a device rich communications realm, it's likely that the human value we'd ascribe on average to each packet is far lower than our current Internet IPv4 world of human-mediated communications. And it's this extravagantly device-equipped world that we see the U.S. Department of Defense heading. If your stock in trade is one of quite astounding feats of logistical deployment of large numbers of people and large numbers of items of equipment, then the communications requirement is of a different order of scale to that of the retail Internet markets, and, yes, I'm sure that there are entirely effective arguments behind that decision to look forward to a communications realm with a uniform base protocol identifier domain in a scale that is 2 to the power 96 times larger than the entire IP address identifier domain of IPv4.

But I would be cautious about high levels of expectation that this immediately translates into an impetus in the market where you and I converse. My host here where I'm typing this message is already IPv6 capable, and if you are running a recent version of host software, then it's a reasonable assumption that yours is too. But I'll send this message over IPv4 and you'll receive it over IPv4, and between my mail sender and your mail receiver the transport channel will also be IPv4. Should we use IPv6 instead? Would I pay my provider additional money to compensate it for part of its additional expenditure to support a simultaneous IPv6 capable network between you and me? To send precisely the same message? In precisely the same time? Along the same path? Using the same transport TCP session? Obviously, to me, as a (hopefully) economically rational consumer of such services, and no doubt to you, in a similar role, there is no value in spending more money to achieve outcomes in IPv6 that are identical to what we can already do today in IPv4. And in the retail Internet world that remains the basic IPv6 conundrum. Why should any provider spend additional resources to service the same market with identical services, and in so doing be unable to raise additional revenue to offset their additional service costs? One interpretation is that there is no natural motivation for such activities in today's market, otherwise it would already be very widespread indeed.

What we've seen in the mainstream Internet world is an emerging mythology about IPv6 that somehow this additional expenditure, ultimately on the part of the consumer, provides some additional benefit for the consumer, motivating them to switch from IPv4-only services to some hybrid of mixed v4 and v6 and ultimately to a v6 world, and thereby funding the additional provider expenditure associated with such a massive transition.

The reality is more sobering in that in the retail Internet world there is so far nothing obvious in the "additional benefit" category. I'm using *Network Address Translation* (NAT) right now, using an *ssh* session back to my mail server that drives through NAT boxes to make a secure SMTP session, across a first step of 802.11 wireless in order to send this message to you.

I've auto-configured in the wireless world, and for me I'm living in a plug-and-play world that supports my level of roaming access. Would IPv6 make this session any more secure? Any different in terms of *Quality of Service* (QoS) ? In plug-and-play models of roaming? Would there be any visible difference in terms of my ability to communicate with you? To all of these questions the basic answer is still "no."

So, for you and I, we look inside the IPv6 technology box, and find nothing new there to motivate us to spend more money for our existing Internet-based communications services, and for some time to come it would appear that this will still hold.

On the other hand there are circumstances where there is a need to operate in a much larger base protocol address space. These include situations where one wants to take advantage of Internet applications that operate across a world of literally billions of devices, large and small. The application space may want to gather constant reports on the characteristics of the "thing" it is attached to, from a ration pack to a component of a large naval vessel. You may want to use supply channels for such devices such that the deployment is a plug-and-play world without a massive variety of detailed configuration processes. You may be looking to an architecture that would be stable for many years. In such circumstances you really want take advantage of a uniform set of Internet application technologies that potentially span massive numbers of addressable devices. Here a large base address space is a definite asset. And for such industry sectors in voicing such requirements where there is also a somewhat different ultimate value proposition for the supported communications activity, then it's quite understandable that there can be an attractive proposition offered by immediate adoption of IPv6.

But back in the communications realm where you and I currently exchange our messages, such requirements remain in a future framework that is still waiting for relevant value propositions that allow it to gain traction with you and me. And as I attempted to point out in the article, adding some elements of mythology and over-stating the IPv6 value case won't help here.

Maybe we just need to be patient. Steam ships did not halt operation the first day a diesel powered vessel appeared. It was a much slower process that lead to an outcome of the change of the maritime fleet—the next generation of mechanization offered cheaper services, and, as often happens, market price won in that commodity market.

Market price often wins in competitive commodity markets. And the Internet retail market is, in many parts of the world and in many sectors, a strongly competitive space with all the characteristics of a commodity offering. In addressing such initial specialized dedicated communications requirements with IPv6 technology as represented by the U.S. DoD, there is a distinct possibility that there may be some effective use of initial investment that translates into the retail world in some form of efficiency gain for IPv6-capable providers.

And there no doubt that if you and I could communicate in precisely the same fashion as we do today, with precisely the same applications and service environment, using precisely the same host devices and operating systems as we do today, but at some attractive fraction of today's price, then I'm sure that neither of us would care in the slightest that our data was encapsulated using a packet framing format and address tokens that used the IPv6 protocol specifications.

Kind regards,

—Geoff Huston, Telstra
gih@telstra.net

Book Review

Google Hacks

Google Hacks: 100 Industrial-Strength Tips & Tools, by Tara Calishain and Rael Dornfest, ISBN 0-596-00447-8, O'Reilly & Associates, 2003, 329 pages.

Hmm, this is a hard one. This is the second go at writing a review—the first one made me sound like a grumpy luddite and I don't want my secret identity to be revealed yet. So, put on some suitable music ("So What" from "Kind of Blue" by Miles Davis) and this time, to start with, "just the facts, ma'am" and we'll get back to the grumpiness later.

What we have here are "100 Industrial-Strength Tips & Tools" for using the Google search engine (or g**gling as we are not allowed to say). All the usual O'Reilly positives about layout and presentation apply so we can take those as read (and the usual negative about murky grey scale illustrations). The tips/tools are gathered into separate sections dealing with searching (surprise!), services, scraping, using the API, games and Web mastering. All the tips have some description, some have code and others have URLs that take you to the code or the service described. And indeed some of these are quite interesting and useful, but, and the grumpiness is starting to creep in again, many of them are really not. Tip #1 for instance—"Setting Preferences." Since when has a brief description of how what you can find on the Google preferences page been "Industrial-strength"? Too many of the tips are like this—simple stuff that you can get from many places on the Web (including Google itself) with little added value. Someone starting out using Google is not going to buy a book called *Google Hacks* because its title is off-putting, and someone who is a regular user of the service is going to know (or not be interested in) most of the content. Why do we need a 300 page paper copy of this information? Much of what is in here could be boiled down into a small, cheap guide just like those O'Reilly have for programming languages, and the rest of the stuff is irrelevant anyway (for instance the TouchGraph browser is fun and interesting, but it isn't really that useful—everyone I know has played with it for 5 minutes and then never returned).

I had better hopes of the API programming material, but it was not to be. I know I am in a tiny minority here, so don't complain, but most of the program examples provided in the book use *Perl*. "Hurrah" say you, "Boo" say I—I don't like Perl, never have and never will. Just like celery. I can put up with it, but I won't pick it when I have a choice.

Note, I am not knocking the Google APIs (though they are a bit baroque, and it would be nice to be able to get more than 10 results at time, and...). Being able to call up a search engine from within a program is a good thing, even if you do have to use Web Services (I'm not that keen on them either—are you surprised?). This book certainly tells you how to do that (at least from within Perl) but again you can pick that info up from the Web for free and it doesn't run to more than twenty pages tops. Most of the programming examples may have been fun to write and think up but are about as useful as a flowchart stencil.

Oh, and "Googlewhacking"[1] is not new—people were doing that on AltaVista long (in Internet terms) before Google appeared.

All things considered, I don't see this book being worth $25. If you know how to use Google even a little bit you ought to be able to use it to find all this information without it. And what of the stablemate book *Amazon Hacks* which is due to appear soon? I fear a miracle of padding there.

—Lindsay Marshall, University of Newcastle upon Tyne
Lindsay.Marshall@newcastle.ac.uk

[1] Googlewhacking is the art of finding a two-word query that has only one result. The two words may not be enclosed in quotes, and the words must be found in Google's own dictionary (no proper names, made-up words, etc).

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Fragments

Several Landmarks Define Push toward IPv6 Deployment in Japan

In April 1998, the KAME Project, **http://www.kame.net/**, an extension of the WIDE Project (**http://www.wide.ad.jp/**; representative Professor Jun Murai, Keio University), was established with eight core members from seven Japanese vendors. Work began under a two-year timeframe to provide free IPv6/IP Security (IPSec) reference code for UNIX BSD variants. The KAME Project remains active today.

The Japanese government's commitment to taking a leadership role in worldwide IPv6 research and deployment was outlined in a speech to open the September 2000 Diet session by then Prime Minister Mori. Mori identified IPv6 as a key discussion area for the national IT Strategy Council—a strategic pillar toward the "rebirth of the nation."

The *IPv6 Promotion Council of Japan* was established shortly thereafter, in Oct. 2000. Its founding members numbered only 18. As of March 2003 the Council's membership body consisted of 320 organizations from a variety of business fields; carriers, *Internet Service Providers* (ISPs), hardware vendors, software vendors, finance companies, general trading companies, automobile manufacturers, etc.

The Council is the most active and influential IPv6 organization in Japan, and is the formal contact point appointed by the Japanese government to handle requests from overseas private IPv6 promotion bodies, such as the various regional IPv6 Task Force bodies, for technical and deployment cooperation.

The Promotion Council is currently running the "IPv6 Appli-Contest 2003." The contest awards developers of applications and software who help to create new possibilities in the IPv6 Internet world, see: **http://www.v6pc.jp/apc/en/concept.html**

Supported by the Ministry of Public Management, Home Affairs, Posts and Telecommunications, and the WIDE Project, the contest is drawing on the cooperation of IPv6 bodies in the EU, North America, India, Korea, Taiwan, and China with the goal of creating a library of freely available IPv6 software.

Details on rules and regulations for entry can be found at the following URL: **http://www.v6pc.jp/apc/en/regulations.html**.
The deadline for entries is August 31, 2003.

Six entries will be selected as "Award of Excellence" winners and will share 1,500,000 JPY in prize money. Award of Excellence winners will also be eligible for the "Grand Prize" of 1,000,000 JPY to be presented at a ceremony during WPC EXPO 2003 to be held September 17–20, 2003, in Tokyo.

An excellent, up-to-date overview of the current status of IPv6 research and commercial service offerings in Japan, including IPv6 case studies and technology tutorials, can be found at IPv6style:
http://www.ipv6style.jp/en/index.shtm

US Department of Defense adopts IPv6

Implementation of the next-generation Internet protocol that will bring the Department of Defense closer to its goal of net-centric warfare and operations was announced on June 13, 2003 by John P. Stenbit, assistant secretary of defense for networks and information integration and DoD chief information officer.

The new Internet protocol, known as IPv6, will facilitate integration of the essential elements of DoD's Global Information Grid—its sensors, weapons, platforms, information and people. Secretary Stenbit is directing the DoD-wide transition.

The current version of the Internet's operating system, IPv4, has been in use by DoD for almost 30 years. Its fundamental limitations, along with the world-wide explosion of Internet use, inhibit net-centric operations. IPv6 is designed to overcome those limitations by expanding available IP address space, improving end-to-end security, facilitating mobile communications, enhancing quality of service and easing system management burdens.

"Enterprise-wide deployment of IPv6 will keep the warfighter secure and connected in a fast-moving battlespace," Secretary Stenbit said. "Achievement of net-centric operations and warfare depends on effectively implementing the transition."

Secretary Stenbit signed a policy memorandum on June 9 that outlines a strategy to ensure an integrated, timely and effective transition. A key element of the transition minimizes future transition costs by requiring that, starting in October 2003, all network capabilities purchased by DoD be both IPv6-capable and interoperable with the department's extensive IPv4 installed base.

For more information, see:

http://www.dod.gov/news/Jun2003/d20030609nii.pdf

http://www.dod.gov/releases/2003/nr20030613-0097.html

http://www.dod.gov/news/Jun2003/n06132003_200306134.html

http://www.dod.gov/transcripts/2003/tr20030613-0274.html

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

September 2003 Volume 6, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

The task of adding security to Internet protocols and applications is a large and complex one. From a user's point of view, the security-enhanced version of any given component should behave just like the old version, just be "better and more secure." In some cases this is simple. Many of us now use a *Secure Shell Protocol* (SSH) client in place of *Telnet*, and shop online using the secure version of HTTP. But there is still work to be done to ensure that *all* of our protocols and associated applications provide security. In this issue we will look at *routing*, specifically the *Border Gateway Protocol* (BGP) and efforts that are underway to provide security for this critical component of the Internet infrastructure. As is often the case with emerging Internet technologies, there exists more than one proposed solution for securing BGP. Two solutions, S-BGP and soBGP, are described by Steve Kent and Russ White, respectively.

The Internet gets attacked by various forms of viruses and worms with some regularity. Some of these attacks have been quite sophisticated and have caused a great deal of nuisance in recent months. The effects following the *Sobig.F* virus are still very much being felt as I write this. Tom Chen gives us an overview of the trends surrounding viruses and worms.

Closely related to the virus attacks is *spam*. Unfortunately, I know of no complete technical, or even legal, solutions to this growing problem, but I would love to hear your views and solutions. Send your comments to: **ipj@cisco.com**, but don't use the string "spam" in the subject field or it may get filtered out!

Following Geoff Huston's opinion piece "The Myth of IPv6" in our previous issue, we received a response from *The IPv6 Forum*. The article is entitled "IPv6 Behind the Wall" and is by Jim Bound.

I was very pleased to hear that professor Peter T. Kirstein of University College London had been awarded the Internet Society's *Jonathan B. Postel Service Award* for 2003. I have known Peter since about 1977, when we collaborated on SATNET packet voice conferences between Oslo, London, Boston, and Marina del Rey. Peter is truly an Internet pioneer. (See "Fragments," page 41).

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Securing the Border Gateway Protocol

by Stephen T. Kent, BBN Technologies

Routing in the public Internet is based on a distributed system composed of many routers, grouped into management domains called *Autonomous Systems* (ASes). ASes are operated by *Internet Service Providers* (ISPs) and by multihomed subscribers. (Throughout the remainder of this article, for brevity, we will talk in terms of ISPs, usually omitting references to multihomed subscribers.) Routing information is exchanged between ASes using the *Border Gateway Protocol* (BGP)[1], via UPDATE messages.

BGP is used in two different contexts. *External BGP* (eBGP) propagates routes between ISPs. BGP also is used within an AS to propagate routes acquired from other ASes. This latter use is referred to as *internal BGP* (iBGP). eBGP is the primary focus of this article, because failures of eBGP can adversely affect large portions of the Internet, well beyond the administrative boundary of the source of the failure. Nonetheless, some ISPs have expressed interest in protecting the distribution of routes within an ISP. The security technology discussed in this article can be used to secure iBGP, but eBGP is the focus of this article. We use the term "BGP" to refer to eBGP throughout the article.

BGP is highly vulnerable to a variety of attacks[2]. In some cases, this vulnerability arises because of a lack of integrity and authentication for BGP messages. However, the more substantive and harder problem is the lack of a secure means of verifying that BGP traffic is authorized, a concept explored in more detail in this article. In April 1997, BBN began work on the security architecture described here, a system we refer to as *S-BGP,* to address the vulnerabilities of BGP. This article begins by reviewing the problem, discusses a model for correct operation of BGP, presents a threat model, and states the goals and assumptions that underlie our proposed security architecture.

Before we begin the discussion of BGP in more detail, a few definitions are in order. A *route* is defined as an *address prefix* and a set of *path attributes*. One of the path attributes is an AS path, and that is the primary focus of BGP security considerations. The AS path specifies the sequence of ASes that subscriber traffic should traverse if forwarded via this route. When propagating an UPDATE to a neighboring AS, the BGP router prepends its AS number to the sequence, and may update certain other path attributes. The first AS included in the path is referred to as the *origin AS*.

Each BGP router (other than at the edges of the Internet) maintains a complete routing table, capable of routing traffic to any reachable destination, and sends its best route for each prefix to each neighbor. In BGP, "best" is very locally defined. The BGP route selection algorithm has few criteria that are universal, thus limiting the extent to which any security mechanism can detect and reject "bad" routes emitted by a neighbor.

Each ISP makes use of local policies that it need not disclose, and this gives BGP route selection a "black box" flavor, which has significant adverse implications for security.

Correct Operation of BGP

Security for BGP should be defined as the correct operation of BGP routers. This definition is based on the observation that any successful attack against BGP will result in other than correct operation, presumably yielding degraded routing. Correct operation of BGP depends upon the integrity, authenticity, and timeliness of the routing information it distributes, as well as each BGP router processing, storing, and distributing this information in accordance with both the BGP specification and local routing policies. Many statements could be made in an effort to characterize correct operation, but they rest on two simple assumptions.

First, control (vs. subscriber traffic) communication between neighbor BGP routers must be authenticity and integrity secure. This is easily achieved through the use of a point-to-point security protocol capable of protecting BGP traffic; for example, *IP Security* (IPSec). Second, BGP routers must execute the route selection algorithm correctly and communicate the results. There are two parts to this assumption: processing received UPDATEs, and generation and transmission of UPDATEs. In terms of an AS trying to protect itself against external attacks, correct operation of its own BGP routers is mostly a local security issue, but not an Internet-wide security issue. However, an AS should not rely on other ASes to operate properly; such reliance permits a failure in one AS to propagate to others, a domino failure effect. Thus it is important for a BGP router to be able to verify that each UPDATE it receives from a peer is valid (authorized) and timely.

The validity of an UPDATE message is based on four primary criteria:

- The router that sent the UPDATE was authorized to act on behalf of the AS it claims to represent; that is, the AS at the front of the AS path.
- The AS from which the UPDATE emanates was authorized by the preceding AS in the AS path (in the UPDATE message) to advertise the prefixes in the UPDATE.
- The first AS in the AS path was authorized, by the owner of the set of prefixes that are represented in the UPDATE, to advertise those prefixes.
- If the UPDATE withdraws one or more routes (specified by the prefixes for the routes), then the sender must have advertised each route prior to withdrawing it.

There are some limitations to the ability of any practical security mechanism to detect all BGP security failures. The local policy feature of BGP allows each ISP considerable latitude in how UPDATEs are processed, making it difficult for an external observer—for example, a router in a neighboring AS—to determine if a router is operating properly.

This is because such behavior might be attributed to local policies not visible outside an AS. To address such attacks, the semantics of BGP itself would have to change. Moreover, because UPDATEs do not carry sequence numbers, a BGP router can emit an UPDATE based on authentic, but old, information; for example, withdrawing or reasserting a route based on outdated information. Thus the temporal accuracy of UPDATEs, in the face of Byzantine failures, is hard to enforce, except in a very coarse fashion. (Simply speaking, a *Byzantine failure* is one in which a nominally trusted or authorized entity misbehaves.)

Threat Model and BGP Vulnerabilities

Routers exhibit both architectural and implementation vulnerabilities. Implementation vulnerabilities are the result of errors that arise in developing design details or coding; for example, translating the BGP specs into software. Architectural vulnerabilities permit various forms of attack, independent of implementation details, and thus are potentially more damaging, because they persist across all implementations. To make Internet routing robust, both forms of vulnerabilities must be addressed. BGP vulnerabilities can be exploited to cause improper routing or nondelivery of subscriber traffic, network congestion, and traffic delays. Misrouting attacks can be used to facilitate both passive and active wiretapping of subscriber traffic. Often an attack against BGP may be part of a larger attack against subscriber computers. For example, there have been BGP attacks that seek to misroute queries to *Domain Name System* (DNS) root servers, as part of an attack against subscriber systems.

BGP can be attacked in many ways. Communication between BGP peers can be subjected to active or passive wiretapping. The BGP software, configuration information, or routing databases of a router may be modified or replaced via unauthorized access to a router, or to a server or management workstation from which router software is downloaded. These latter attacks transform routers into hostile insiders, so security measures must address such Byzantine failures.

Improved physical and procedural security for network management facilities, and routers, and cryptographic security for BGP traffic between routers would help reduce some of these vulnerabilities. However, physical and procedural security is expensive and imperfect, and these countermeasures would not protect the Internet against accidental or malicious misconfiguration by operators, nor against attacks that mimic such errors. Misconfiguration of this sort has been a source of Internet outages in the past and seems likely to persist. Any security approach that relies on ISPs to act properly violates the "principle of least privilege" and leaves the Internet routing system vulnerable at its weakest link. In contrast, the security approach described in this article satisfies this principle, so that any attack on any component of the routing system is limited in its impact on the Internet as a whole.

Routers also are susceptible to resource exhaustion attacks based on delivery of large quantities of management traffic, BGP or otherwise. This vulnerability arises because these devices are designed with the not unreasonable model that management traffic is a very tiny percentage of all the traffic that arrives at a router. Router interfaces can deliver traffic to the management processor at very high rates, because they are designed to accommodate subscriber traffic flows. Solutions to this problem need to be generic, to accommodate all types of router management traffic, and thus are outside the scope of the BGP security measures discussed in this article.

Goals, Constraints, and Assumptions

Any proposed security architecture must exhibit dynamics consistent with the existing BGP system; for example, responding automatically to topology changes, including the addition of new networks, routers, and ASes. These actions take place on different time scales and have different scopes. For example, in the current BGP system, if an ISP replaces a failed router, the action can take place fairly quickly and has only local impact, because ISPs are not aware of the identity of routers in other, non-neighboring, ISPs. The issuance of new AS numbers, representing new nets, is not a fast process, nor is the allocation of new blocks of address space (new prefixes). But both of these actions are globally visible. Changes in routes also may have global impact, and they may occur very quickly.

Solutions also must scale in a manner consistent with the growth of the Internet. The countermeasures must be consistent with the BGP protocol standards and with the likely evolution of these standards. This includes packet size limits and features such as path aggregation, communities, and multiprotocol support (for example, *Multiprotocol Label Switching* [MPLS]). The security measures must be incrementally deployable; there cannot be a "flag day" when all BGP routers suddenly begin executing a new security protocol. It is desirable to not create new organizational entities that must be accepted as authorities by ISPs and subscribers, in order to make routing secure.

S-BGP Architecture

S-BGP consists of four major elements:

- A *Public Key Infrastructure* (PKI) that represents the ownership and delegation of address prefixes and AS numbers
- *Address Attestations* that the owner of a prefix uses to authorize an AS to originate routes to the prefix
- *Route Attestations* that an AS creates to authorize a neighbor to advertise prefixes
- *IPSec* for point-to-point security of BGP traffic transmitted between routers

These elements are used by an S-BGP router to secure communication with neighbors, and to generate and validate UPDATE messages relative to the authorization model represented by the PKI and address attestations. Together, the combination of these security mechanisms prevents a compromised AS from propagating erroneous routing data to other, secured ASes. Each element is described in more detail in the following section.

S-BGP Public Key Infrastructure

S-BGP uses a PKI based on X.509 (v3) certificates to enable routers to validate the authorization of other routers to represent ASes (ISPs). The PKI also allows routers to verify the authorization of each ISP as the owner of one or more prefixes (contiguous blocks of address space). This PKI was described in[14], and the reader is referred to that paper for additional details. The PKI parallels the existing IP address and AS number assignment delegation system and takes advantage of this infrastructure. Because the PKI mirrors existing infrastructure, it avoids most of the "trust" issues that often complicate the creation of a PKI. This PKI is unusual in that it emphasizes authorization, not authentication. The names used in the certificates in this PKI are not employed to determine whether a given ISP or router is authorized to do anything, and the names are not even meaningful outside of S-BGP.

S-BGP calls for a certificate to be issued to each ISP (or subscriber) that owns (more properly, has a right to use) a portion of the IP address space. This certificate is issued through the same procedures employed for address allocation, starting with the *Internet Assigned Numbers Authority* (IANA) and continuing through a *Regional Internet Registry* (RIR), and, if applicable, an ISP. If an ISP owns multiple prefixes, we issue a single certificate containing a list of prefixes, to minimize the number of certificates in the system. The PKI represents address-space ownership by binding prefixes to a public key belonging to the ISP to which the prefixes have been assigned. Each certificate contains a private extension that specifies the set of prefixes that has been allocated to the ISP. Certificates issued under this PKI also represent the binding between an ISP and the AS numbers allocated to it. The PKI allows each ISP to issue certificates to its routers, certifying that these routers represent the ISP and hence, the ASes owned by the ISP. Here too, the PKI parallels the existing AS allocation system; that is, the IANA allocates AS numbers to RIRs, which in turn assign AS numbers to ISPs that run S-BGP.

Attestations

An *attestation* is a digitally signed datum asserting that its target (an AS) is authorized by the signer (an ISP) to advertise a path to one or more specified prefixes. There are two types of attestations, address and route, which share a common format. For an *Address Attestation* (AA), the signer is the ISP or subscriber that controls the prefixes in the AA, and the target is a set of ASes that the ISP/subscriber authorizes to originate a route to the prefixes. AAs are relatively static data items, because relationships between address-space owners and ISPs change relatively slowly.

For a *Route Attestation* (RA), the signer is an S-BGP router (operating on behalf of an ISP), and the target is an AS or set of ASes, representing the neighbors to which the UPDATE containing the RA will be sent. RAs, unlike AAs, are very dynamic, possibly changing for each transmitted UPDATE.

UPDATE Validation

Attestations and certificates are used by S-BGP routers to validate routes asserted in UPDATE messages; that is, to verify that the first AS in the route has been authorized to advertise the prefixes by the prefix owner(s), and that each subsequent AS has been authorized to advertise the route for the prefixes by the preceding AS in the route. To validate a route received from AS_n, AS_{n+1} requires:

- An AA for each organization owning a prefix represented in the UPDATE (not for prefixes in the UPDATE that represent routes being withdrawn)
- A certified public key for each organization owning a prefix in the UPDATE
- An RA corresponding to each AS along the path (AS_n to AS_1), where the RA generated and signed by the router in AS_n encompasses the *Network Layer Reachability Information* (NLRI) and the path from AS_{n+1} through AS_1
- A certified public key for each S-BGP router that signed an RA along the path (AS_n to AS_1), to check the signatures on the corresponding RAs

An S-BGP router verifies that the advertised prefixes and the origin AS are consistent with AA information. The router verifies the signature on each RA and verifies the correspondence between the signer of the RA and the authorization to represent the AS in question. There also must be a correspondence between each AS in the path and an appropriate RA. If all of these checks pass, the UPDATE is valid.

AAs are not used to check withdrawn routes in an UPDATE. Use of IPSec to secure communication between each pair of S-BGP routers, plus the fact that BGP uses a separate *Adjacency Routing Information Base* (Adj-RIB-In) for each neighbor, ensures that only the advertiser of a route can withdraw it.

Distribution of S-BGP Data

Each S-BGP router must have the public keys required to validate the RAs in UPDATEs, a scenario that translates into securely distributed keys for every router that implements S-BGP (and that is reachable via an S-BGP path). Each router also needs access to all AA information, to verify that the origin AS is authorized to originate a route to the prefixes in the UPDATE. S-BGP does not distribute certificates, *Certificate Revocation Lists* (CRLs), or AAs via UPDATE messages; transmission of these items via UPDATEs would be very wasteful of bandwidth, because each BGP router would receive many redundant copies from its neighbors.

Also, an UPDATE is limited to 4096 bytes and thus generally could not carry all of this data for the route represented by the UPDATE. Instead, S-BGP distributes this data to routers via out-of-band means. The data is relatively static and thus is a good candidate for caching and incremental update. Moreover, the certificates and AAs can be validated and reduced to a more compact format by ISP operation centers prior to distribution to routers. This avoids the need for each router to perform this processing, saving both bandwidth and storage space. It also means that routers do not need to be able to parse X.509 certificates and validate certificate paths for S-BGP purposes, although some capability in this area may be required for IPSec key management.

S-BGP uses *repositories* for distribution of this data. We initially described a model in which a few replicated, loosely synchronized repositories were operated by the RIRs. Discussions with ISPs suggest a model in which major ISPs and Internet exchanges operate repositories, and smaller ISPs and subscribers make use of these repositories. In either model, each ISP periodically, for example daily, uploads new/changed certificates, its current CRL, and AAs. Each ISP also downloads all of this data for all other ISPs that are running S-BGP. The repositories periodically transfer new data to one another to maintain loose synchronization. ISPs process the repository information to create more compact files that contain the AA data and the public keys and prefix and AS data from the certificates, but none of the certificate management information or CRLs. These resulting "extracted" files are transferred to the routers executing S-BGP under the control of the ISP.

Because certificates, AAs, and CRLs are signed and carry validity interval information, they require minimal additional security while in transit to or from a repository or while stored on a repository. Nonetheless, S-BGP employs the *Secure Sockets Layer* (SSL) protocol, with both client and server certificates, to protect access to the repositories, as a countermeasure to denial-of-service attacks. The simple, hierarchic structure of the PKI allows repositories to automatically effect access control checks on the uploaded data, for example, to prevent one ISP from accidentally or maliciously overwriting the certificates, CRLs, and AAs from another ISP.

Distribution of Route Attestations

S-BGP distributes RAs with BGP UPDATEs in a newly defined, optional, *transitive path attribute*. Because routes may change quickly, it is important that RAs accompany the UPDATEs that are validated using them. If any other means of distribution is employed for this data, there is a likelihood that the UPDATEs and the data will be out of synch, creating a conundrum for a router; that is, what should the router do when the UPDATE and the security data differ? RAs employ a compact encoding scheme to help ensure that they fit within the BGP packet size limits, even when route or address aggregation occurs. (S-BGP accommodates aggregation by explicitly including signed attribute data that otherwise would be lost when aggregation occurs.) An S-BGP router receiving an UPDATE from a peer caches the RAs with the route in the Adj-RIB for the peer, and in the *Local Routing Information Base* [Loc-RIB] (if the route is selected).

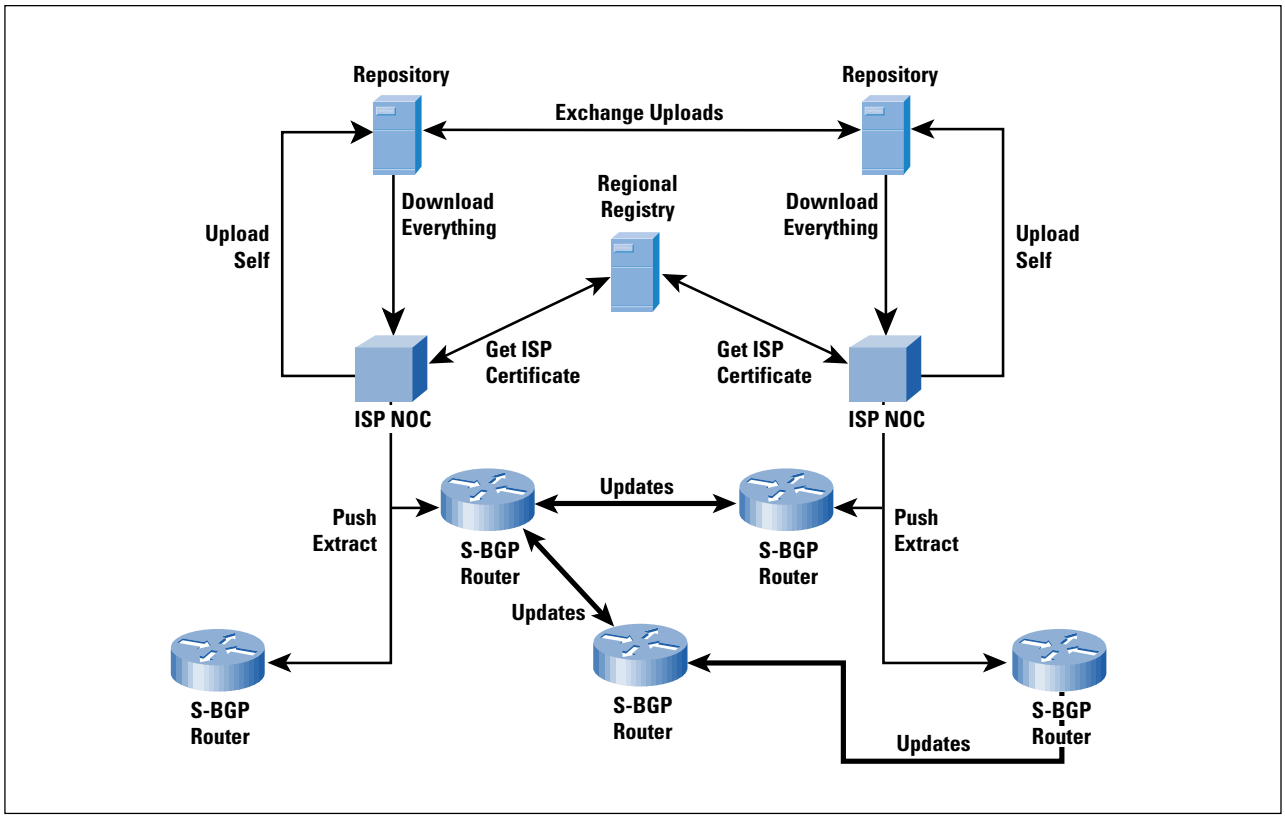
As noted in the following discussion, the bandwidth required to support in-band distribution of route attestations is negligible (compared to subscriber traffic).

Although the RA mechanism was designed to protect AS path data, it can also accommodate other new path attributes; for example, communities[11] and confederations[12]. Specifically, there is a provision to indicate what data, in addition to the AS path, is covered by the digital signature that is part of the RA.

Putting It All Together

Figure 1 illustrates how the major elements of S-BGP interact, using a simplified example. The figure shows two ISPs, each with a *Network Operations Center* (NOC), a repository, and three routers. A third ISP is represented by a single (S-BGP-enabled) router. Each ISP interacts with an RIR to acquire a certificate representing the prefixes and AS numbers assigned to the ISP. Each NOC interacts with a repository to upload data (certificates, CRLs, and AAs) from that ISP, and to download the same data acquired from all other ISPs. The repositories interact with one another to exchange uploaded ISP data, to make that data available to all other ISPs. Within an ISP, the NOC pushes a copy of the extracted certificate and AA data, produced from the downloads acquired from a repository, to each router. Routers exchange UPDATE messages, containing RAs, that enable validation of each received UPDATE.

Figure 1: S-BGP Element Interactions



IPSec and Router Authentication

S-BGP uses IPSec[6,7,8], specifically the *Encapsulating Security Payload* (ESP) protocol, to provide authentication, data integrity, and antireplay for all BGP traffic between neighboring routers. The *Internet Key Exchange* (IKE) protocol[9,10] is used for key management services in support of ESP. The S-BGP PKI includes certificates for IKE, separate from those used for RA processing.

The use of IPSec is preferable to the current option of the *Message Digest Algorithm 5* (MD5) TCP checksum option[15], in several respects. IPSec uses keyed hash functions in a way that is cryptographically more secure that the MD5 checksum option, and IKE provides automated key management, a feature sorely lacking in the option. Protecting BGP traffic at the IP layer, vs. the TCP layer, counters more vulnerabilities, because the TCP implementation is protected as well, for example, including SYN flooding and spoofed RSTs (resets), are rejected.

Residual Vulnerabilities in S-BGP

Despite the extensive security offered by S-BGP, architectural vulnerabilities exist that are not eliminated by its use. For example, an S-BGP router may reassert a route that was withdrawn earlier, even if the route has not been readvertised. The router also may suppress UPDATEs, including ones that withdraw routes. These vulnerabilities exist because BGP UPDATEs do not carry sequence numbers or time stamps that could be used to determine their timeliness. However, RAs do carry an expiration date and time, so there is a limit on how long an attestation can be misused this way. S-BGP restricts malicious behavior to the set of actions for which a router or AS is authorized, based on externally verifiable, authoritative constraints.

Performance and Operational Issues

In developing the S-BGP architecture, we paid close attention to the performance and operational impact of the proposed countermeasures, and reported our analysis in earlier papers. In preparing this article, we updated our data, utilizing a variety of sources; for example, the *Route Views* project. Although much data about BGP and associated infrastructure is available, other data is difficult to acquire in a fashion that is representative of a "typical" BGP router. This is because each AS in the Internet embodies a slightly different view of connectivity, as a result of local policy filters applied by other ASes.

It is important that the transmission, storage, and processing requirements imposed by S-BGP not be so great as to overwhelm routers. Each of these requirements must be analyzed separately.

The transmission of RAs in UPDATEs does significantly increase the size of these messages, by about 800 percent. However, because the volume of this traffic is minuscule relative to subscriber traffic, the increase is negligent. The set of files containing certificates, AAs, and CRLs would be about 75–85 MB. Daily transmission of these files between ISPs and repositories would not represent a significant increase in traffic volume for the Internet.

Although the transmission overhead is not a concern, storage of the RAs in each Adj-RIB and the Loc-RIB is a problem. The additional space required to hold these RAs is estimated at about 30–35 MB per peer, if S-BGP were fully deployed today. This is a modest amount of memory for a typical router with a few peers, but a significant amount of storage for routers at Internet exchanges, where a router may have tens or even hundreds of peers.

Thus the management CPU in a router might need a gigabyte or more of RAM under these conditions. (When a large ISP peers with many other ISPs at an exchange, the peering is not symmetric; that is, the large ISP accepts only a few routes from each of the smaller ISPs, filtering out the rest. Thus the amount of additional memory required for RAs in Adj-RIBs for each of these small ISP peers may be considerably less than for symmetric peer relationships.) This requisite memory seems modest by current workstation standards, but most deployed routers cannot be configured with this much memory.

The computational burden of router processing of RAs in UPDATEs is a function of the path length in each UPDATE and the rate at which UPDATEs arrive. The arrival rate is a function of the number of S-BGP peers the router sees, and the rate at which each peer sends UPDATEs. Our analysis suggests that the long-term (24-hour) UPDATE rate for a router with 30 peers is about 0.5 UPDATEs per second. On average, each UPDATE would contain about 3.7 RAs. We originally estimated the busy minute rate as about 10 times the average rate. At this rate, a router could probably perform the requisite signature verification in software (about 18 signature verifications per second). Recent evidence suggests a factor of 100–200 might be a better estimate, in light of experience with major worm attacks, and at that rate it would be hard for software to keep pace.

Heuristics are available to reduce this burden. Analysis shows that about 50 percent of all UPDATEs are sent as a result of route "flaps"; that is, transient communication failures that, when remedied, result in a return to the former route. Thus if a router maintained a depth-two cache for each Adj-RIB-In, it could avoid signature validation about 50 percent of the time. However, this would double the storage requirements for these RIBs, and that would exacerbate the storage problem cited previously.

Our previous analysis also assumed that receipt of each UPDATE would result in transmission of an UPDATE with one new signature. This was an oversimplification; a router generates and transmits an UPDATE only if the newly received route is "better" than the current best route (for the prefix), or if the best route for the prefix is withdrawn by the UPDATE. When a router has many peers, most of the UPDATEs it receives may not yield a better route, and thus will not trigger transmission of a new UPDATE.

On the other hand, when a router does select a new route, an UPDATE may be constructed and sent to each neighbor, requiring one signature per neighbor. This is because an RA specifies the AS number of the neighbor to which it is directed. It is possible to construct an RA that identifies the next hop as a set of AS numbers, corresponding to all the neighbors to which an UPDATE is authorized to be sent. The downside of this strategy is that it makes the RAs larger, contributing to the storage problem noted previously.

The observation made previously suggests a heuristic for UPDATE processing to mitigate signature validation costs. A router can defer validation of the RAs in any UPDATE that it receives, if the UPDATE would not represent a new best route. This optimization could be especially helpful for routers that receive the greatest number of UPDATEs; that is, routers with many neighbors. One might worry that this strategy allows an attacker to force processing, by sending what would be considered "very good" routes, but an S-BGP router could detect such fraudulent UPDATEs and could choose to drop its connection to a peer that behaved this way, in order to counter such an attack.

Initialization/reboot of a BGP router also results in a surge in UPDATE processing, and the deferred processing heuristic is applicable here too, even though reboots are relatively infrequent. Saving RIBs in nonvolatile storage addresses this problem. Most deployed routers do not have sufficient nonvolatile storage to adopt this strategy, but some do have hard drives that would easily accommodate the RIBs.

It is reasonable to assume that next-generation routers could be configured with enough RAM for the RIBs, but this analysis shows that full deployment is not feasible with the currently deployed router base. To add RAM, and possibly to add nonvolatile storage, router vendors will have to upgrade the processor boards where net management processing takes place. That suggests that addition of a crypto accelerator chip would be prudent as part of the board redesign process, for example, to deal with surge conditions noted previously.

Deployment and Transition Issues

Adoption of S-BGP requires cooperation among several groups. ISPs and subscribers running BGP must cooperate to generate and distribute AAs. Major ISPs must implement the S-BGP security mechanisms in order to offer significant benefit to the Internet community. The IANA and RIRs must enhance operational procedures to support generation of prefix and AS number allocation certificates. Router vendors need to offer additional storage in next-generation products, or offer ancillary devices for use with existing router products, and revise BGP software to support S-BGP.

There is some good news; S-BGP can be deployed incrementally. Only neighboring ASes receive full benefit from such deployment. Although we chose a transitive path attribute syntax to carry RAs, and thus it might be possible for non-neighbor ASes to exchange RAs, it seems likely that intervening ASes would not have sufficient storage for the RAs in their RIBs.

Also, the controls needed in routers to take advantage of noncontiguous deployment of S-BGP are quite complex, hence our suggestion that only contiguous deployment of S-BGP be attempted.

External routes received from S-BGP peers need to be redistributed within the AS, both to interior routers and to other border routers, in order to maintain a consistent and stable view of the exterior routes across the AS. Thus an AS must switch to using S-BGP for all its border routers at once, to avoid route loops within the AS.

Status

As of early 2003, an implementation of S-BGP has been developed and demonstrated on small numbers of workstations representing small numbers of ASes. We also developed software for a simple repository, and for NOC tools that support secure upload and download of certificates, CRLs, and AAs to and from repositories, and for certificate management for NOC personnel and routers. This suite of software, plus CA software from another *Defense Advanced Research Projects Agency* (DARPA) program, provide all of the elements needed to represent a full S-BGP system. All of this software is available in open source form.

Summary

S-BGP represents a comprehensive approach to addressing a wide range of security concerns associated with BGP. It detects and rejects unauthorized UPDATE messages, irrespective of the means by which they arise; for example, misconfiguration, active wiretapping, compromise of routers or management systems, etc. S-BGP is not perfect; it has a few residual vulnerabilities, but these pale in comparison to the security features S-BGP provides, and removal of these vulnerabilities would require more fundamental changes to BGP semantics.

The S-BGP design is based on a top-down security analysis, starting with the semantics of BGP and factoring in the wide range of attacks that have or could be launched against the existing infrastructure.

Acknowledgements

Many individuals contributed to the design and development of S-BGP, including Christine Jones, Charlie Lynn, Joanne Mikkelson, and Karen Seo.

References

[1] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, March 1995.

[2] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE Journal on Selected Areas in Communications,* Vol. 18, No. 4, April 2000.

[3] C. Villamizar, R. Chandra, and R. Govindan, "BGP Route Flap Damping," RFC 2439, November 1998.

[4] B.R. Smith, and J.J. Garcia-Luna-Aceves, "Securing the Border Gateway Routing Protocol," Proceedings of Global Internet '96, November 1996.

[5] S. Murphy, panel presentation on "Security Architecture for the Internet Infrastructure," Symposium on Network and Distributed System Security, April 1995.

[6] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, November 1998.

[7] R. Glenn and S. Kent, "The NULL Encryption Algorithm and Its Use with IPsec," RFC 2410, November 1998.

[8] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)," RFC 2406, November 1998.

[9] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)," RFC 2408, November 1998.

[10] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)," RFC 2409, November 1998.

[11] R. Chandra, P. Traina, and T. Li, "BGP Communities Attribute," RFC 1997, August 1996.

[12] P. Traina, "Autonomous System Confederations for BGP," RFC 1965, June 1996.

[13] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, "Multiprotocol Extensions for BGP-4," RFC 2283, February 1998.

[14] K. Seo, C. Lynn, and S. Kent, "Public-Key Infrastructure for the Secure Border Gateway Protocol (S-BGP)," DARPA Information Survivability Conference and Exposition, June 2001.

[15] A. Heffernan, "Protection of BGP Sessions via the TCP MD5 Signature Option," RFC 2385, August 1998.

STEPHEN KENT received the S.M., E.E., and Ph.D. degrees in computer science from MIT, and a B.S. in mathematics from Loyola University of New Orleans. He has worked at BBN for over 25 years, where he serves today as Chief Scientist–Information Security. He served on the IAB for over a decade, and chaired the Privacy & Security Research Group of the IRTF and the PEM WG in the IETF, where he currently co-chairs the PKIX WG. He has served on several committees for the National Research Council, and chairs a committee on authentication and privacy for the NRC. His current work focuses on PKI issues, BGP security, and very high speed IP encryption. He is a Fellow of the ACM, and a member of the Internet Society and Sigma Xi. His e-mail address is: **kent@bbn.com**

Securing BGP Through Secure Origin BGP

by Russ White, Cisco Systems

Networks have come under increasing scrutiny in the area of security. Routing, the part of the network that provides information on how to reach destinations within the network, has been gaining attention from a security perspective as well. *The Internet Engineering Task Force* (IETF) has, in fact, formed a new working group, the *Routing Protocols Security Requirements Working Group* (**http://www.rpsec.org**), to analyze security in routing systems.

Of course, the biggest network in existence is the Internet, and the routing protocol that provides reachability and path information for the Internet is the *Border Gateway Protocol* (BGP), specified in RFC 1771. Several methods of securing the information carried within BGP have been proposed:

- *Internet Route Verification* (IRV), described in "Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing," Symposium on Network and Distributed Systems Security, February 2003, by Geoffrey Goodell, William Aiello, Timothy Griffin, John Ioannidis, Patrick McDaniel, and Aviel Rubin. IRV relies on out-of-band communication with a route originator to verify the correctness of a route.
- S-BGP, described in the companion article and at:
  www.net-tech.bbn.com/projects/s-bgp
- *Domain Name System* (DNS)-based *Network Layer Reachability Information* (NLRI) origin *Autonomous System* (AS) verification in BGP, which is the oldest attempt at validating the information carried within BGP, is described in **draft-bates-bgp4-nlri-orig-verif-00.html**,

This article discusses *Secure Origin BGP* (soBGP), a solution recently proposed by a group (including me) mostly within Cisco Systems. We believe soBGP to be a deployable mechanism for validating the correctness and authorization of the data carried within BGP, and also for preventing the sorts of attacks resulting from misconfiguration or intentional insertion of bad data into the Internet routing system.

We address four goals when we consider security in terms of BGP:

- Is the AS originating the destination (prefix) authorized to advertise it? In other words, if a router receives an advertisement for the 10.1.1.0/24 network originating in AS65500, is there any way to verify that AS65500 is supposed to be advertising 10.1.1.0/24?
- Does the AS advertising the destination actually have a path to the destination? In other words, if a router is receiving an advertisement from a BGP peer in AS65501 that it can reach 10.1.1.0/24, is there any way to verify that AS65501 actually has a path to the AS origination 10.1.1.0/24?

- Is the peer advertising the route authorized by the originator, or owner, of the destination, to advertise a path to the destination?
- Does the path advertised by a peer AS fall within the policies the local network administrators have set forward? The most obvious issue is whether or not the AS Path advertised by the peer is an acceptable path to send the traffic along.

We argue elsewhere that the second two goals cannot be fully met within an operational internetwork, for many reasons; see **`draft-white-pathconsiderations-00.txt`** for further discussion on this point. In this article, then, we discuss how soBGP can meet the first two goals in operational networks.

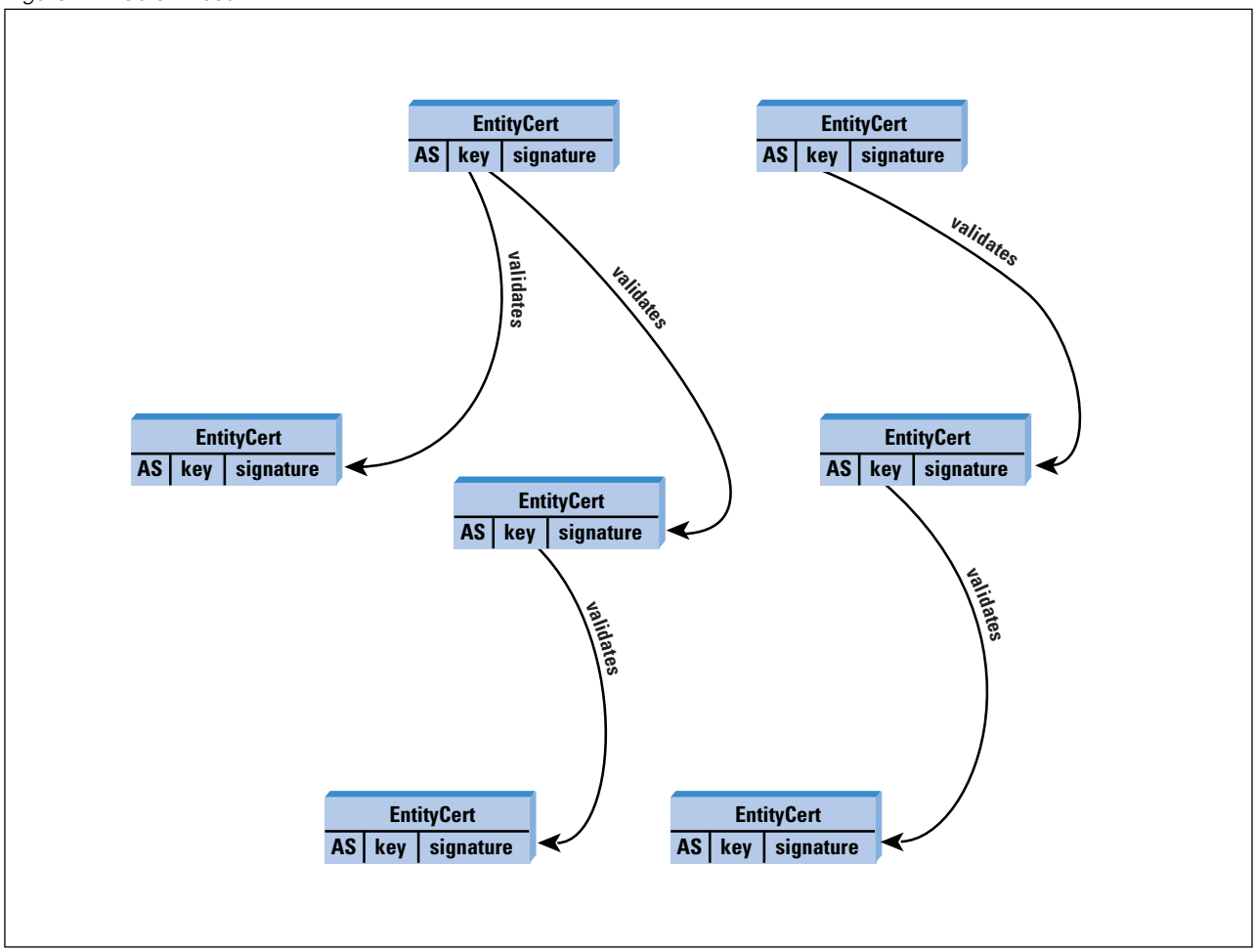
Begin at the Beginning: Who Are You?

The first step in securing anything is authentication; each participant must have some way of knowing who the other participants are, and what information they will be using to sign or encrypt their data. This is a classic problem in cryptography, called *key distribution*. There must be some way to receive keys used to sign or encrypt data, and then to validate that the keys received actually belong to the participant we believe they belong to.

This problem is addressed in soBGP using an *EntityCert,* which ties an AS number to a public key (or a set of public keys) corresponding to a private key the AS will be using to sign various other certificates. An EntityCert is defined in soBGP to be an X.509v3 certificate, similar to those used by *Transport Layer Security* (TLS) and *IP Security* (IPSec). The main problem we face when accepting an EntityCert is knowing whether or not the key carried within the certificate is actually the key of the advertising AS.

soBGP resolves this by requiring the EntityCert to be signed by a third party, validating that this AS actually belongs with this key. A small number of "root keys" distributed out of band could then be used to validate a set of advertised EntityCerts. These are used in turn to build up the database of known good ASm/key pairs in the system, allowing even more EntityCerts to be validated. Thus, EntityCerts can form a web of trust, built on the public keys of a small number of well-known entities, such as top-level backbone service providers, key authentication service providers (such as Verisign), and others.

Figure 1: Web of Trust

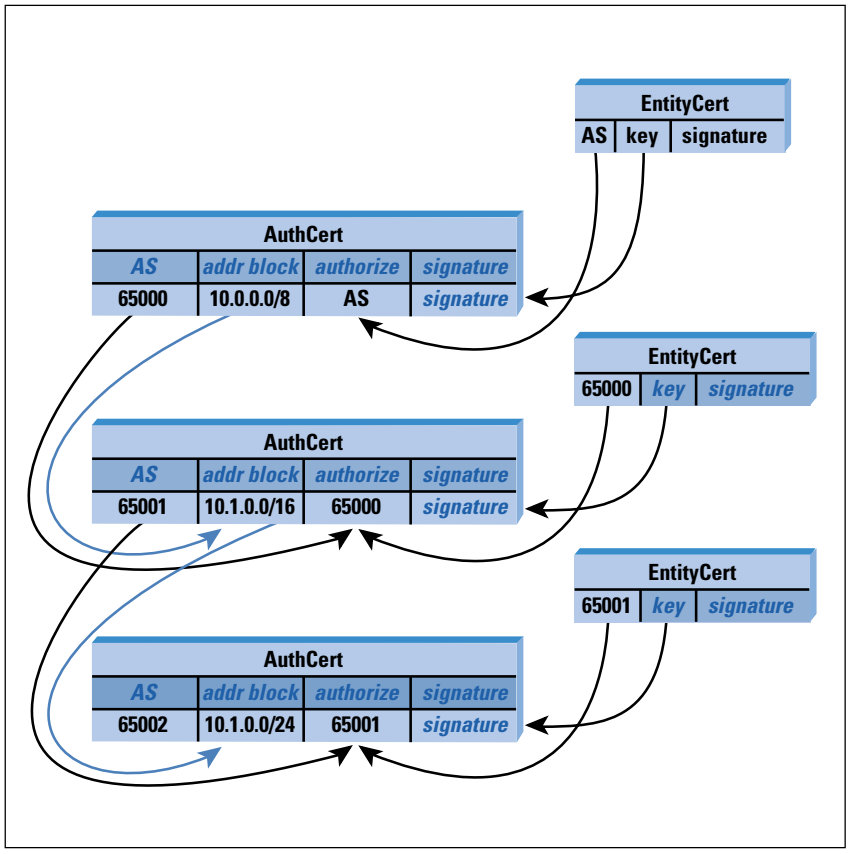


The key each AS distributes in its EntityCert is actually the public half of a private/public key pair. An AS would keep its private key entirely private, holding it on one highly secure device in its network (which is not even required to be online), and generating signatures for other certificates as needed. Only an AS public key is ever exposed in this way, so no special protection mechanisms (for example, tamper-resistant hardware) are required at any border to prevent private keys from being compromised.

The First Goal: Are You Authorized?

Now that we have distributed a public key per AS, we can build a certificate that will provide authorization for an AS to advertise a specific block of addresses. This authorization is provided through an *Authorization Certificate*, or *AuthCert*. An AuthCert ties an AS to a block of addresses that the AS may advertise, as Figure 2 illustrates.

Figure 2: Authorization Example



Starting at the top of the illustration, we find that some AS has authorized AS65000 to advertise prefixes within the block 10.0.0.0/8. The AuthCert is signed using the authorizing AS key. To delegate some part of this block of address space to another AS, AS65001, AS65000 builds an AuthCert tying 10.1.0.0/16 to AS65001. AS65001, in turn, suballocates a smaller part of this address space to AS65002, by building an AuthCert tying AS65002 to 10.1.1.0/24.

Any device receiving these three AuthCerts can check them by:

- Looking up the public key of the authorizer, and verifying the signature on the AuthCert
- Making certain the authorizer is permitted to advertise the address space it has suballocated this block of address space from

The device then builds a local table of address blocks and corresponding ASs authorized to advertise prefixes within those address blocks. Received updates can be checked against this database to verify authorization of the originating AS to advertise a prefix.

Blocks of address space are used here, rather than individual prefixes; an AuthCert can authorize an AS to advertise any number of prefixes within a block of addresses. This reduces the number of certificates within the system, thereby reducing overall cryptographic processing requirements. If a specific AS desires per-prefix authorization, it can build individual AuthCerts for each allocated prefix, rather than for blocks of address space.

Per-Prefix Policy

AuthCerts are not advertised as independent certificates within soBGP; instead, they are wrapped in a *PrefixPolicyCert*. PrefixPolicyCerts contain an AuthCert, a set of policies the originator would like to apply to prefixes advertised within this block of addresses, and a signature generated using the private key of the authorized AS. Policies that may be included in the PrefixPolicyCert include the longest prefix length allowed within the address block, and possibly other policies, such as a list of ASs that may not be or must be in the AS Path of routes to destinations within the address block.

In reality, the per-prefix policies available to the originator are limitless; the main problem is enforcing those policies when they are received by other ASs.

The Second Goal: Do You Really Have a Path?

Our second goal is to be able to verify that the advertiser of a given route actually has a path to the destination. This goal is met in soBGP by building a topology map of the paths of the entire internetwork. Each AS attached to the internetwork builds an *ASPolicyCert*, which contains, primarily, a list of its peers, and signed using the originator's private key. Using this list of transit peers, a map of the internetwork topology may be built, as Figure 3 illustrates.

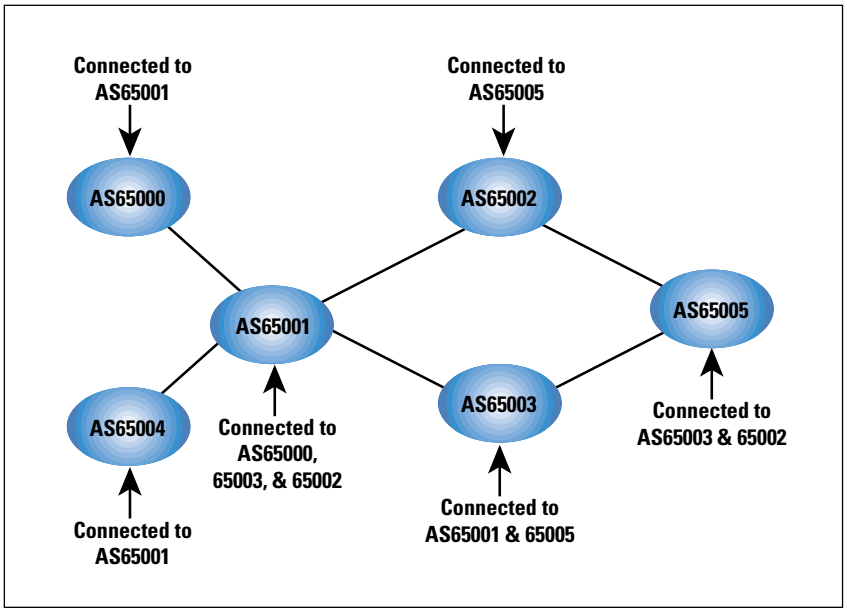


Figure 3: Connectivity Graph Example

If AS65005 receives an update from AS65002, claiming it can reach a destination in AS65000 through the path {65002, 65001, 65000}, it can:

- Check to make certain AS65002 claims to be connected to AS65001 in its ASPolicyCert, and that AS65001 claims to be connected to AS65002 in its ASPolicyCert
- Check to make certain AS65001 claims to be connected to AS65000 in its ASPolicyCert, and that AS65000 claims to be connected to AS65001 in its ASPolicyCert

If, for instance, AS65002 claims a path to a destination inside AS65000 through the path (65002, 65000), AS65002 would be able to discover that the path is invalid, because AS65000 does not claim to be connected to AS65002. This simple two-way connectivity check along a graph can be mixed with various policy statements—stating a specific peer is not a transit, not advertising certain peers, etc.—to provide a much wider range of policies than AS Path-based methods.

Transporting Certificates

One of the primary problems any security system such as soBGP is going to face is transporting security information through the internetwork. We would like to make certain we do not rely on the routing system to provide information about the security of the routing system. In other words, we would not like to rely on unsecured routing information in order to reach a server providing the information required to secure the path to the server itself.

soBGP resolves this by proposing to advertise certificates in much the same way as routing information is propagated today—through an interdomain protocol. Currently the soBGP drafts specify a new type of BGP message, the SECURITY message, which can be used to transport the required certificates, the EntityCert, the PrefixPolicyCert, and the ASPolicyCert, throughout an internetwork. Other methods of transporting data such as these certificates throughout an internetwork are currently being pursued by the IETF; if other methods are offered, soBGP could transport certificates across any such distribution mechanism.

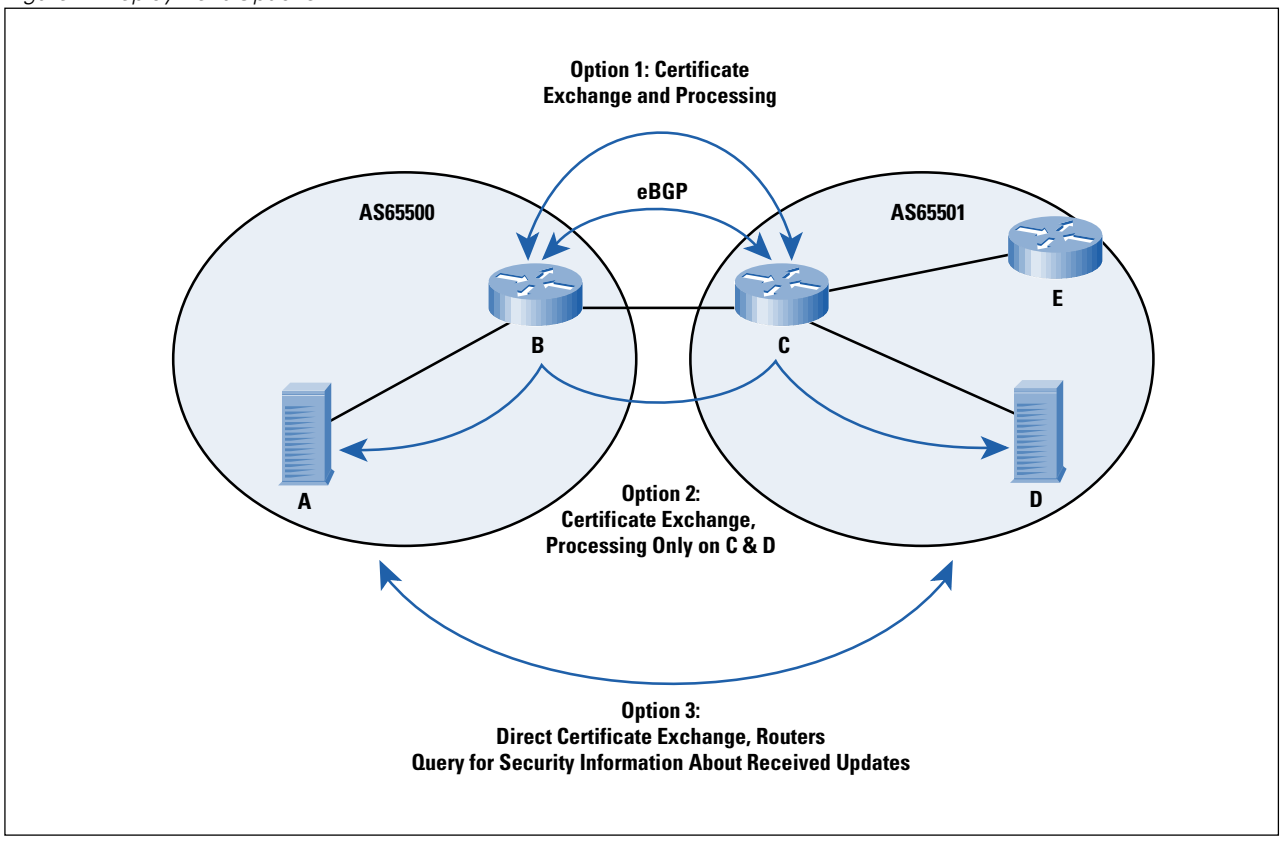
Deployment

Finally, we come to the hardest problem any routing security system is going to face: actually getting it operating in the field, with useful results, with a minimum of equipment changes, and a minimum number of participants. Here, soBGP provides a wide variety of options, primarily because it is not transport-dependent, nor dependent on a yet-to-be constructed centralized set of servers.

Although deployment options abound, here we discuss three, just to show the range of options available. Figure 4 illustrates these options.

The first option shown in this network is direct certificate exchange and processing between border routers. With this option, routers that are capable of the cryptographic processing required to validate received certificates exchange certificates with their peers in other ASs (just as they exchange routing information today), process those certificates, and build local databases from which they perform security checks on received updates.

Figure 4: Deployment Options



Although it may appear that processing, in this situation, would be extensive, it is actually possible to spread the processing required out among the border routers in a large AS. For instance, each certificate that router C receives and processes can be subsequently sent over an encrypted link to Router E. Router E could treat these certificates as though they had been validated locally, because they are received across an encrypted link from a trusted peer within the same administrative domain. Thus, only the edge router that has learned a certificate would actually process the certificate. This spreads the processing along all the edges in the AS.

A second option is for the edge routers, B and C, to exchange the certificates, but not process them. Instead, each edge router would relay the not-yet-validated certificates to internal servers A and D, respectively, thereby validating the certificates by performing the necessary cryptographic operations. As the border routers receive updates, they can query the server about the validity of each update, and take action based on the reply received.

Finally, it is possible for the servers to exchange certificates directly, over a multihop session. Servers A and D would then process the certificates, and the border routers, B and C, would query these servers to determine if received updates are valid or invalid.

Summary

Through this short survey of soBGP, we have shown it to be a flexible, moderately lightweight, yet strong system for validating the information carried through BGP in a large internetwork. It has low overhead processing requirements and very flexible deployment options, but no reliance on centralized servers. We are currently working to develop prototypes of soBGP on several platforms, to show how the technology will work on a wide range of devices.

For more information on soBGP, refer to:
`ftp://ftp-eng.cisco.com/sobgp/index.html`

You will find the most recent versions of the drafts, several slide shows, and other information about soBGP at this site.

RUSS WHITE works for Cisco Systems in the Routing Protocols Deployment and Architecture (DNA) team in Research Triangle Park, North Carolina. He has worked in the Cisco Technical Assistance Center (TAC) and Escalation Team in the past, has coauthored several books on routing protocols, including *Advanced IP Network Design, IS–IS for IP Networks,* and *Inside Cisco IOS® Software Architecture.* He is currently in the process of publishing a book on BGP deployment, and is the cochair of the Routing Protocols Security Working Group within the IETF. E-mail: **`riw@cisco.com`**

Trends in Viruses and Worms

by Thomas M. Chen, Southern Methodist University

The modern computer *virus* was conceived and demonstrated by Fred Cohen in 1983. Like biological viruses, computer viruses reproduce by attaching to a normal program or document and taking over control of the execution of that program to infect other programs. Early viruses could spread slowly mostly by floppies (such as the 1986 *Brain* virus), but the Internet has made it much easier for viruses to move among computers and spread rapidly. Networks have created a fertile environment for worms, which are related to viruses in their ability to self-replicate but are not attached to other programs. Worms are particularly worrisome as standalone automated programs designed to exploit the network to seek out vulnerable computers. The term *worm* was originated by John Shoch and Jon Hupp during their experiments on mobile software at Xerox PARC in 1979, inspired by the network-based *tapeworm* monster in John Brunner's novel, *The Shockwave Rider*[1]. Shoch and Hupp thought of worms as multi-segmented programs distributed across networked computers.

The Internet increases the vulnerability of all interconnected machines by making it easier for malicious programs to travel between computers by themselves. Recent virus and worm outbreaks, such as the *Blaster* worm in August 2003 and the SQL *Sapphire/Slammer* worm in January 2003, have demonstrated that networked computers continue to be vulnerable to new attacks despite the widespread deployment of antivirus software and firewalls. Indeed, a review of the history of viruses and worms shows that they have continually grown in sophistication over the years. This article highlights a series of significant past innovations in virus and worm technology. The purpose is to show that viruses and worms continue to pose a major risk today and most likely into the future as their creators persist in seeking ways to exploit security weaknesses in networked systems.

Stealth

The earliest viruses attempted to hide evidence of their presence, a trend that continues to today. The 1986 DOS-based *Brain* virus hid itself in memory by simulating all of the DOS system calls that normally detect viruses, causing them to return information that gave the appearance that the virus was not there.

The 2001 *Lion* worm installed a rootkit called *t0rn,* which is designed to make the actions of the worm harder to detect through numerous system modifications to deceive *syslogd* from properly capturing system events (*syslogd* is often used to detect worm activity)[2]. More recently, viruses and worms have attempted to hide by actively attacking antivirus software on the infected computer (refer to the section "Armoring").

Social Engineering

The 1987 *Christma Exec* virus was an early example of social engineering, spreading by e-mail among IBM mainframes. An arriving message tricks the user into executing the virus by promising to draw a Christmas tree graphic. The virus does produce a Christmas card graphic on the computer display (drawn using a scripting language called *Rexx*) but sends a copy of itself in the user's name to that user's list of outgoing mail recipients. The recipients believe the e-mail is from the user, so they are more likely to open the e-mail.

Social engineering continues to be common practice in today's viruses and worms, particularly those spread by e-mail. In January 1999, the *Happy99/Ska* worm/Trojan horse hybrid spread by e-mail with an attachment called **`Happy99.exe`**[3]. When the attachment was executed, it displayed fireworks on the screen to commemorate New Year's Day, but secretly modified the **`WSOCK32.DLL`** file (the main Windows file for Internet communications) with a Trojan horse program that allowed the worm to insert itself into the Internet communications process. Every e-mail sent by the user generated a second copy without any text but carried the worm to the same recipients.

The 1999 *PrettyPark* worm propagated as an e-mail attachment called **`Pretty Park.exe`**. The attachment is not explained, but it bears the icon of a character from the television show, *South Park*. If executed, it installs itself into the Windows System folder and modifies the Registry to ensure that it runs whenever any **`.EXE`** program is executed. In addition, the worm e-mails itself to addresses found in the Windows Address Book. It also mails some private system data and passwords to certain *Internet Relay Chat* (IRC) servers. Reportedly, the worm also installs a backdoor to allow a remote machine to create and remove directories, and send, receive, and execute files.

In February 2001, the *Anna Kournikova* virus demonstrated social engineering again, pretending to carry a JPG picture of the tennis player. If executed, the virus e-mails a copy of itself to all addresses in the Outlook address book.

In March 2002, the *Gibe* worm spread as an attachment in an e-mail disguised as a Microsoft security bulletin and patch. The text claimed that the attachment was a Microsoft security patch for Outlook and Internet Explorer. If the attachment is executed, it displays dialog boxes that appear to be patching the system, but a backdoor is secretly installed on the system.

Macro Viruses

The *Concept* virus was the first macro virus, written for Word for Windows 95. The vast majority of macro viruses are targeted to Microsoft Office documents that save macro code within the body of documents. Macro viruses have the advantages of being easy to write and independent of computing platform. However, macro viruses are no longer widespread after people have become more cautious about using the Office macro feature.

Mass E-Mailers

In March 1999, the *Melissa* macro virus spread quickly to 100,000 hosts around the world in three days, setting a new record and shutting down e-mail for many organizations using Microsoft Exchange Server[4]. It began as a newsgroup posting promising account names and passwords for erotic Web sites. However, the downloaded Word document actually contained a macro that used the functions of Microsoft Word and the Microsoft Outlook e-mail program to propagate. Up to that time, it was widely believed that a computer could not become infected with a virus just by opening e-mail. When the macro is executed in Word, it first checks whether the installed version of Word is infectable. If it is, it reduces the security setting on Word to prevent it from displaying any warnings about macro content. Next, the virus looks for a certain Registry key containing the word "Kwyjibo" (apparently from an episode of the television show, *The Simpsons*). In the absence of this key, the virus launches Outlook and sends itself to 50 recipients found in the address book. Additionally, it infects the Word **`NORMAL.DOT`** template using the Microsoft *Visual Basic for Applications* (VBA) macro auto-execute feature. Any Word document saved from the template would carry the virus.

In June 1999, the *ExploreZip* worm appeared to be a WinZip file attached to e-mail but was not really a zipped file[5]. If executed, it appears to display an error message, but the worm secretly copies itself into the Windows Systems directory or loads itself into the Registry. It sends itself via e-mail using Outlook or Exchange to recipients found in unread messages in the inbox. It monitors all incoming messages and replies to the sender with a copy of itself.

In May 2000, the fast-spreading *Love Letter* worm demonstrated a social engineering attack[6]. It propagated as an e-mail message with the subject "I love you" and text that encourages the recipient to read the attachment. The attachment is a Visual Basic script that could be executed with Windows Script Host (present if the computer has Windows 98, Windows 2000, Internet Explorer 5, or Outlook 5). Upon execution, the worm installs copies of itself into the Windows System directory and modifies the Registry to ensure that the files are run when the computer starts up. The worm also infects various types of files (for example, **`.VBS`**, **`.JPG`**, **`.MP3`**, etc.) on local drives and networked shared directories. If Outlook is installed, the worm e-mails copies of itself to addresses found in the address book. In addition, the worm makes a connection to IRC and sends a copy of itself to anyone who joins the IRC channel. The worm has a password-stealing feature that changes the startup URL in Internet Explorer to a Website in Asia. The Website downloads a Trojan horse designed to collect various passwords from the computer.

In 2002, 90 percent of the known viruses were mass e-mailers. Two of the most prevalent ones, *Bugbear* and *Klez,* began a trend of carrying their own *Simple Mail Transfer Protocol* (SMTP) engines. Although e-mail continues to be the most common infection vector, recent worms have been exploring new vectors (see the section "New Infection Vectors").

In addition, mail servers are becoming more powerful in their capabilities to detect and filter malicious code. For these reasons, mass e-mailing may decline as an infection vector for future viruses.

Polymorphism

Polymorphism is based on the simpler idea of encryption, which makes a virus harder to detect by antivirus software scanning for a unique virus signature (byte pattern). Encryption attempts to hide a recognizable signature by scrambling the virus body. To be executable, the encrypted virus is prepended with a decryption routine and encryption key. However, encryption is not effective because the decryption routine remains the same from generation to generation, although the key can change, scrambling the virus body differently. Antivirus scanners can detect a sequence of bytes identifying a specific decryption scheme.

Polymorphic viruses permute continuously to avoid detection by antivirus scanning[7]. The earliest polymorphic virus might have been a virus found in Europe in 1989. This virus replicated by inserting a pseudorandom number of extra bytes into the decryption algorithm, preventing any common sequence of more than a few bytes between two successive infections. Polymorphism became practical when a well-known hacker, *Dark Avenger,* developed a user-friendly *Mutation Engine* program to provide any virus with variable encryption. With a static signature so small, the risk of false positives by antivirus scanners became very high. Other hackers soon followed with their own versions of so-called mutation engines. The 1995 *Pathogen* and *Queeg* viruses were polymorphic DOS file-infecting viruses produced by Black Baron's *Simulated Metamorphic Encryption enGine* (SMEG)[7].

Blended Attacks

The famous 1988 *Morris* worm was the first to use a combination of attacks (or blended attacks) to spread quickly to 6000 UNIX computers in a few hours (10 percent of the Internet at that time)[8].

- It captured the password file and ran a password-guessing program on it using a dictionary of common words.
- It exploited the debug option in the UNIX *sendmail* program, allowing it to transfer a copy of itself.
- It carried out a buffer overflow attack through a vulnerability in the UNIX *fingerd* program.

In May 2001, the *Sadmind/IIS* worm spread by targeting two separate vulnerabilities on two different operating systems. It first exploited a buffer overflow vulnerability in Sun Solaris systems and installed software to carry out an attack to compromise Microsoft *Internet Information Services* (IIS) Web servers.

The July 2001 *Sircam* worm uses two ways to propagate. First, it e-mails itself as an attachment using its own SMTP engine, and if the attachment is executed, e-mails a copy of itself to addresses found in the Windows address book. Second, it spreads by infection of unprotected network shares.

In September 2001, *Nimda* raised new alarms by using five different ways to spread to 450,000 hosts within the first 12 hours[9]. *Nimda* seemed to signal a new level of worm sophistication.

- It found e-mail addresses from the computer Web cache and default *Messaging Application Programming Interface* (MAPI) mailbox. It sent itself by e-mail with random subjects and an attachment named **`readme.exe`**. If the target system supported the automatic execution of embedded MIME types, the attached worm would be automatically executed and infect the target.
- It infected Microsoft IIS Web servers, selected at random, through a buffer overflow attack called a *unicode* Web traversal exploit.
- It copied itself across open network shares. On an infected server, the worm wrote *Multipurpose Internet Mail Extensions* (MIME)-encoded copies of itself to every directory, including network shares.
- It added JavaScript to Web pages to infect any Web browsers going to that Website.
- It looked for backdoors left by previous *Code Red II* and *Sadmind* worms.

Armoring

In November 2002, the *Winevar* worm was an example of an "armored" worm that contained special code designed to disable antivirus software using a list of keywords to scan memory to recognize and stop antivirus processes and scan hard drives to delete associated files[10].

Klez and *Bugbear* are recent examples of worms that attack antivirus software by stopping active processes and deleting registry keys and database files used by popular antivirus programs. The 2003 *Fizzer* and *Lirva* worms also attempt to disable antivirus software.

Dynamic Software Updates

In October 2000, the *Hybris* worm propagated as an e-mail attachment[11]. It connected to the **`alt.comp.virus`** newsgroup to receive encrypted plug-ins (code updates). The method is sophisticated and potentially very dangerous, because the worm payload (destructive capability) can be modified dynamically.

The 2003 *Lirva* worm attempted to connect to a Website on **`web.host.kz`** to download BackOrifice, a notorious remote-access software package that gives complete control to a remote attacker. It also attempted to download another unknown file that was not found on the Website.

This technique was given an interesting twist by the *Welchia* or *Nachi* worm, which began spreading on August 18, 2003, soon after the *Blaster* worm. Apparently, its creator intended *Welchia* as a "good" worm to remove *Blaster*. It attempted to download and install a fix for *Blaster* from a Microsoft Website.

New Infection Vectors

The Linux *Slapper* worm, appearing in September 2002, was among the first to exploit *peer-to-peer* (P2P) technology[12]. It spread to Linux computers by exploiting the long *Secure Sockets Layer 2* (SSL2) key argument buffer overflow in the *libssl* library, used by the *mod_ssl* module of the Apache 1.3 Web server. When the worm infects a new machine, it binds to *User Datagram Protocol* (UDP) port 2002 and becomes part of a P2P network. The parent of the worm on the attacking machine sends to its offspring the list of all hosts on the P2P network and broadcasts the address of the new worm on the network. Then periodic updates to the host list are exchanged between machines on the network. The new worm also scans the network for other vulnerable machines, sweeping randomly chosen class B networks.

In March 2003, the *AimVen* worm spread by the *America OnLine Instant Messager* (AIM) by modifying the AIM program. Whenever an **`.EXE`** file is sent through AIM, the worm overwrites the file with a copy of itself.

The *Fizzer* worm discovered in May 2003 is a mass e-mailer that includes its own SMTP engine like *Klez* and *Bugbear*. It also tries to spread via *KaZaa*, a popular P2P file-sharing application, and shared directories.

The 2003 *Lirva* worm, named after the singer, Avril Lavigne, is a mass e-mailer taking advantage of the same MIME header exploit as *Badtrans* and *Klez*, but also tries to spread by IRC, "I seek You" (ICQ), *KaZaa,* and open network shares[13].

Data-Stealing Payloads

Most fast-spreading worms in the past have not carried destructive payloads. Instead, they have tended to appear to be proof-of-concepts to demonstrate a particular security weakness. Some worms, though, such as *Code Red,* have installed *Denial-of-Service* (DoS) agents or backdoors on infected machines. Recently worms have begun to carry keyloggers and password-stealing Trojans in their payloads.

The 2003 *Fizzer* worm includes a keystroke logging Trojan horse that stores the data in an encrypted file. It establishes its own accounts on IRC and AIM to wait for instructions from the virus writer, who could conceivably fetch the keystrokes data.

The 2003 *Lirva* worm e-mails cached Windows dialup networking passwords to the virus writer, and e-mail random **`.TXT`** and **`.DOC`** files to various addresses.

Bugbear installs a keystroke logging tool into the Windows System folder that e-mails the keystrokes data to preprogrammed addresses[14]. It listens on port 36794 for commands from a remote hacker.

Fast and Furious Worms

A particularly worrisome new trend is extremely fast worms targeted to specific (usually Windows-related) vulnerabilities that might saturate their target population within a few hours or even less than an hour. These worms tend to be simpler and targeted to single rather than multiple vulnerabilities, in order to be highly efficient in their probing for other vulnerable machines.

The first example might be the *Code Red* worm, which actually appeared in three different versions[15]. The first version of *Code Red I* appeared on July 12, 2001, targeted to a buffer overflow vulnerability in Microsoft IIS Web servers. However, a programming error in its pseudorandom address generator caused each worm copy to probe the same set of IP addresses and prevented the worm from spreading quickly. A week later on July 19, a second version of *Code Red I* with the programming error apparently fixed was able to infect more than 359,000 servers within 14 hours. At its peak, the worm was infecting 2000 hosts every minute. A more complex and dangerous *Code Red II* targeted to the same IIS vulnerability appeared on August 4.

More recently, the *Structured Query Language* (SQL) *Sapphire/ Slammer* worm appeared on January 25, 2003, targeted to Microsoft SQL Server machines not running *Service Pack 3* (SP3), such as SQL Server 2000 and *Microsoft Desktop Engine* (MSDE) 2000[16]. It reportedly infected 90 percent of vulnerable hosts within 10 minutes (about 120,000 servers)[17]. The spreading rate was surprisingly fast and resulted in DoS effects (network outages and high packet loss) due to traffic overloading servers and routers. In the first minute, the infection doubled every 8.5 seconds, and hit a peak scanning rate of 55,000,000 scans per second after only 3 minutes. In comparison, *Code Red* infection doubled in 37 minutes (slower but infected more machines). *Slammer* was able to spread so quickly because it appeared to be designed simply for efficient replication. The worm carried no payload and consisted of a single 404-byte UDP packet (including 376 bytes for the worm) that could be sent without having to wait for responses from targeted machines. In contrast, *Code Red* was about 4000 bytes and *Nimda* was 60,000 bytes, and their scanning depended on the time to establish TCP connections to targeted machines. The *Slammer* worm was much more efficient, simply generating copies of itself at the full rate of the infected machine.

Latest Developments

The week of August 12–19, 2003, has been called the worst week for worms in history, seeing *MS Blaster, Welchia* (or *Nachi*), and *Sobig.F* in quick succession. *MS Blaster* or *LovSan* was another fast worm, which appeared on August 12, 2003, targeted to a *Windows Distributed Component Object Model* (DCOM) *Remote Procedure Call* (RPC) vulnerability announced on July 16, 2003[18]. The worm probes for a DCOM interface with RPC listening on TCP port 135 on Windows XP and Windows 2000 PCs. Through a buffer overflow attack, the worm causes the target machine to start a remote shell on port 4444 and send a notification to the attacking machine on UDP port 69.

A *Trivial File Transfer Protocol* (TFTP) "get" command is then sent to port 4444, causing the target machine to fetch a copy of the worm as the file **`MSBLAST.EXE`**. In addition to a message against Microsoft, the worm payload carries a DoS agent (using TCP SYN flood) targeted to the Microsoft Website **`windowsupdate.com`** on August 16, 2003. Although *Blaster* has reportedly infected about 400,000 systems, experts reported that the worm did not achieve near its potential spreading rate because of novice programming.

Six days later on August 18, 2003, the apparently well-intended *Welchia* or *Nachi* worm spread by exploiting the same RPC DCOM vulnerability as *Blaster*. It attempted to remove *Blaster* from infected computers and download a security patch from a Microsoft Website to repair the RPC DCOM vulnerability. Unfortunately, its scanning resulted in a DoS effect on some networks, such as Air Canada's check-in system and the U.S. Navy and Marine Corps computers.

The very fast *Sobig.F* worm appeared on the next day, August 19, 2003, only seven days after *Blaster*[19]. The original *Sobig.A* version was discovered in January 2003, and apparently underwent a series of revisions until the most successful *Sobig.F* variant. Similar to earlier variants, *Sobig.F* spreads among Windows machines by e-mail with various subject lines and attachment names, using its own SMTP engine. The worm size is about 73 kilobytes with a few bytes of garbage attached to the end to evade antivirus scanners. It works well because it grabs e-mail addresses from a variety of different types of files on the infected computer and secretly e-mails itself to all of them, pretending to be sent from one of the addresses. At its peak, *Sobig.F* accounted for 1 in every 17 messages, and reportedly produced over 1 million copies of itself within the first 24 hours. Interestingly, the worm was programmed to stop spreading on September 10, 2003, suggesting that the worm was intended as a proof-of-concept. This is supported by the absence of a destructive payload, although the worm is programmed with the capability to download and execute arbitrary files to infected computers. The downloading is triggered on specific times and weekdays, which are obtained via one of several *Network Time Protocol* (NTP) servers. The worm sends a UDP probe to port 8998 on one of several preprogrammed servers, which responds with a URL for the worm to download. The worm also starts to listen on UDP ports 995–999 for incoming messages, presumably instructions from the creator.

Conclusions

Why does the Internet remain vulnerable to large-scale worm outbreaks? Since at least 1983, the Internet community has understood the risks and mechanics of viruses. The 1988 Morris worm taught the community to be watchful for potentially dangerous worms. Over the years, a variety of antivirus software, firewalls, intrusion detection systems, and other security equipment have been installed. Moreover, the *Computer Emergency Response Team* (CERT) at CMU was established as the first computer security incident response team, which later joined an expansive global coalition of security incident response teams called the *Forum of Incident Response and Security Teams* (FIRST)[20].

Despite our knowledge and infrastructure defenses, many viruses and worms have broken out regularly in the Internet over the years. By some reports, 5 to 15 new viruses and worms are released every day, although a fraction of that number are not released in the wild and most do not spread well. Still, fast-spreading viruses and worms continue to appear with regularity. Outbreaks have become so commonplace that most organizations have come to view them as a routine cost of operation.

The problem is sometimes portrayed as a perpetual struggle between virus writers who keep innovating (as described here) and the antivirus industry, which tries to keep up. However, the problem is actually larger, involving the entire computer industry. Viruses and worms are successful because computers have security vulnerabilities that can be exploited. Clearly, the Internet itself is simply serving its purpose of interconnecting computer systems. The security vulnerabilities exist in the host end systems. Security vulnerabilities continue to exist for many reasons. First, software is often written in an unsecure manner, for example, vulnerable to buffer overflow attacks that are commonly used by worms. Buffer overflow attacks have been widely known since 1995, but this type of vulnerability continues to be found very often (on every operating system.) Second, when vulnerabilities are announced with corresponding software patches, many people are slow to apply patches to their computer for various practical reasons. Weakly protected computers can be compromised, putting the entire community at risk, including secured computers that can still be impacted by the traffic effects of a worm outbreak.

However, there is reason to be hopeful for a solution. Fortunately, worms typically have a weakness of exploiting vulnerabilities that have been known for some time. Worm writers do not invent new exploits for the simple reason that they want to ensure that their worm will spread after it is released. For example, the *Code Red I* worm took advantage of a buffer overflow vulnerability in Microsoft IIS servers that had been known for a month. The *Nimda* worm exploited a unicode Web traversal vulnerability in Microsoft IIS servers that was published a year earlier. The SQL *Slammer/Sapphire* worm exploited a buffer overflow vulnerability in Microsoft SQL servers that had been known for six months. The recent *Blaster* worm exploited a Windows DCOM RPC vulnerability announced two months earlier. Watching for probing activity attempting to exploit known vulnerabilities could help detect and block worm outbreaks at an early stage. Ideas for automatic detection and quarantine of new epidemics is attracting research[21].

Aside from technological considerations, an important issue is accountability. The most obvious parties to hold liable are the virus creators, but it has been observed many times that few virus writers have been prosecuted, and sentences have tended to be light. The author of the 1988 Internet worm, Robert Morris, was sentenced to three years of probation, 400 hours of community service, and a $10,000 fine.

Chen Ing-hau was arrested in Taiwan for the 1998 *Chernobyl* virus, but he was released when no official complaint was filed. Onel de Guzman was arrested for writing the 2000 *LoveLetter* virus, which resulted in $7 billion of damages, but he was released because of the lack of relevant laws in the Philippines. Jan De Wit was sentenced for the 2001 *Anna Kournikova* virus to 150 hours of community service. David L. Smith, creator of the 1999 *Melissa* that caused at least $80 million of damages, was sentenced to 20 months of custodial service and a $7500 fine.

It is notoriously difficult to trace a virus or worm to its creator from analysis of the code, unless inadvertent clues are left in the code. In addition, cases are difficult to prosecute, and malicious intention (as opposed to just recklessness) is difficult to prove. Moreover, long prison sentences have been perceived as overly harsh for arrested virus creators, who have tended to be teenagers and university students. In addition, in the absence of a serious legal deterrent, the general perception persists that virus creators can easily avoid the legal consequences of their actions. Perhaps to address this problem, authorities have been diligently investigating the creators of *Blaster* and *Sobig*. So far, a teenager, Jeffrey Lee Parson, has been arrested for writing the *Blaster.B* variant, a slight modification of the original *Blaster*. Soon afterward, Dan Dumitru Ciobanu was arrested in Romania for writing the *Blaster.F* variant.

Some have argued wishfully that software vendors should be held financially liable for damages resulting from the security vulnerabilities in their products. The assumption is that accountability would increase motivation to write and sell more secure software, a solution that would result in a less inviting environment for viruses and worms. So far, software vendors have managed to acknowledge their role but avoid accountability.

References

[1] J. Shoch and J. Hupp, "The 'worm' programs—early experience with a distributed computation," *Communications of ACM,* Volume 25, pp. 172–180, March 1982.

[2] A. Kasarda, "The Lion worm: king of the jungle?" SANS reading room, **http://www.sans.org/rr**

[3] CERT incident note CA-1999-02, "Happy99.exe trojan horse," **http://www.cert.org/incident_notes/IN-99-02.html**

[4] CERT advisory CA-1999-04, "Melissa macro virus," **http://www.cert.org/advisories/CA-1999-04.html**

[5] CERT advisory CA-1999-06, "ExploreZip trojan horse program," **http://www.cert.org/advisories/CA-1999-06.html**

[6] CERT advisory CA-2000-04, "Love letter worm," **http://www.cert.org/advisories/CA-2000-04.html**

[7] D. Harley, R. Slade, and R. Gattiker, *Viruses Revealed,* Osborne/McGraw-Hill, 2001.

[8] E. Spafford, "The Internet worm program: an analysis," *ACM Computer Communications Review,* Volume 19, pp. 17–57, January 1989.

[9] CERT advisory CA-2001-26, "Nimda worm,"
http://www.cert.org/advisories/CA-2001-26.html

[10] Virus Bulletin, "W32/WineVar,"
http://www.virusbtn.com/resources/viruses/winevar.xml

[11] CERT incident note IN-2001-02, "Open mail relays used to deliver Hybris worm,"
http://www.cert.org/incident_notes/IN-2001-02.html

[12] F-Secure, "F-Secure virus descriptions: Slapper,"
http://www.f-secure.com/v-descs/slapper.shtml

[13] Symantec Security Response, "W32.lirva.C@mm,"
http://securityresponse.symantec.com/avcenter/venc/data/w32.lirva.c@mm .html

[14] Sophos, "W32/Bugbear-A,"
http://www.sophos.com/virusinfo/analyses/w32bugbeara.html

[15] H. Berghel, "The Code Red worm," *Communications of ACM,* Volume 44, pp. 15–19, December 2001.

[16] CERT advisory CA-2003-04, "MS-SQL server worm,"
http://www.cert.org/advisories/CA-2003-04.html

[17] D. Moore, et al., "The spread of the Sapphire/Slammer worm,"
http://www.caida.org/outreach/papers/2003/sapphire/sapphire.html

[18] CERT advisory CA-2003-20, "W32/Blaster worm," Aug. 11, 2003,
http://www.cert.org/advisories/CA-2003-20.html

[19] Symantec Security Response, "W32.Sobig.F@mm,"
http://securityresponse.symantec.com/avcenter/venc/data/w32.sobig.f@mm .html

[20] Forum of Incident Response and Security Teams (FIRST),
http://www.first.org

[21] D. Moore, C. Shannon, G. Voelker, and S. Savage, "Internet quarantine: requirements for containing self-propagating code," IEEE Infocom 2003, San Francisco, April 2003.

THOMAS M. CHEN holds BS and MS degrees in electrical engineering from MIT, and a PhD in electrical engineering from the University of California, Berkeley. From 1989 to 1997, he worked on ATM networking research at GTE Laboratories (now Verizon). He is currently an Associate Professor in the Department of Electrical Engineering at SMU in Dallas, Texas. He is the associate editor-in-chief of *IEEE Communications Magazine,* a senior editor of *IEEE Network,* an associate editor of *ACM Transactions on Internet Technology,* and founding editor of *IEEE Communications Surveys.* He is the coauthor of *ATM Switching Systems* (Artech House, 1995). E-mail: **tchen@engr.smu.edu**

IPv6 Behind the Wall

by Jim Bound

IPv6 has technology advantages over IPv4, and most of them will not be seen by the end user any more than users see features added to other extensions to the Internet Protocol suite, sensors on their automobiles, or from any core technology evolution. This article focuses on three of those IPv6 technology advantages "Behind the Wall."

An essential catalyst for the Next-Generation Internet is the *Internet Protocol Version 6* (IPv6), which will provide an evolution to a more pervasive use of the Internet and networking in general. The current Internet, using IPv4, is insufficient to support the business and operational preconditions for peer-to-peer applications and security, billions of mobile devices, sensor networks, and the requisite distributed computing infrastructure to support a mobile society. The "band aids" applied to permit the current Internet to keep it operating has created additional operational costs and reduced operational capabilities for users and networks.

This article is an IPv6 Forum (**`www.ipv6forum.com`**) statement of the technology advantages of IPv6.

IPv6 Supports End-to-End Applications and Security

There are several schools of thought and opinions on the issue of address space and all project different results, depending on one's mathematical view and philosophy regarding use models. There is also the effect of disruptive technology, which can make moot any projections of IPv4 address space. In that sense, rationing is justified and intelligent. The IPv6 Forum believes we already are experiencing the initial quake of disruptive technology, and that there is a need for users and markets to evolve further with a basic tenet that end-to-end applications and security are a priori for that evolution to begin. The IPv6 Forum believes that *Network Address Translatio*n (NAT) is about control, but that control comes at a cost of the freedom to use peer-to-peer computing over client to server-only computing.

Two users on the Internet today generally cannot each initiate peer-to-peer communications with each other because their location and identity are not available to each other from two disparate networks. In addition, security between them must trust a third party, and absolute private communications is impossible. The reason is that the Internet has evolved so that users are generally behind NATs that preclude peer-to-peer communications, or the exchange of private security credentials. Some will say this affords users security on the Internet. Although NAT does provide a denial-of-service perimeter, it also provides a denial of service to a direct trust relationship between peers. IPv6 is the only way to have peer-to-peer security for the Next-Generation Internet at a reasonable cost and a true privacy trust model on the Internet.

In the field of network computer science when engineers and architects implement translation functions in a solution, a cost is incurred that would not exist without translation. This is due to the need to keep *state* before, during, and after the translation. In software engineering terminology, these *state machines* add time and space costs to the entire operation. In addition, a NAT box is a single point of failure, because it is the only point on the network where a user can exit or enter when translation exists. Translation also does not permit the use of all functions possible without translation because too many participants need to know the mappings, and each function requires a separate state to be maintained, and the time + space costs increase exponentially. The time + space costs of NAT to keep the Internet operational have been passed on to every part of the current Internet business, consumer, and government market sectors, and cannot even support the original functions of the Internet before NAT. The current Internet has no hope of supporting the functions of the Next-Generation Internet required or of offering a solution to the great digital divide that exists currently and is increasing daily.

The good news is that IPv6 is evolving, early adopter deployment has begun, and vendors have delivered initial IPv6 products to the market. IPv6 will not require NAT, and the infrastructure supports a stateless architecture for the Internet, using statefull properties only where they can be used without a translation attribute or policy. IPv6 inherently supports mobile communications, billions of devices, and sensor networks that will be pervasive at a reasonable cost and provide the option to eliminate the digital divide within the current Internet.

IPv6 Supports a Stateless Node Discovery Architecture

A Next-Generation Internet base technology advantage for mobile user devices, ad hoc networks, mobile network providers, and generally for all users is the *Stateless Node Discovery Architecture* inherent within IPv6.

IPv6 nodes can discover each other and form IPv6 addresses to communicate on a network using what is called *Neighbor Discovery* and *Stateless Autoconfiguration*. IPv6 supports an extensible stateless node discovery paradigm, which provides the following features:

- Discover presence of nodes on the network
- Discover Datalink Layer nodes on the network
- Discover routers on the network
- Discover link configuration parameters on the network

These features permit an IPv6 node to obtain and maintain information about the accessibility of another node on the network for communications. Node Discovery is the predecessor to the node obtaining an address from IPv6 autoconfiguration. This core IPv6 technology framework also permits nodes to communicate on networks where there are no routers within an ad hoc network.

A host, when booted on an IPv6 link, first creates a *link-local* address by taking the architecturally defined prefix in Neighbor Discovery **`FE80`**, and appending an *End User Identifier* (EUI), determined by the host, to that prefix. This link-local address is then verified on the link that it is not duplicated with other link-local addresses on that host's link. This host communication is performed using link IPv6 multicast packets, to avoid duplicate link-local addresses, which are not permitted on an IPv6 Link.

The host then uses the link-local address to send on the IPv6 link *Neighbor Solicitations,* and all other hosts on that link see those multicast solicitations, and then return *Neighbor Advertisements* to the host. After this communications process, all nodes on the IPv6 link can now communicate, and communication was accomplished without the use of servers or routers in a stateless manner.

The host also listens for *Router Advertisements* on the IPv6 link (or sends *Router Solicitations*), which provide address prefixes, link configuration parameters, and information as to whether or not to use a stateless or stateful method for address assignment, and additional network configuration parameters using the *Dynamic Host Configuration Protocol for IPv6* (DHCPv6)[1].

If the host is instructed to use the stateless method for address configuration, then it can use the router prefixes announced to form IPv6 addresses from those prefixes by appending the EUI determined from the link-local address to that prefix to create an IPv6 Address. IPv6 supports multiple address types within the address architecture[2,3]. If the host is instructed to use the stateful method for address configuration, then DHCPv6 can be used to configure additional hosts' addresses.

Users will not see these IPv6 stateless advantages for network communications, but they will exist behind the wall of the user to provide a new and improved set of mechanisms for Node Discovery and Address Autoconfiguration far more robust and efficient than using the current IP Version 4 (IPv4) protocol. The IPv6 Stateless Architecture for Node Discovery permits a new model for node communications on links.

The Mobile IPv6 Technology Value Proposition

Mobile IPv6 offers many improvements over Mobile IPv4. Mobile IP as a technology permits users to remain connected across wireline (for example, Ethernet, xDSL) and wireless (for example, 802.11, cellular, satellite) networks, while roaming between networks. This permits users to stay connected while on the way to the airport from home, rather than shutting down their personal digital assistant (PDA)/laptop at home, and reconnecting at the WiFi location at the airport.

Figure 1: Route Optimization with Built-In Security

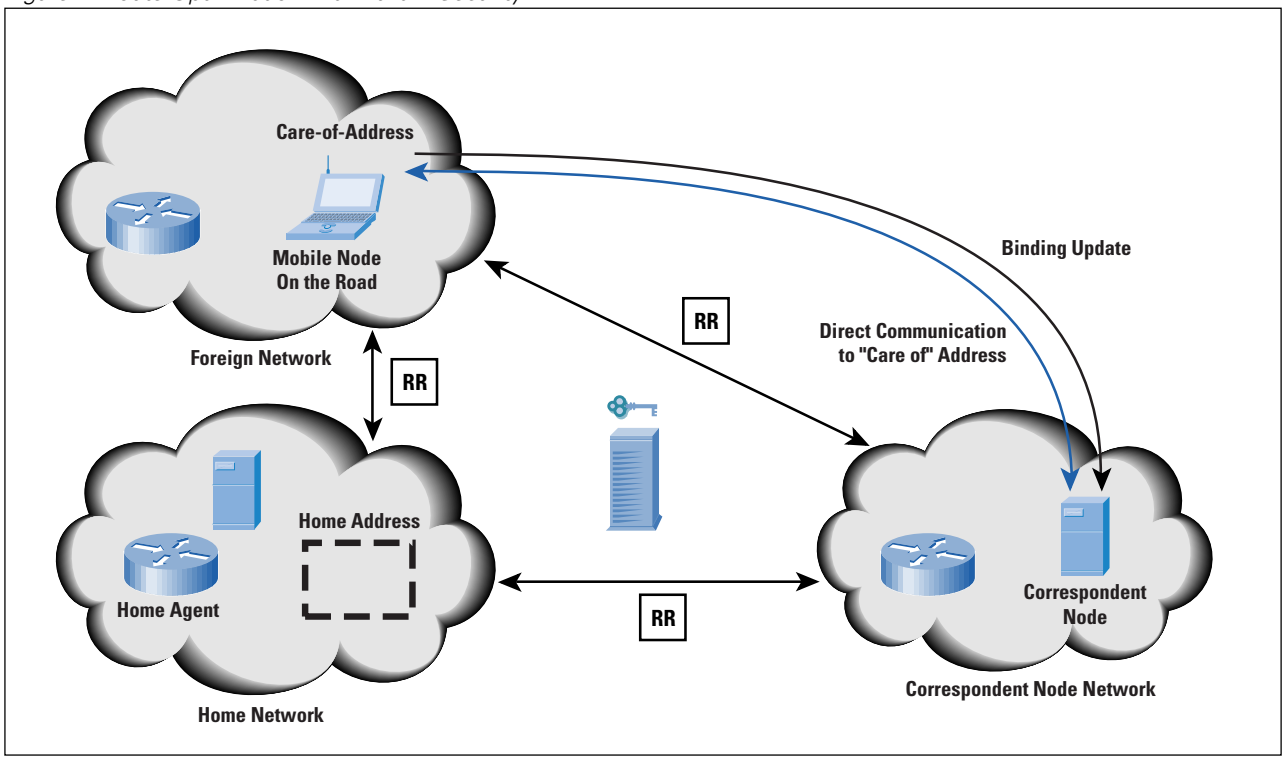


Figure 1 depicts the multiple phases of a mobile IPv6 connection. On the home network, a mobile node receives its home address as any IPv6 node. The mobile node registers that address with the *Home Agent*, which is a router that keeps the location information for the mobile node when it moves to a foreign network, stores the mobile-node *care-of address* when the mobile node is away from home, and performs other functions on behalf of the mobile node when it is away from home. A peer node that the mobile node communicates with is defined as the *Correspondent Node* (which may be stationary or mobile).

Security between the mobile node and home agent can be accomplished using the *IP Security Protocol* (IPSec) architecture. This permits secure communications between the mobile node and the home agent. When a correspondent node receives a packet from a mobile node, it first checks its binding caches to see if it has a cache of the mobile-node care-of address, and if it does not, the correspondent node sends the packet to the mobile-node home address. The home agent receives all packets sent to the mobile node when it is away from home and then tunnels the packets to the mobile-node care-of address.

To permit a mobile node and correspondent node to communicate directly, without going through a home agent, requires the use of *Mobile IPv6 Route Optimization*. First the connection to the correspondent node needs to be secure from the home agent and directly from the mobile node. In the figure, that is done using a procedure defined as *Return Routability* (RR) within the Mobile IPv6 protocol. The network path between the mobile node and correspondent node is secured through the RR procedure.

Mobile IPv6 uses the extensibility of the IPv6 protocol defining new Neighbor Discovery messages and types, *Routing Header,* and the use of the *Destination Option* in an IPv6 packet, which does not exist in IPv4. Discussion of those extensions is beyond the scope of this article, and is left as an exercise for readers to read the actual Mobile IPv6 specification.

Mobile IPv6 has core technical operational advantages over Mobile IPv4, as follows:

- There is no need to deploy special routers as "foreign agents," as in Mobile IPv4. Mobile IPv6 operates in any location without any special support required from the local router.
- Support for route optimization is a fundamental part of the protocol, rather than a set of nonstandard extensions.
- Mobile IPv6 route optimizations can operate securely even without prearranged security associations. It is expected that the route optimizations can be deployed on a global scale among all mobile-node correspondent nodes.
- Support is also integrated into Mobile IPv6 for allowing route optimizations to coexist with routers that perform ingress filtering.
- The IPv6 *Neighbor Unreachability Detection* assures symmetric reachability between the mobile node and its default router in the current location.
- Most packets sent to a mobile node away from home in Mobile IPv6 are sent using an IPv6 routing header rather than IP encapsulation, reducing the amount of resulting overhead compared to Mobile IPv4.
- Mobile IPv6 is decoupled from any particular link layer because it uses IPv6 Neighbor Discovery instead of IPv4 *Address Resolution Protocol* (ARP). This also improves the robustness of the protocol.
- The use of IPv6 encapsulation (and the routing header) removes the need in Mobile IPv6 to manage tunnel soft state.
- The dynamic home-agent address discovery mechanism in Mobile IPv6 returns a single reply to the mobile node. The directed broadcast used in IPv4 returns separate replies from each home agent.

Summary

This article has presented three of the key technology advantages of IPv6 behind the wall. There are others, but they are technically too complex to define in a short article, but rather the subject of IPv6 implementation white papers. The IPv6 architecture extends the potential for the Next-Generation Internet to support rapid renumbering of networks, Quality of Service, extensions for ad hoc networks, and the hope of extending the Internet beyond the capabilities and functions today with IPv4. Most important is that IPv6 enhancements will be developed without using "band aids," as is currently being done with today's IPv4 architecture. The author of this article would like to thank Tony Hain and Patrick Grossetete from Cisco Systems for their review.

For Further Reading

[1] R. Droms, Ed., J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3315, July 2003.

[2] R. Hinden, S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture," RFC 3513, April 2003.

[3] R. Hinden, S. Deering, E. Nordmark, "IPv6 Global Unicast Address Format," RFC 3587, August 2003.

Additional information regarding IPv6 can be found at the International IPv6 Forum Web site **www.ipv6forum.com** and the North American IPv6 Task Force Web site **www.nav6tf.org**. Specifically, readers can view the IPv6 Forum basic value proposition at:

http://www.nav6tf.org/summit_slides/IPv6_Value_Proposition_June_2003final.ppt

JIM BOUND works at Hewlett Packard Corporation as an HP Fellow and is a Network Technical Director within the Enterprise UNIX (HP-UX) Division's Network and Security Lab Engineering Group. Jim was a member of the Internet Protocol Next Generation (IPng) Directorate within the IETF, which selected IPv6, among several proposals, to become the basis of the IETF's work on an IPng in 1994. Jim has been a key designer and implementor of IPv6, and contributor and coauthor of IPv6 specifications. Jim founded an ad-hoc IPv6 deployment group working with implementors across the Internet in 1998, which became the IPv6 Forum, where Jim is now Chair of the IPv6 Forum Technical Directorate and Member of the Board of Directors. Jim is also Chair of the North American IPv6 Task Force. Jim is a pioneer member of the Internet Society, and member of the Institute of Electrical and Electronics Engineers (IEEE). In July 2001, Jim received the IPv6 Forum Internet IPv6 Pioneer Award as the IPv6 Forum's "Lead Plumber." Jim has been working in the field of networking as engineer and architect since 1978, and is a subject matter expert to government and industry, for IPv6 and network-centric technology. E-mail: **jim.bound@hp.com**

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

Fragments

Peter T. Kirstein Receives Postel Award

Peter Kirstein is this year's recipient of the prestigious *Jonathan B. Postel Service Award*. A founding member of the Internet Society, Professor Kirstein is one of the pioneers of the Internet and was directly involved with its development and evolution. He was awarded the Postel Service Award in recognition of his foresight, persistence and innovation in navigating international technical and political complexities, and thus enabling the global propagation of the Internet. The Postel Award was presented on July 16, during the 57th meeting of the *Internet Engineering Task Force* (IETF) in Vienna, Austria.

"The Internet Society is pleased to recognize Peter's significant contribution to the development of the Internet by awarding him this year's Postel Award," said Internet Society President/CEO Lynn St. Amour. "His commitment to the evolution and growth of the Internet, particularly during the 1970s, made possible the global infrastructure we have today. And, his efforts continue, most recently working in the Southern Caucasus and Central Asia regions." Steve Crocker, noted Internet authority and chair of this year's Postel award committee, commented on Kirstein's foresight in laying the groundwork for the Internet's global scope. "Peter Kirstein saw that the future of networking lay in international cooperation and interconnection, and deftly organized the steps to make it happen. He used both technical and personal skills and enabled many others to do magnificent work."

In 1973, Kirstein established one of the first two international nodes of the ARPANET, playing a very active part in the ensuing SATNET activity, which covered five countries. His group continued to provide the principal Internet link between the UK and the US throughout the 1980s, during which time he was responsible for both the **`.UK`** and **`.INT`** domains. He continues to collaborate in US *Defense Advanced Research Agency* (DARPA) programs. He has led six European projects in computers and communications funded by the European Commission, and participated in twelve more. Currently, he is leading the *Silk Project,* which is providing satellite-based Internet access to the Newly Independent States in the Southern Caucasus and Central Asia. In June, he was awarded a *Commander, Order of the British Empire,* for his services to Internetworking research.

He has chaired the International Collaboration Board, which currently involves six NATO countries, since 1983, and served on the Networking Panel of the *NATO Science Committee* (serving as chair in 2001). He has been on Advisory Committees for the *Australian Research Council,* the *Canadian Department of Communications,* the German GMD, and the Indian *Education and Research Network* (ERNET) Project. Kirstein obtained his undergraduate degree in Mathematics and Engineering from Gonville and Caius College, Cambridge University, his PhD in Electrical Engineering from Stanford University, and was awarded a DSc in Engineering from the University of London.

Kirstein expressed his appreciation for the award and respect for Jon Postel's work, explaining, "Postel's efforts to ensure the successful development and deployment of the Internet was an inspiration to us all. His stewardship of the RFC series was essential to the successful development of the Internet. His conscientious and painstaking operation of the Domain Name System and the Internet Assigned Numbers Authority were indispensable to the international growth of the system. I am particularly pleased to be recipient of an award in his name, and feel greatly honored to be considered worthy of having my activities linked with his memorial."

The Jonathan B. Postel Service Award was established by the Internet Society to honor those who have made outstanding contributions in service to the data communications community. The award is focused on sustained and substantial technical contributions, service to the community, and leadership. With respect to leadership, the nominating committee places particular emphasis on candidates who have supported and enabled others in addition to their own specific actions.

The award is named after Dr. Jonathan B. Postel, who embodied all of these qualities during his extraordinary stewardship over the course of a thirty-year career in networking. He served as the editor of the RFC series of notes from its inception in 1969, until 1998. He also served as the ARPANET "numbers Czar" and the Internet Assigned Numbers Authority over the same period of time. He was a founding member of the *Internet Architecture* (nee *Activities*) *Board* (IAB) and the first individual member of the Internet Society, where he also served as a trustee.

Previous recipients of the Postel Award include Jon himself (posthumously and accepted by his mother), Scott Bradner, Daniel Karrenberg and Stephen Wolff. The award consists of an engraved crystal globe and $20,000.

The *Internet Society* (ISOC) (**`www.isoc.org`**) is a not-for-profit membership organization founded in 1991 to provide leadership in Internet related standards, education, and policy. With offices in Washington, DC, and Geneva, Switzerland, it is dedicated to ensuring the open development, evolution and use of the Internet for the benefit of people throughout the world. ISOC is the organizational home of the IETF, the IAB, the *Internet Engineering Steering Group* (IESG) and other Internet-related bodies who together play a critical role in ensuring that the Internet develops in a stable and open manner. For over 12 years ISOC has run international network training programs for developing countries and these have played a vital role in setting up the Internet connections and networks in virtually every country connecting to the Internet during this time.

Deployment of Internationalized Domain Names

The *Internet Corporation for Assigned Names and Numbers* (ICANN) recently announced the commencement of global deployment of *Internationalized Domain Names* (IDNs)[2,3,4], which will allow use on the Internet of domain names in languages used in all parts of the world.

In October 2002, the IESG approved the publication of a standardized way of integrating IDNs into the Internet's *Domain Name System* (DNS). After the proposed technical standard was published in March 2003, the ICANN Board endorsed an approach for implementation of the technical standard that had been developed cooperatively by ICANN and leading IDN registries.

Following up on the Board's endorsement, ICANN and the leading IDN registries finalized an agreed text of the principles to be followed in IDN registration activities. Those "Guidelines for the Implementation of Internationalized Domain Names"[1] were published. IDN registries adhering to the Guidelines will employ language-specific registration and administration rules that are documented and publicly available. These IDN registries will work collaboratively with each other and with interested stakeholders to develop the language-specific policies, with the objective of achieving consistent approaches to IDN implementation to maintain Internet interoperability for the benefit of DNS users worldwide.

The registries for the **.cn** (China), **.jp** (Japan), and **.tw** (Taiwan) country codes, as well as for the **.info** and **.org** generic top-level domains, have committed to adhere to the Guidelines. As authorized by the ICANN Board in March, registries seeking to deploy IDNs under their agreements with ICANN will be authorized to do so on the basis of the Guidelines. In addition, the ICANN Board has recommended the Guidelines to other registries, and encourages broad participation by registries, language experts, and others in consultative, collaborative, community-based processes to study and develop appropriate language-specific IDN registration rules and policies.

As the deployment of IDNs proceeds, ICANN and the participating IDN registries have agreed to work together to review Guidelines at regular intervals based on their deployment experience, and to make any necessary adjustments.

[1] **http://www.icann.org/general/idn-guidelines-20jun03.htm**

[2] P. Faltstrom, P. Hoffman, A. Costello, "Internationalizing Domain Names in Applications (IDNA)," RFC 3490, March 2003.

[3] P. Hoffman, M. Blanchet, "Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)," RFC 3491, March 2003.

[4] A. Costello "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)," RFC 3492, March 2003.

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

December 2003 Volume 6, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

I will remember 2003 as the year when high-speed Internet access became widely available in public locations such as airports, hotels, and coffee shops. As a frequent traveler, I really appreciate not having to find a suitable telephone jack and corresponding country-specific telephone adapter plug in order to get my e-mail. The IEEE 802.11 "WiFi" standard has truly arrived. I even stayed in a new hotel in Norway that provided WiFi access in every room by placing base stations in the hallways. When I first stepped into my hotel room and noticed that it had only a *digital* telephone and no sign of any Ethernet jacks I worried, but a quick check revealed that I could purchase a scratch-off card at reception that provided me with a username and password valid for 24 hours. A clear example of a "technology generation leap."

The year 2003 was also the year in which unsolicited e-mail, or "spam," became a major problem for all Internet users. Various filtering systems have thankfully been devised and deployed, but this problem has no easy solution. It will be interesting to see what impact new antispam legislation will have over the coming months and years.

The first article presents an in-depth look at the IP Version 4 address space and its measured and projected consumption rate. When work first started on the design of IP Version 6, projections indicated that we'd run out of IPv4 addresses within a few years. Geoff Huston takes a fresh look at this in an article entitled "IPv4—How long do we have?"

The job of System Administrator, or "sysadmin," is a challenging one, and if your job includes keeping the network running 24 hours a day, you will probably appreciate some of the tips in our second article, entitled "Low-Tech Network Maintenance."

For the second time recently, Queen Elizabeth II has honored an Internet pioneer. Tim Berners-Lee, the inventor of the World Wide Web and director of the *World Wide Web Consortium* (W3C), was made a *Knight Commander, Order of the British Empire* in the 2004 New Years Honours list. (See "Fragments," page 28).

Which brings us to the IPJ publication schedule. If you are a regular subscriber to the IPJ, you probably have noticed a somewhat irregular publishing schedule in 2003. This December 2003 issue is indeed being published in January 2004. This results from our effort to produce timely quality articles in a world where the experts are not staff writers. Of course, you should still expect to receive four issues per year, and your feedback to **ipj@cisco.com** will help make IPJ even better.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

IPv4—How long do we have?

by Geoff Huston, Telstra

One of those stories that keeps on appearing from time to time is the claim that somewhere in the world, or even all over the world, we are "running out of IP addresses," referring to the consumption of unallocated IPv4 addresses[1]. In one sense this is a pretty safe claim, in that the IPv4 address pool is indeed finite, and, as the IPv4 Internet grows it makes continual demands on previously unallocated address space. So the claim that the space will be exhausted at some time in the future is a relatively safe prediction. But the critical question is not "if" but "when," because this is a question upon which many of our current technology choices are based.

Given this revived interest in the anticipated longevity of the IPv4 address space, it is timely to revisit a particular piece of analysis that has been a topic of some interest at various times over the past decade or more. The basic question is: "How long can the IPv4 address pool last in the face of a continually growing network?" This article looks at one approach to attempt to provide some indication of "when." Like all predictive exercises, many assumptions have to be made, and the approach described here uses just one of numerous possible predictive models—and, of course, the future is always uncertain.

The IPv4 Address Space

The initial design of IPv4 was extremely radical for its time in the late 1970s. Other contemporary vendor-based computer networking protocols were designed within the constraints of minimizing the packet header overhead in order to improve the data payload efficiency of each packet. At the time address spans were defined within the overall assumption that the networks were deployed as a means of clustering equipment around a central mainframe. In many protocol designs 16 bits of address space in the packet headers was considered to be extravagant. To use a globally unique address framework of 32 bits to address network hosts was, at the time, a major shift in thinking about computer networks from a collection of disparate private facilities into a truly public utility.

To further add to the radical nature of the exercise, the Internet Network Information Center was prepared to hand out unique blocks of this address space to anyone who submitted an application. Address deployment architectures in other contemporary protocols did not have the address space to support such address distribution functions, nor did they even see a need for global uniqueness of computer network addresses. Network administrators numbered their isolated corporate or campus networks starting at the equivalent of "1," and progressed onward from there. Obviously network splits and mergers caused considerable realignment of these private addressing schemes, with consequent disruption to the network service.

By comparison, it seemed, the address architecture of the Internet was explicitly designed for interconnection. But even with 32 bits to use in an address field, getting the right internal structure for addresses is not as straightforward as it may initially seem.

The Evolution of the IPv4 Address Architecture

IP uses the address to express two aspects of a connected device: the identity of this device (endpoint identity) and the location within the network where this device can be reached (location or forwarding identity). The original IP address architecture used the endpoint identity to allow devices to refer to each other in end-to-end application transactions, whereas within the network the address is used to direct packet-forwarding decisions. The address was further structured into two fields: a *network* identifier and a *host* identifier within that network. The first incarnation of this address architecture used a division at the first octet: the first 8 bits were the network number and the following 24 bits were the host identifier. The underlying assumption was one of deployment across a small number of very large local networks. This view was subsequently refined, and the concept of a class-based address architecture was devised for the Internet. Half of the address space was left as a 8/24-bit structure, called the *Class A* space (allowing for up to 127 networks each with 16,777,216 host identities). A quarter of the remaining space used a 16/16-bit split (allowing for up to 16,128 networks, each with up to 65,536 hosts), defining the *Class B* space. A further eighth of the remaining space was divided using a 24/8-bit structure (allowing for 2,031,616 networks, each with up to 256 hosts), termed the *Class C* space. The remaining eighth of the space was held in reserve.

This address scheme was devised in the early 1980s, and within a decade it was pretty clear that there was a problem with impending exhaustion. The reason was an evident run on Class B addresses. Although very few entities could see their IP network spanning millions of computers, the personal desktop computer was now a well-established part of the landscape, and networks of just 256 hosts were just too small. So if the Class A space was too big, and the Class C too small, then Class B was the only remaining option. In fact, the Class B blocks were also too large, and most networks that used a Class B address consumed only a few hundred of the 65,535 possible host identities within each network. The addressing efficiency of this arrangement was very low, and a large amount of address space was being consumed in order to number a small set of devices. Achieving even a 1 percent host density (expressed as a ratio of number of addressed hosts to the total number of host addresses available) was better than normal at the time, and 10 percent was considered pretty exceptional.

Consequently, Class B networks were being assigned to networks at an exponentially increasing rate. Projections from the early 1990s forecast exhaustion of the Class B space by the mid-1990s. Obviously there was a problem, and the *Internet Engineering Task Force* (IETF) took on the task of finding some solutions. Numerous responses were devised by the IETF.

As a means of mitigation of the immediate problem, the IETF altered the structure of an IP address. Rather than having a fixed-length network identifier of 8, 16, or 24 bits, the network part of the address could be any length at all, and a network identifier was now the couplet of an IP address field containing a network part and the bit length of the network part. The boundary between the network and host part could change across the network, so rather than having "networks" and "subnetworks" as in the class-based address architecture, there was the concept of a variable length network mask. This was termed the "classless" address architecture (or "CIDR"), and the step was considered to be a short-term expediency to buy some additional time before address exhaustion. The longer-term plan was to develop a new IP architecture that could encompass a much larger connectivity domain than was possible with IPv4.

We now have IPv6 as the longer-term outcome. But what has happened to the short-term expediency of the classless address architecture in IPv4? It appears to have worked very well indeed so far, and now the question is: how long can this supposedly short-term solution last?

Predictions of Address Consumption

Predicting the point of IPv4 address exhaustion has happened from time to time since the early 1990s within the IETF[2]. The initial outcomes of these predictive exercises were clearly visible by the mid-1990s: the classless address architecture was very effective in improving the address utilization efficiency, and the pressures of ever-increasing consumption of a visibly finite address resource were alleviated. But a decade after the introduction of CIDR addressing, it is time to understand where we are heading with the consumption of the underlying network address pool.

Dividing up the Address Space

There are three stages in address allocation. The pool of IP addresses is managed by the *Internet Assigned Numbers Authority* (IANA). Blocks of addresses are allocated to *Regional Internet Registries* (RIRs), who in turn allocate smaller blocks to *Local Internet Registries* (LIRs) or *Internet Service Providers* (ISPs).

Currently 3,707,764,736 addresses are managed in this way. It is probably easier to look at this in terms of the number of "/8 blocks," where each block is the same size as the old Class A network, namely 16,777,216 addresses. The total address pool is 221 /8s, with a further 16 /8s reserved for multicast use, 16 /8s held in reserve, and 3 /8s designated as not for use in the public Internet.

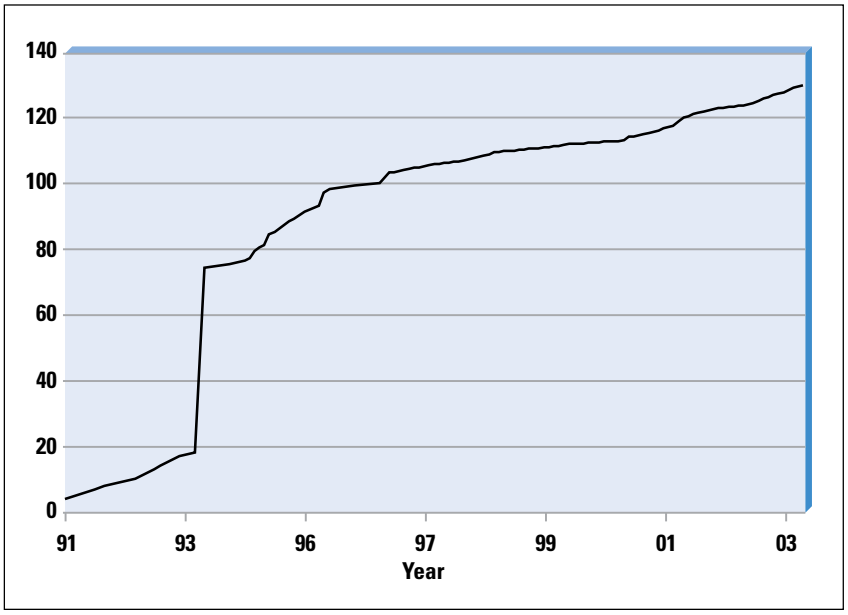
In looking at futures, there are three sources of data concerning address consumption:

- How quickly is the IANA passing address blocks to the RIRs, and when will IANA run out?
- How quickly are the RIRs passing address blocks to LIRs, and when will this run out?
- How much address space is actually used in the global Internet, and how quickly is this growing? When will this run out?

The IANA Registry

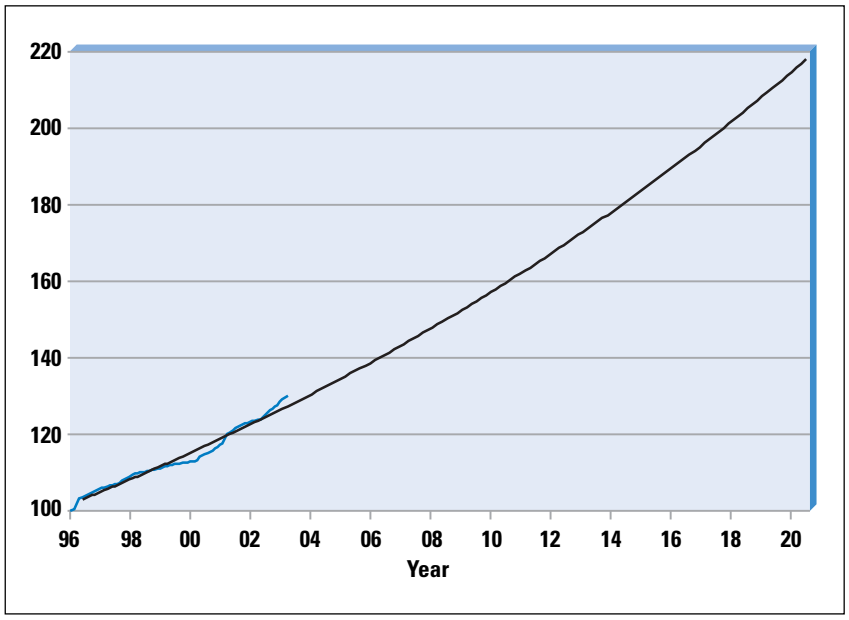
So the first place to look is the IANA registry file[3]. This registry reveals that of these 221 /8 blocks, 89 /8 blocks are still held as unallocated by the IANA, 129.9 /8 blocks have been allocated, and the remaining 2.1 / 8 blocks are reserved for other uses. The IANA registry also includes the date of allocation of the address block, so it is possible to construct a time series of IANA allocations, as shown in Figure 1.

Figure 1: IANA Allocated IPv4 /8 Address Blocks



Interestingly, there is nothing older than 1991 in this registry. This exposes one of the problems with analyzing registry data, in that there is a difference between the current status of a registry and a time-stamped log of the transactions that were made to the registry over time. The data published by the IANA is somewhere between the two, and the log data is incomplete; in addition, the current status of some address blocks is unclear. It appears that the usable allocation data starts in 1995. So if we take the data starting from 1995 and perform a linear regression to find a best fit of an exponential projection, it is possible to make some predictions as to the time it will take to exhaust the remaining unallocated 89 /8s. (Figure 2).

Figure 2: IANA Allocated IPv4 /8 Address Blocks



It is worth a slight digression into the method of projection being used here. The technique is one of using a best fit of an exponential growth curve to the data. The underlying assumption behind such a projection is that the growth rate of the data is proportional to the size of the data, rather than being a constant rate. In network terms, this assumes that the rate of consumption of unallocated addresses is a fixed proportion of the number of allocated addresses, or, in other words, the expansion rate of the network is a proportion of its size, rather than being a constant value. Such exponential growth models may not necessarily be the best fit to a network growth model, although the data since 1995 does indicate an underlying exponential growth pattern. Whether this growth model will continue into the future is an open issue.

The projection of 2019 as the date for consumption of the unallocated address space using this technique is perhaps surprising, because it seems that the network is bigger now than ever, yet the amount of additional address space required to fuel further accelerating growth for a further decade is comparatively small. This is true for many reasons, and the turning point when these aspects gained traction in the Internet appeared to be about 1995. They include:

- The first 1.6 billion addresses (equivalent to some 100 /8 blocks) were allocated using the class-based address architecture. Since this date address allocation has used a classless architecture, and this has enabled achievement of significantly improved efficiencies in using the address space.
- The RIRs came into the picture, and started using conservation-based policies in address allocations. The RIR process requires all address applicants to demonstrate that they can make efficient and effective use of the address space, and this has dampened some of the wilder sets of expectations about the address requirements of an enterprise.

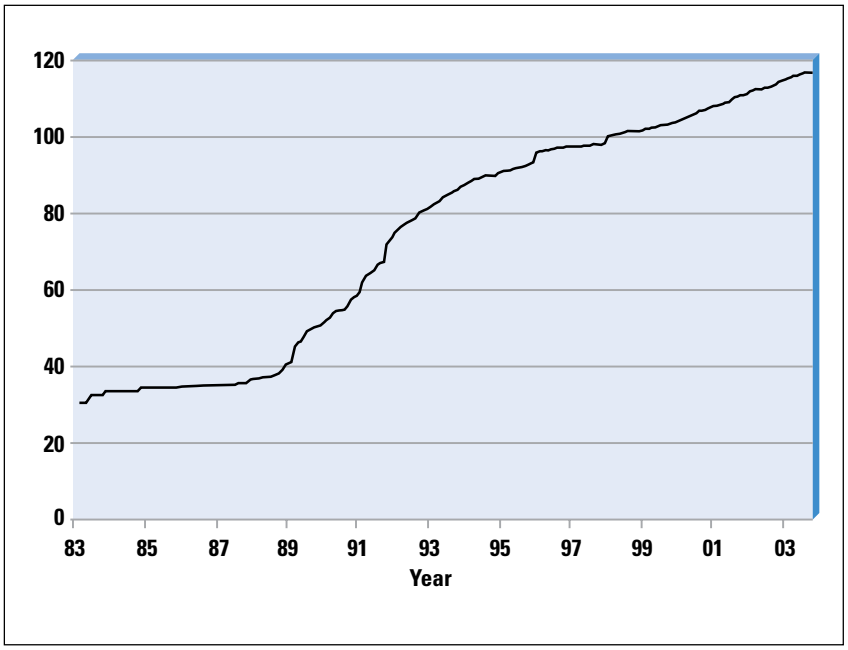
- Address compression technologies became widely deployed. Dynamic *Network Address Translation* (NAT) devices have, for better or worse, become a common part of the network landscape. NAT devices allow large "semi-private" networks to use a very small poll of public addresses as the external view of the network, while using private address space within the network. *Dynamic Host Configuration Protocol* (DHCP) has allowed networks to recycle a smaller pool of addresses across a larger set of intermittently connected devices.

Whether these factors will continue to operate in the same fashion in the future is an open question. Whether future growth in the use of public address space operates from a basis of a steadily accelerated growth is also an open question. The assumption made in this exercise is that the projections depend on continuity of effectiveness of the RIR policies and their application, continuity of technology approaches, and absence of disruptive triggers. Although the RIRs have a very well-regarded track record and there are strong grounds for confidence that this will continue, obviously the latter two assumptions about technology and disruptive events are not all that comfortable. With that in mind, the next step is to look at the RIR assignment data.

The RIR Registries

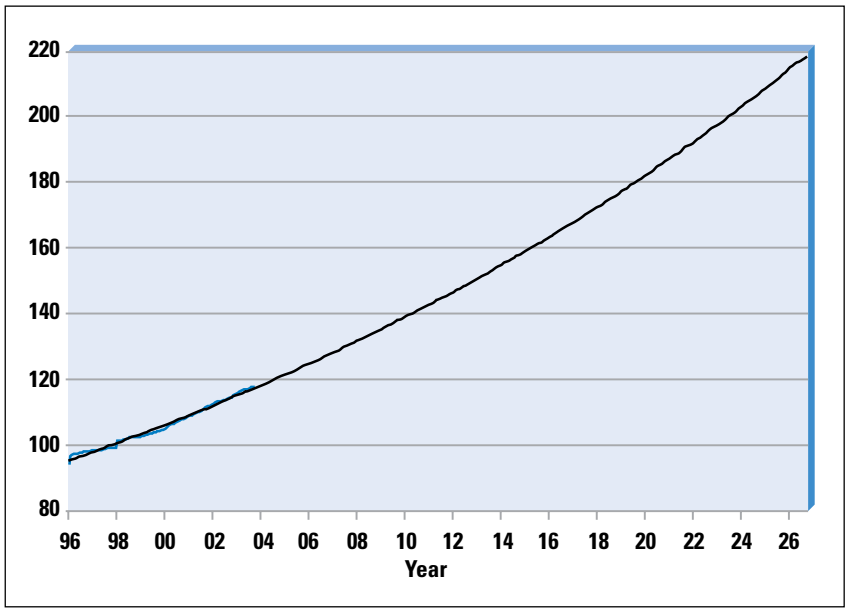
The RIRs also publish a registry of their transactions in "stats" files. For each currently allocated or assigned address block the RIRs have recorded, among other items, the date of the RIR assignment transaction that assigned an address block to a LIR or ISP. Using this data we can break up the 129.9 /8 blocks further, and it is evident that the equivalent of 116.7 /8 blocks have been allocated or assigned by the RIRs, and the remaining space, where there is no RIR allocation or assignment record, is the equivalent of 13.2 /8 blocks. These transactions can again be placed in a time series, as shown in Figure 3.

Figure 3: RIR Assigned IPv4 /8 Address Blocks



The post-1995 data used to extrapolate forward using the same linear regression technique described previously to find a curve of best bit using the same underlying growth model assumptions yields:

Figure 4: RIR Assigned IPv4 /8 Address Blocks—Projection



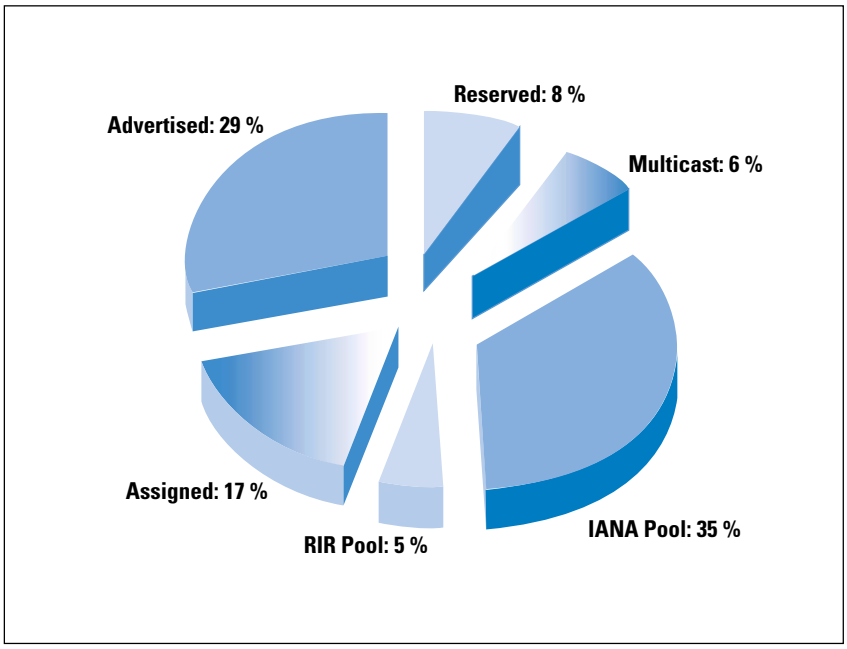
This form of extrapolation gives a date of 2026 for the time at which the RIRs will exhaust the number pool. Again the same caveats about the use of this approach as a reliable predictor apply here, and the view forward is based on the absence of large-scale disruptions, or some externally induced change in the underlying growth models for address demand.

The BGP Routing Table

When addresses are assigned to end networks, the expectation is that these addresses will be announced to the network in the form of routing advertisements. So some proportion of these addresses is announced in the Internet routing table. The next task is to establish the trends of the amount of address space covered by the routing table. The approach used has been to take a single view of the address span of the Internet. This is the view from one point, inside the AS1221 network operated by Telstra.

The data as of October 2003 shows that some 29 percent of the total IPv4 address space is announced in the *Border Gateway Protocol* (BGP) routing table, whereas 17 percent has been allocated to an end user or LIR but is not announced on the public Internet as being connected and reachable. A total of 5 percent of the address space is held by the RIR's pending assignment or allocation (or at least there is no RIR recorded assignment of the space), while 35 percent of the total space remains in the IANA unallocated pool. A further 8 percent of the space is held in reserve (Figure 5).

Figure 5: IPv4 /8 Address Space



This BGP data is based on an hourly inspection of the amount of address space advertised within the Internet routing table. The data collection commenced in late 1999, and the data gathered so far is shown in Figure 6. The problem with this data is that there is some considerable amount of fluctuation in the amount of address space advertised over time. The major step changes are due to a small number of /8 advertisements that periodically are announced and withdrawn in BGP. In order to obtain reasonable data for generating projections, some noise reduction on this data needs to be undertaken. The approach used has been to first filter the data using a constant value of 18 /8 prefix announcements, and then use a sliding average function to create a smoothed time series. This is indicated in Figure 7.

The critical issue when using this data for projection is to determine what form of function can provide a best fit to the data. A good indication of the underlying trends in the data can be found by analyzing the first-order differential of the data. An underlying increasing growth model would have an increasing first-order differential, whereas a decreasing growth model would have a negatively inclined differential. A least-squares best-fit analysis of the data shows that the growth rates have not been consistent over the past three years. A reasonable fit for this data appears to be a constant growth model, or a linear growth projection, with a consumption rate of some 3 /8 blocks per year.

Figure 6: Advertised IPv4 /8 Address Space (/8 Blocks)

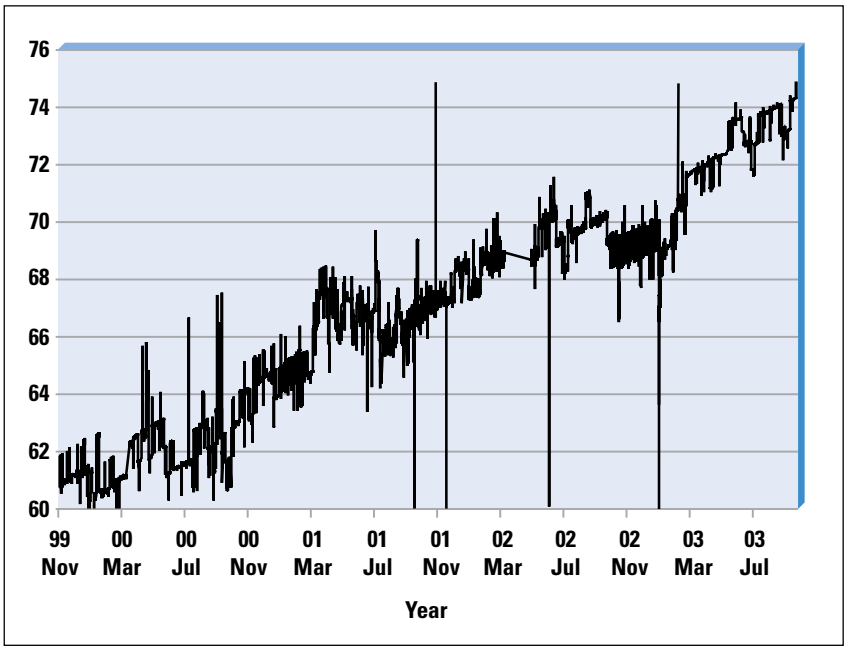
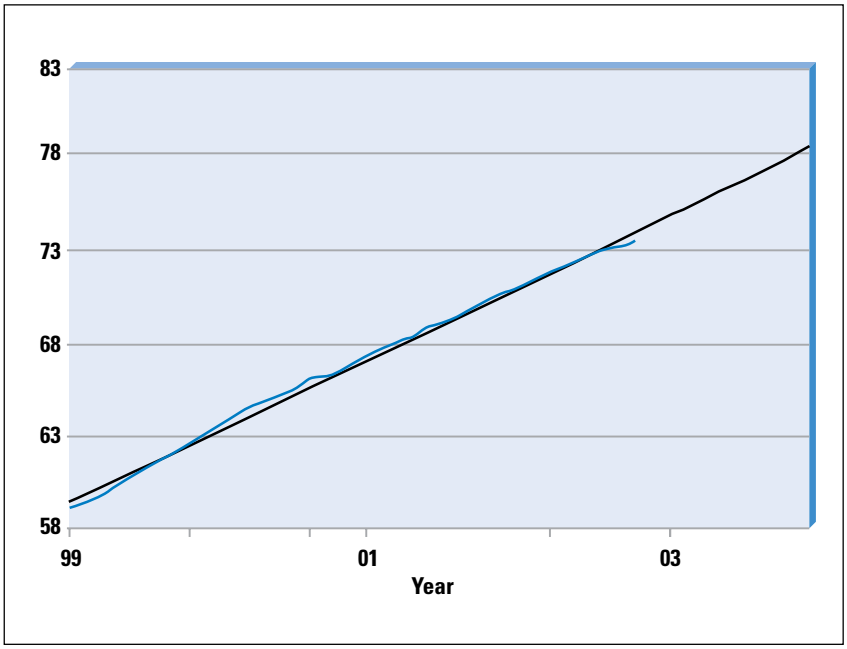


Figure 7: Smoothed IPv4 /8 Advertised Address Blocks



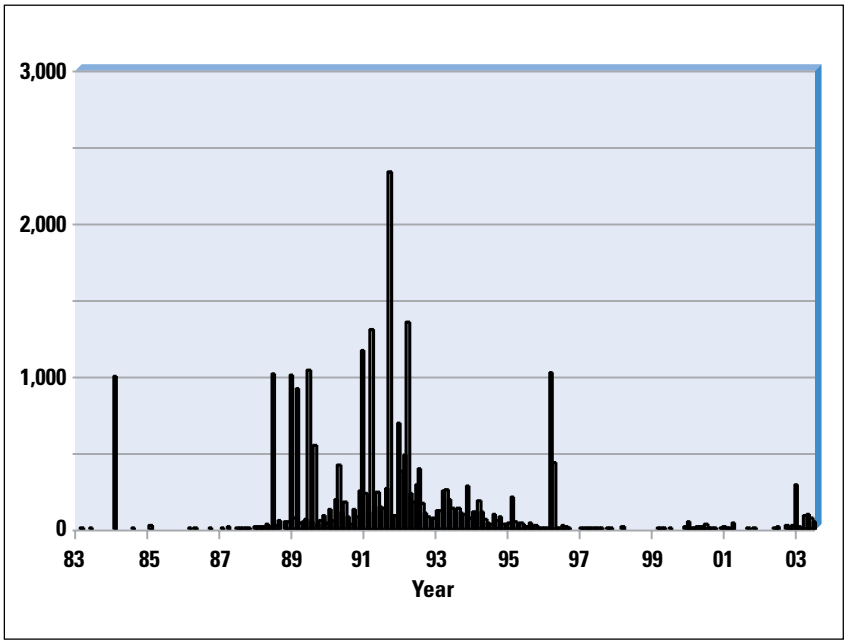
Combining the Three Views

One question remains before we complete the projections for IPv4 address space. There are 43.3 /8 blocks, or some 17 percent of the total IPv4 address space that has been allocated for use, but is not visible in the Internet routing table. This is a very significant amount of address space, and if it is growing at the same rate as the advertised space, then this will have a significant impact on any overall model of consumption of the use of address space.

The question here is whether this "invisible" address pool is a legacy of the address allocations policies in place before the RIR system came into operation in the mid 1990s, or some intrinsic inefficiency in the current system. If it is the latter, then it is likely that this pool of unannounced addresses will grow in direct proportion to the growth in the announced address space, whereas if it is the former, then the size of the pool will remain relatively constant in the future.

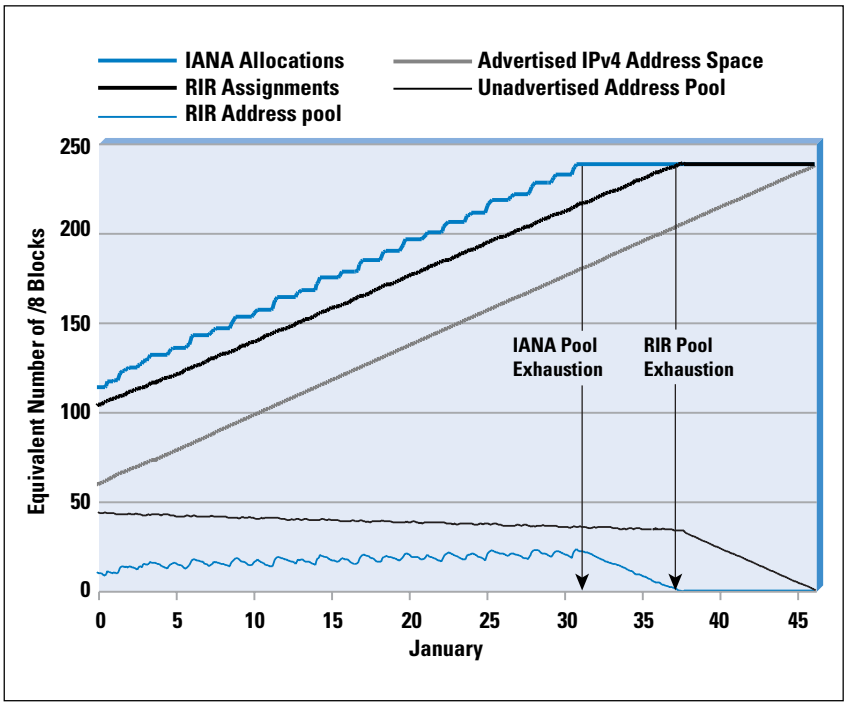
We can look back through the RIR allocation data and look at the allocation dates of unannounced address space (Figure 8). This view indicates that the bulk of the space is a legacy of earlier address allocation practices, and that since 1997, when the RIR operation was fully established, there is an almost complete mapping of RIR allocated address space to BGP routing announcements. The recent 2003 data indicates that there is some lag between recent allocations and BGP announcements, most probably due to the time lag between an LIR receiving an allocation and subsequent assignments to end users and advertisement in the routing table.

Figure 8: Age Distribution of Unadvertised Address Blocks (/8 Address Blocks)



This confirms that in recent years all the address space that has been assigned by the RIRs appears in the Internet routing table, implying that projections of the amount of address space advertised in the routing table is a good correlation to projections of address space consumption. With this in mind it is now possible to construct a model of the address distribution process, working backward from the BGP routing table address size. From the sum of the BGP table size and the LIR holding pool, we can derive the total RIR-managed address pool. To this number is added the RIR holding pool low size and its low threshold where a further IANA allocation is required. This allows a view of the entire system, projected forward over time, where the central driver for the projection is the growth in the network itself, as described by the size of the announced IPv4 address space. This is shown in Figure 9.

Figure 9: IPv4 Projections of Address Consumption



It would appear that the point of effective exhaustion is the point where the RIRs exhaust available address space to assign. In this model, RIR exhaustion of the unallocated address pool would occur in 2037.

Uncertainties

Of course such projections are based on the underlying assumption that tomorrow will be much like today, and the visible changes that have occurred in the past will smoothly translate to continued change the future. This assumption obviously has some weaknesses, and many events could disrupt this prediction.

Some disruptions could be found in technology evolution. An upward shift in address take-up rates could occur because of an inability of NAT devices to support emerging popular applications. Widespread deployment of peer-to-peer applications implies the need for persistent address presentation, which may imply greater levels of requirement for public address space. The use of personal mobile IP devices (such as PDAs in their various formats) using public IPv4 addresses would place a massive load on the address space, simply because of the very large volumes associated with deployment of this technology[4].

Other disruptions have a social origin, such as the boom and bust cycle of Internet expansion in recent years. Another form of disruption in this category could be the adoption of a change in the distribution function. The current RIR and LIR distribution model has been very effective in limiting the amount of accumulation of address space in holding pools, and allocating addresses based on efficiency of utilization and conformance to the routing topology of the network.

Many other forms of global resource distribution use a geopolitical framework, where number blocks are passed to national entities, and further distribution is a matter of local policy[5]. The disruptive nature of such a change would be to immediately increase the number of "holding" points in the distribution system, locking away larger pools of address space from being deployed and advertised and generating a significant upward change in the overall address consumption rates due to an increase in the inefficiency of the altered distribution function.

The other factor to be aware of is the steadily decreasing "buffer" of unallocated addresses that can be used to absorb the impacts of a disruptive change in address consumption rates. Although at present some 60 percent of the address space—or some 2.6 billion addresses—are available in the unallocated address pools or held in reserve, this pool will reduce over time. If a disruptive event is, for example, a requirement to directly address some 500 million devices, then such an event would reduce the expectancy of address space availability by some years, assuming it occurred within the period when sufficient address space remains to meet such a surge of demand.

The other source of uncertainty is that this form of predictive modeling assumes that the ratios of actual connected devices and the amount of address space deployed to service this device pool remain relatively constant.

This model also assumes some form of continuity of current address allocation polices. This is not a likely scenario, because it is likely that address policies will reflect some notion of balance between the level of current demand against future demands. As the unallocated address pool shrinks it is possible that policies will alter to express the increased level of competitive demand for the remaining resource. Consumption rates would be moderated by such a change in allocation policy. The commonly cited intended evolutionary path for the Internet is to a transition to ubiquitous use of IPv6, and at some point in that transition process it is reasonable to assume that further demands for IPv4 space will dwindle. It may be that at such a "crossover" time allocation policies may then be altered to reflect a drop in both current and future demands for IPv4 address space.

In attempting to assess the possible future path of address allocation policies, it is also evident that, from a market rationalist perspective, there is a certain contrivedness about the current address allocation process. The current address management system assumes a steady influx of new addresses to meet emerging demands, and the overall address utilization efficiency is not set by any form of market force, but by the outcomes of the application of RIR address allocation policies to new requests for address space. A market rationalist could well point to the use of market price as a means of determining the most economically efficient form of utilization of a commodity product. Such a position is based on the observation that the way that the consumer chooses between alternative substitutable services is by a market choice that is generally price sensitive.

If price is removed from an IPv4 address market, the choices made by market players are not necessarily the most efficient choices, and some would argue that the current situation underprices IPv4 at the expense of IPv6.

However, in venturing into these areas we are perhaps straying a little too far from exploring the degree of uncertainty in these predictive exercises. A discussion of the interaction between various forms of distribution frameworks and likely technology outcomes is perhaps a topic for another time.

So just how long does IPv4 have?

The assumptions used here include assuming that the trends in the growth in the advertised space are directly proportional to the future consumption rates for IP addresses, and that the constant growth model remains a best fit for this time series of data. It also assumes a continuation of the current utilization efficiency levels in the Internet, a continuing balance between public address utilization and the use of various forms of address compression, and continuity of current address allocation policies, as well as the absence of highly disruptive events. With all this in mind, then it would appear that the IPv4 world, in terms of address availability, could continue for up to another three decades or so without reaching any fixed boundary of exhaustion.

But it must be remembered that each of these assumptions is relatively sweeping, and to combine them as we have done here is pushing the predictive exercise to its limits, or possibly beyond them. Three decades out is way over the event horizon for any form of useful prediction for the Internet, so if we restrict the question to at most the next five to eight years, then we can answer with some level of confidence that, in the absence of any significant disruptions to the current deployment model of the Internet, there is really no visible evidence that IPv4 will exhaust its address pool by 2010, based on the available address consumption data.

Data Sources

IANA IPv4 Address Registry:
`http://www.iana.org/assignments/ipv4-address-space`

Registry "stats" report files:
APNIC: **`ftp://ftp.apnic.net/pub/apnic/stats`**
ARIN: **`ftp://ftp.arin.net/pub/stats`**
LACNIC: **`ftp://ftp.lacnic.net/pub/stats`**
RIPE NCC: **`ftp://ftp.ripe.net/ripe/stats`**
BGP Address Data: **`http://bgp.potaroo.net`**

Notes

[1] "Tackling the net's number shortage." BBC News, World Edition, 26 October 2003. The item starts with the claim: "BBC ClickOnline's Ian Hardy investigates what is going to happen when the number of net addresses—Internet Protocol numbers—runs out sometime in 2005."
http://news.bbc.co.uk/2/hi/technology/3211035.stm

[2] The work was undertaken in the *Address Lifetime Expectations* (ALE) Working Group of the IETF in 1993–1994. The final outcome from this effort was reported from the December 1994 meeting of this group: "Both models currently suggest that IPv4 addresses would be depleted around 2008, give or take three years."

[3] This registry is online at:
http://www.iana.org/assignments/ipv4-address-space

[4] On the other hand, it is evident that the growth of the Internet in recent years has been fueled by the increasing prevalence of NAT devices. In order for applications to be accepted into common use in today's Internet, they need to be able to function through various NAT-based constraints, and increasing sophistication of applications in operating across NAT devices is certainly evident today.

[5] Such a geopolitical distribution system is used in the E.164 number space for telephony ("ENUM").

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Scientist in the Internet area for Telstra. He is also the Executive Director of the Internet Architecture Board, and is a member of the APNIC Executive Committee. He is author of *The ISP Survival Guide*, ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks*, ISBN 0471-378089, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@telstra.net**

Low-Tech Network Maintenance

by Locum sysadmin

In an ideal world, we all maintain networks composed of shiny, high-end equipment. Server rooms are stacked to the brim with racks of blinking lights. Neat bundles of cable wend their way through cable loops to orderly, labeled patch bays. When the occasional piece of equipment fails, a hot replacement is slotted in by trained technicians, often before users even notice the outage. Sleek, modern servers hum contentedly, offering their services all day, every day. All is well.

And then there are the other environments ...

Imagine, if you will, that you are a programmer, working for a small company. You are perhaps vaguely aware that all is not well with the small network that you use each day, but the system administrator (*sysadmin*, if there is one) is so busy with other duties that addressing your concerns seems to be last on the list. The occasional delay in CVS checkouts or e-mail that just never quite makes it seem like minor issues compared to... well, whatever it is that so occupies the sysadmin.

Or perhaps there is no sysadmin ... the network topology is neither ring, nor star, but more "accreted." It is possible that the nephew of one of the managers was responsible for its setup. Like coral, successive waves of employees have washed over the network, leaving their small additions—a cheap 8-port hub here, some gaffer-taped wiring there.

You become aware that your LAN/WAN environment is a real-world test of how deeply Ethernet hubs may be cascaded. A trip to the server room (or server closet) reveals a mess of cabling that closely resembles blue spaghetti. Access to the outside world can take several forms, but it is not uncommon to find a couple of dialup modems lurking quietly in the mess, unnoticed until a failure in the regular link means a failover to the pleasures of 30 employees sharing a 33.6k modem. The concept of labeling cables never made it to this paleolithic theme park, so if you ever trip on one of the floor-dwelling blue vines, locating its original socket can be a challenging occupation.

The servers themselves seem to be an interactive museum display charting the history of computing up until the late 1990s. Old UNIX boxes spill a mess of cables and hard drives over the bench, generic white-box servers of unknown vintage litter the room, "Powered by Linux" or FreeBSD stickers adorning them. Discolored 15-inch monitors sometimes display a blue screen of death, letting you know that some people still love NT4. Assorted tape drives blink quietly away, backing up regularly, though no one seems quite sure what they are backing up, or how to recover them. An elderly Sun box whiles away its retirement transferring mail and playing host to the occasional crackers who exploit security holes in its ancient *sendmail*, then give up in disgust.

The spare parts for the network might occupy a shelf in the server room, or perhaps they nestle on top of a rack unit. A motley assortment of chewed-looking Category 5 cables, network cards so ancient that their manufacture date is in Roman numerals, and a sculpture of BNC connectors—the thought of turning here for help fills you with dread. A dead network adapter usually means a surreptitious raid of the petty cash and a trip to the local computer-parts store for a no-name Ethernet card.

Then—as it always does—disaster strikes. Somewhere, something goes wrong. One thing that you can be sure of is that it will happen at the worst possible time. It is likely that a crucial presentation will be under way, or perhaps a software release is due by close of business. Maybe you are hosting a server for a client, and the client has noticed its absence, and is on the phone, using words like "unscheduled outage" and "penalty clause." If your clients are so inclined, words like "kneecap" and "sledgehammer" might also be heard. Another fact you can be reasonably sure of is that the sysadmin will not be present, and the next-most technical person will be called upon to work up a minor miracle to fix the ailing network.

Sound far-fetched? Believe it or not, I have been in this situation more than once. What follows are some hints that may help in fixing networks in suboptimal conditions, and as always, with the understanding that it must be done as cheaply as possible.

Many of the hints use features found on Linux boxes, beloved for its technical excellence (and its low cost). Most of the tips here can be adapted for whatever type of operating system you have.

Audible Ping

Ping is the venerable tool that we all know and love, and is the reigning king of the low-tech diagnostic tools. Linux (and other operating systems that use GNU tools) features an extension to *ping* that produces a beep on receipt of a response. The *audible ping* is designated by the **`-a`** command-line option.

Something as simple as **`ping -a missinghost.your.net`**, left running from a console in the server room, can alert you when you have finally reestablished network connectivity. It is like having a cable tester that can traverse routers.

Where Are You?

In a server room full of unlabeled generic boxes, it can sometimes be tricky to know which box is which. The following conversation is typical:

Hapless1: "Okay, I've logged into **`srv7`** by SSH [*Secure Shell Protocol*], and I think its second hard drive has died. Can you turn off its power switch when I shut it down?"

Hapless2: "Sure, which box is it?

Hapless1: "Ummm... its hostname is **`srv7`**..."

Hapless2: "None of them are labeled!"

Hapless1: "Okay... [**`cat /proc/cpuinfo`**] it's a Pentium 2."

Hapless2: "That narrows it down to five boxes..."

This kind of guessing game can continue for quite some time. Following the ground-breaking research of Murphy, if you guess wrong, it is reasonably certain that you will pick a critical server to drop. My least-favourite twist on this is when the boxes have been labeled—but labeled wrong—or labeled with yellow post-it notes (which fall off as the temperature in the server room increases).

If you are using a Linux box, and it has a CD-ROM drive, why not try ejecting it? Using the **`eject /dev/cdrom`** (or other device name as appropriate) command will make the box spit out its CD tray. It is like telling the real **`srv7`** to put its hand up.

[Cautionary note: Be careful of doing this to machines where the CD-ROM tray is behind a closed door, such as with the Digital Prioris or the IBM NetVista. Like a tractor-pull for plastic components, you *will* find out whether the server door is stronger than the internal tray mechanism of the CD-ROM drive.]

[Disappointing note: Calling eject on a nonremovable drive does not cause the hard drive to eject its platters. Bummer! A hard drive that could unleash a couple of platters at 10,000 revolutions per minute would be an interesting sight.]

Change Default Passwords (and record them for your successor)

Sometimes in one of these computer ghettos, you will stumble across an unexpectedly nice piece of equipment, such as a managed switch or a decent router. The chances are strong that it will have been left in its default configuration, so that any devious member of staff can *telnet* to it, change its configuration, leaving the network even more fouled up.

Your natural inclination should be to change these passwords—even if people do not act maliciously, they can sometimes foul up equipment accidentally. However, because you have been pressed into service as the network admin, remember that the same fate will likely befall another hapless victim one day. As a mark of consideration, record the equipment description, location, serial number, and new password, on paper. If the company has a safe, store it there. If the company has a safety deposit box, store it there. Make sure someone (a manager or director) knows about it. The time you save may be your own.

Do-It-Yourself Router

Perhaps you have identified that the network really ought to be split up—maybe moving testing to its own segment so that the incessant load-testing does not choke the network for everyone. However, requests for budget allocation to buy a router might not actually be fulfilled. It is at times like this that an old Pentium, two network cards, and a copy of the *Linux Router Project* (LRP) can be pressed into service as a cheap router.

The throughput of such a lo-fi router may not match that of a dedicated unit, but it may suffice for a small organization.

For bonus points, you might also consider setting up some firewall rules on the router, so that the next virus-ridden e-mail opened by someone in marketing does not flood the entire network with excess traffic.

Nagios

Network monitoring tools can make a world of difference to your quality of life as a temporary network administrator. Rather than waiting for users to alert you to a downed Internet connection, you can detect and repair problems as they occur. The ability to maintain logs of link downtime can also help support arguments to replace unreliable links.

Nagios[1] is a free network monitoring tool. It provides services such as:

- Monitor if a host is up
- Monitor if key services on a host are up
- Monitor if a host is running services it should not

A Web interface allows easy access to status reports. It can be configured to notify you when problems occur, for example, with an e-mail message. Of course, if the mail server is down, this notification method might not be so useful. Such a situation might be better handled by using the Nagios *Short Message Service* (SMS) messaging component.

Given that you might not have a dedicated *Global System for Mobile Communications* (GSM) modem available for sending these SMS notifications, you might like to investigate the Gnokii project[2]. Ostensibly a project to assist the user in communicating with a mobile phone handset (over data-link cable or infrared), with a capable handset users can initiate sending SMS messages from their handset with Gnokii.

Snort

Intrusion detection might seem a luxury on a network that is struggling to stay operational, but when the price is right (free) and you can spare time to set it up, *Snort* offers a range of features that is surprisingly good. *Snort* can even run without an IP address, making its host computer a fairly difficult target for intruders. The documentation at the *Snort* Website[3] is quite comprehensive, and I recommend it.

Squid

Squid[4] is a popular, free HTTP and FTP proxy server. The simple act of caching banner and button graphics for frequently accessed sites can give an apparent increase in Internet bandwidth. The impression for the end user is that things just get faster, because all those pretty graphics load immediately. You may know it is just a nifty trick, but why let on?

Nmap

One characteristic of chaotic networks is that, like weeds after heavy rain, network services spring up everywhere. Programmers are prime offenders in this respect. But be wary—a service with a security flaw, running on an exposed server, can provide an easy beachhead for crackers (a lesson I learned the hard way).

Nmap[6] is a free network scanner that can assist in finding servers that seem to be running more services than they ought to. It operates in several modes, and offers a range of switches to control its operation.

One of the features that seems more oriented toward people who are scanning networks they are not supposed to is the "Timing policy," specified with the **-T** command-line switch. The options offered here are *Paranoid, Sneaky, Polite, Normal, Aggressive, and Insane*. This feature actually comes in handy if the target of your attentions is heavily laden, or lives at the end of a slow link. If you are in the process of tuning a firewall to detect port scans, *Nmap* offers an excellent test facility too.

Another feature that will likely be helpful is the *Nmap* OS fingerprinting facility. Using a combination of techniques[5], it produces remarkably accurate results for most scans. Combine this result with a port scan and you can build a great picture of which machine has grabbed the wrong IP address (a favorite trick of laptop users: "I didn't know what my IP address was supposed to be, so I picked one.") You also can form a rough network map by OS-fingerprinting every active host on your network.

Immunization

It is a good idea to stay up-to-date on your tetanus shots because occasionally you will nick your hands on the sharp bits of metal found in computer equipment.

Traceroute

When licenses for your VisualRouteAnalyser2000 and TrafficGraphic tools have expired, remember that *traceroute* can be one of the most valuable tools to ascertain exactly where things are going wrong. The only (obvious) word of caution is to be aware that overzealous firewall rules can produce spurious results from *traceroute*.

Tag Cables

The desirability of labeling cables is so obvious that it seems silly to even mention it, but it might not have been standard practice for the sysadmin before you. All the more reason you should do the right thing. Sure, *you* know that the purple cable is the link from **gw-eng** to **gw-test**, but will the next person who has to diagnose network issues?

The other impediment to labeling cables is that the sheer volume of unmarked cables makes the task seem futile. Why bother labeling the new one you have just put in, when there are another 40 unknowns? Take heart—by gradually labeling a few here and there, the cables will gradually get less scary each time. Sometimes it can seem like the labor of Sisyphus, but every little bit helps.

Label Equipment

Post-it notes do not constitute an adequate label for network equipment or servers. You are strongly urged to preserve the sanity of other sysadmins by clearly labeling all equipment, using adhesive labels (in a pinch, the labels for a floppy disk will do).

At a minimum I would suggest that host name and operating system (where appropriate), IP address, and a dire warning against tampering with the unit be included. Bonus points are awarded to people who also maintain an equipment audit and record the details of the unit, plus a list of known services that it is running. Of course these will quickly become outdated, but with a known starting point confusion may be reduced.

Destroy Faulty Cables

After several hours of cable tracing, network-card replacement, checking switch link lights, and so on, it may be that you identify a network problem as being caused by a faulty network cable. It can happen anywhere, and is not necessarily a reflection on the skills of the [acting] sysadmin. (Although if the network cable has clearly been mangled and you should have spotted it with a quick visual inspection, you will probably feel a little silly if the time to locate the fault exceeded two hours).

So you whip a replacement cable out of your secret stash (you should have a secret stash of known-good cables) and voila! Network outage fixed. Now comes the most important duty of all—do not discard the damaged cable anywhere that subsequent admins might find it. On several occasions, damaged cables have been put back in operation, only to cause a repeat of the problem that caused them to be removed from service in the first place. It is not uncommon in server rooms to have an empty box that serves as a rubbish bin, but those unfortunates who come after you may not recognize its role as a waste repository in a time of crisis.

If waste is so abhorred that discarding cables is frowned upon, perhaps you can redo the ends of the cable and vigourously retest. Some even maintain that a long cable run can be split into several shorter runs and reused, because the cable fault is likely to be caused by a single break. I disagree—any cable that has broken in one place is likely to suffer further breaks. Demonstrating this principle to overly frugal managers is sometimes best achieved by ensuring the outcome of the demonstration. I suggest laying the cable through a close-fitting door frame and slamming the door on it a few times prior to testing.

Help Dying Equipment on Its Way

Sometimes it can be difficult to discard equipment. Combine this with the almost pathological frugality common in the small business owner, and you find the most decrepit network gear being nursed along. “I just know this old hub has another few years in it. Sure, a few of the Ethernet ports are stuffed, it overheats on warm days, and looks like it might have a mouse nest in the power supply, but that is no reason to discard it.” Nothing is going to convince the owner of this piece of gear that it is time to “redeploy” it in the rubbish bin.

Sometimes you have to be cruel to be kind. Without wanting to seem too much like the *Bastard Operator from Hell* (BOFH)[7], you may have to help some of this equipment meet its end. It is difficult to identify any one method that fulfills this requirement. My best suggestion is to avoid solutions that leave any externally visible marks (unless they are carbonization marks caused by electrical fault).

You may find that some equipment shows a perverse ability to survive conditions well outside their "recommended operating environment," and nothing short of a sledgehammer will cause those last two operational ports to die. My recommendation here is to do some network reorganization so that the people responsible for the retention of the equipment are directly affected by it. Nothing says "replace me" quite like frequent trips to the server room to toggle the power switch on an ailing hub. It is surprising how fast requisition orders get signed when managers can no longer browse their favorite Websites.

Conclusion

The crisis has passed. Your time as a sysadmin has passed, and you are free to return to your real job. You have acquitted yourself admirably as sysadmin, and you have learned something in the process.

Like the end of a horror movie, you know that it does not really end here. Somewhere, something is waiting to go wrong. Will you be ready the next time?

References

[1] Nagios: **http://www.nagios.org/**

[2] Gnokii project: **http://gnokii.org/**

[3] Snort: **http://www.snort.org/**

[4] Squid: **http://www.squid-cache.org/**

[5] **http://www.insecure.org/nmap/nmap-fingerprinting-article.html**

[6] Nmap: **http://www.insecure.org/nmap/**

[7] BOFH: **http://bofh.ntk.net/Bastard.html**

LOCUM SYADMIN is the nom de guerre of a roving programmer who often seems to find himself in sysdamin roles. Operating in deep secrecy, this elusive creature may sometimes be seen tracing cables and cursing. E-mail: **locum_sysad@yahoo.com**

Letters to the Editor

Ole,

I just finished reading the article about Secure BGP [*Border Gateway Protocol*] by Stephen T Kent. It was very informative and educational with regard to the application and overhead of using the additional BGP attributes and IPSec [*IP Security*]. However, it should be noted that the reliance of a PKI [*Public Key Infrastructure*]-based system, although strong, may also present another possible exploit. If the PKI KDS (*Key Distribution System*) is attacked and subsequently knocked out, including redundant *Key Distribution Engine* (KDE) servers, this may cause serious ramifications to the operation of *Secure BGP* [S-BGP].

Here is a very informative link regarding S-BGP resources for your readers: **http://www.ir.bbn.com/projects/s-bgp**

Also, did you know that the *North American Operators' Group* (NANOG) in conjunction with Cisco engineers recently conducted a BGP vulnerability test? This test confirms that BGP implemented properly is pretty secure in and of itself, without the need for something like S-BGP. The article, titled "BGP Vulnerability Testing: Separating Fact from FUD," was written by Sean Convery and Matthew Franz, Cisco Systems. The article can provide a contrast to the one submitted by Kent and give the technical community both sides of the BGP security issues. Following is the link:
http://www.nanog.org/mtg-0306/pdf/franz.pdf

I thoroughly enjoy IPJ and look forward to each issue. Keep up the great work.

—*Jeffrey J. Sicuranza, Applied Methodologies Inc.*
jsicuran@optonline.net

The author responds:

Ole,

Jeffrey makes a few observations about S-BGP in his letter, and they merit responses.

First, I would hope that the discussion of the security features of S-BGP and their direct derivation from the semantics of BGP was as informative as the discussion of performance aspects of the system. After all, a system with good performance but questionable security is probably a poor candidate to S-BGP routing.

Jeffrey raises the question of whether the reliance of S-BGP on certificates, CRLs [*Certificate Revocation Lists*], and address attestations creates significant vulnerabilities that need to be addressed. This is a fair question, but one which I think we have addressed.

The data that S-BGP stores in repositories is data that changes slowly, and thus the system tolerates unavailability of these repositories fairly well. Note that no router ever accesses these repositories in order to verify a route attestation received in an UPDATE. Instead, each ISP [*Internet Service Provider*] or multihomed subscriber NOC [*Network Operations Center*] accesses the repositories to retrieve this data, process it, and distribute the extracted public keys and authorization data to the routers in its network. We anticipate that this process might occur roughly every 24 hours. Because the information represented by the signed objects in the repositories changes very slowly, this retrieval rate seems appropriate. One would expect that these repositories can be engineered to meet these availability requirements. In the worst case, network operators can choose to keep working with the last set of data that they have successfully retrieved. This works because operators process the data before distributing it to their network, and thus can override expired CRLs, etc. So, I think the answer to Jeffrey's cited concern is that S-BGP is not very vulnerable to attacks against these repositories.

I strongly disagree with the conclusions Jeffrey draws from the BGP vulnerability tests he cites. Numerous incidents of BGP security breaches have been reported over the last few years, so there is no question that BGP, as implemented, deployed, and operated, is insecure. Correct implementation of BGP and improved network operator management practices certainly can reduce BGP vulnerabilities. However, the article in question is hardly a refutation of the wide range of vulnerabilities that exist both in practice and in principle. Much of it focuses on a narrow range of attacks, not broader security concerns.

In addressing broader security concerns, for example, the article argues that proper filtering of routes will mitigate the impact of a compromised router. But we know that such filtering is not feasible for many transit network connections, and route filterers are prone to configuration errors. Reliance on transitive trust (for example, assuming that peers filter routes appropriately) makes BGP intrinsically insecure. Relying on *all* ISP operators to *never* make exploitable errors in configuring their route filters, where such filters can be used, is a fundamentally flawed security approach. S-BGP accounts for the reality that not every ISP will operate its network perfectly, and employs mechanisms to allow other ISPs to detect and reject a wide class of errors (or attacks) that may result from such imperfect operation. Thus I reassert that the security vulnerability characterizations that appear in the S-BGP publications are accurate, not overblown.

As a side note, I find it odd that some critics of S-BGP argue that it fails to account for operational reality, yet they offer alternatives that are based on unrealistic assumptions about network operators acting perfectly!

—Steve Kent, BBN Technologies
kent@bbn.com

Book Review

IP for 3G

IP for 3G, Networking Technologies for Mobile Communications, by David Wisely, Philip Eardley, and Louise Burness, ISBN 0-471-48697-3, John Wiley & Sons, 2002.

I was looking for a book covering mobile communication issues from an IP perspective and IP issues from a mobile communications perspective in order to better clarify details of IP and *third-generation* (3G) convergence. The issue is becoming more and more concrete with the early implementations of 3G networks, so this is a timely book for networking professionals.

Organization

This well-organized textbook helps readers easily understand the "IP-for-3G" issues. It gives a clear vision of that convergence as well as the current snapshot of the recent developments about the subject within the research community. The book is more than an introductory textbook; but readers interested in more technical elaboration can refer to a detailed list of references and further readings given at the end of each chapter.

The book begins with a short chapter that explains the case for IP for 3G. The authors discuss in detail what the term means. They give possible interpretations of IP (Internet, IP Protocol, applications) and their consequent implications on the meaning of IP for 3G. Then they elaborate the IP case within first the "Engineering Reasons for IP for 3G" and then "Economic reasons for IP for 3G" sections.

The second chapter is an introduction to 3G networks. The chapter mostly concerns the core and the access part of 3G networks, skipping the air interface part, because core and access are where IP would make a real difference to the performance and architecture of a 3G network. The chapter reviews briefly the history of 3G developments, from conception to implementation. Then the architecture of *Universal Mobile Telecommunications Service* (UMTS) is introduced, followed by the section where elements of the core network and the architecture of the radio access part are examined. For each part, main functional components such as *Quality of Service* (QoS), mobility management, security, transport, and network management are discussed in detail.

The third chapter discusses the basics of IP and IP networks. Authors give excellent remarks about IP design principles, which are then compared to those of classical telecommunications. Subsequent short sections inform readers about IP addressing schemes, routing, layer behavior, etc. The final section covers the issue of application layer security, which is irrelevant to me for the content of this book. A note: Some of the following chapters require better IP know-how, especially about domain segmentation and intra- and interdomain routing issues. Readers with no prior information are encouraged to refer to other materials before examining the details of, for example, mobility management and QoS.

The fourth chapter is about the multimedia support and session management. First, the concept of session management is introduced. The chapter focuses mainly on the control plane functions of the session management, and the data plane functions are covered in detail in the sixth chapter. The concept of the *Virtual Home Environment* (VHE) is introduced, which forms one of the major requirements of the next-generation mobile system. The authors then review control plane session management protocols, namely H.323 and the *Session Initiation Protocol* (SIP). More discussion is given to SIP, because it is included in the next generation of UMTS standards as the major session management protocol.

The fifth chapter reviews a major problem of the IP-for-3G concept: mobility management. Other key issues of IP such as QoS, IPv6, and session management have always been subject to preceding studies, because those protocols have already been proposed for use in stationary networks. However, the issue of mobility management is a major subject to be investigated for any proper convergence scenario. Personally, I find that this is the biggest challenge of the "long-time-discussed" convergence of IP and mobile communications, and hard work is still ongoing in order to properly resolve the mobility problem. The chapter reviews the basics of mobility such as personal or terminal mobility. From there, macromobility (interdomain or global mobility) and micromobility (intradomain or local mobility) concepts are discussed, followed by proposed protocols for each type of mobility. Mobile IP is examined as the (unique) macromobility protocol. More attention is given to micromobility because it is the most sensitive part of the mobility, under the assumption that 3G BTSs (B nodes) will be simple routers with some extra capabilities. Two variants are discussed, mobile IP schemes, which are based on dynamic tunneling mechanisms, and "per-host forwarding" schemes based on dynamic routing functions. A comparison of major proposals for micromobility management protocols follows.

The sixth chapter considers current IP QoS mechanisms, their operation and capabilities. Those mechanisms created mostly for stationary IP networks may provide a bounded QoS for some "non-real-time" applications, but they are not enough to support any QoS request within the wireless or mobile environment. After giving details of current QoS mechanisms and discussing wireless implications for TCP QoS as well as mobility and wireless issues for *Real Time Protocol* (RTP) QoS, the chapter examines the key elements of QoS and generic features that any prospective QoS mechanism must have. Finally, the authors analyze recent Internet QoS mechanisms such as *Integrated Services* (IntServ), *Differentiated Services* (DiffServ), *Multiprotocol Label Switching* (MPLS), and *Resource Reservation Protocol* (RSVP). The closing section proposes a possible outline solution for how to provide IP QoS for 3G, based on previous work done during the EU BRAIN project.

In the final chapter, the authors summarize all previously given subjects to sketch out the vision of an "All-IP" mobile network. Principles, architecture, routing and mobility issues, QoS, security issues, and interfaces are all discussed to elaborate the generic vision of All-IP networks. Finally, 3G network evolution covering UMTS R4 and R5, and what is beyond 3G, are all discussed.

The book is perfect in the sense that it touches a very hot topic, most of the technical details of which are still in the process of evolving. The authors manage very well the level of details about each subject; they first discuss the overall material before examining details, so readers can obtain a generic but complete view before studying technical details. Each chapter is followed by a comprehensive list of references and further readings, each of them classified by topic. The only fault I find in the book is that SIP should be discussed in more detail.

Recommended

Overall, I would highly recommend this book to any network professional, especially one who is part of any IP-3G convergence process for mobile operators. Still, data network professionals can glean much from the book, because the aim is to carry—a little differently—the same old data, whether or not it contains multimedia, voice, or standard data information.

—Dr. K. Murat Eksioglu, RT.NET, Turkey
murat.eksioglu@o2.net.tr

[Ed.: A version of this review was previously published in the October 2003 issue of *IEEE Communications Magazine* (Vol. 41, No. 10). Used with permission.]

Fragments

Tim Berners-Lee Knighted by Queen Elizabeth

31 December 2003 — Tim Berners-Lee, the inventor of the World Wide Web and director of the *World Wide Web Consortium* (W3C), will be made a *Knight Commander, Order of the British Empire* (KBE) by Queen Elizabeth. This was announced earlier today by Buckingham Palace as part of the 2004 New Year's Honours list.

The rank of Knight Commander is the second most senior rank of the Order of the British Empire, one of the Orders of Chivalry awarded. Berners-Lee, 48, a British citizen who lives in the United States, is being knighted in recognition of his "services to the global development of the Internet" through the invention of the World Wide Web.

"This is an honor which applies to the whole Web development community, and to the inventors and developers of the Internet, whose work made the Web possible," stated Berners-Lee. "I accept this as an endorsement of the spirit of the Web; of building it in a decentralized way; of making best efforts to keep it open and fair; and of ensuring its fundamental technologies are available to all for broad use and innovation, and without having to pay licensing fees."

"By recognizing the Web in such a significant way, it also makes clear the responsibility its creators and users share," he continued. "Information technology changes the world, and as a result, its practitioners cannot be disconnected from its technical and societal impacts. Rather, we share a responsibility to make this work for the common good, and to take into account the diverse populations it serves." For more information see:

http://www.w3c.org/2003/12/timbl_knighted

SECSAC Publishes DNS Report

The *Security and Stability Advisory Committee* (SECSAC) has published a report entitled "DNS Infrastructure Recommendation." For details see:

http://www.icann.org/committees/security/dns-recommendation-01nov03.htm

Coordination, not Governance says ISOC re WSIS

The *Internet Society* (ISOC) published the following text at the *World Summit on the Information Society* (WSIS 2003) which was held in Geneva in early December, 2003:

ISOC is a global not-for-profit membership organisation founded in 1991 to provide leadership in Internet-related standards, education, and policy issues. We are dedicated to ensuring the open development, evolution and use of the Internet for the benefit of people throughout the world. Our education initiatives, for example, have helped bring Internet connectivity to virtually all developing countries over the last 12 years.

ISOC is the organisational home of the *Internet Engineering Task Force* (IETF)—an open consensus-based group responsible for defining Internet protocols and standards. Through our participation in WSIS 2003 we aim to increase understanding and awareness of what is important in order to develop and maintain the Internet's stability, open nature and global reach.

The Internet has come of Age

In many countries, the Internet has become a mass medium. This has brought with it reflexive pressure on policy makers to regulate it as if it were radio, television, or other mass media. While Governments naturally seek to address their citizens' interests regarding online privacy, spam, Internet security, intellectual property protection, the price of Internet access, and the digital divide, our position is that better use of technology, and broad participation in today's Internet coordination processes, not Government regulation, are the most effective and appropriate ways to satisfy these concerns.

The biggest barrier to the Internet fulfilling its immense potential could turn out to be misinformed and inappropriate intervention in the way in which the Internet's technologies, resources and policies are developed, deployed and coordinated. The Internet Society can help provide guidance here.

What is the nature of the Internet?

The Internet is a modern distributed communications medium. No one is in charge of the Internet and yet everyone is in charge. Unlike the antiquated system of national telephone network monopolies, the global Internet consists of tens of thousands of interconnected networks run by Internet Service Providers, individual companies, universities, Governments, and other institutions. Some of these are global in scope, others regional or local. Hundreds of different organisations and thousands of different companies make decisions every year that contribute to how the Internet develops.

These varied entities, together with the users of the Internet and the developers of Internet technologies and applications, have specific needs for coordination. Collaborative processes that are critical for the future stability and evolution of the Internet, and which should not be modified arbitrarily or abruptly, satisfy these needs.

Coordination, not Governance

It is misleading to use the term "Internet Governance" when the Internet is clearly not a single entity to govern. It is more useful to refer to "Internet Coordination." The multiple facets of the Internet require different types of coordination, each calling for specific competencies and sensitivities to balance the needs of the Internet user community globally and locally. Specific Internet Coordination activities are taking place globally at three levels:

- Coordination of the definition of Internet standards
- Coordination of the availability and assignment of Internet resources
- Coordination of the policies preventing misuse of the Internet

This coordination is best performed by the existing set of organisations using proven processes. Because of the diverse nature of these activities, it is unrealistic to expect a single body— Government or otherwise—to take on all these roles effectively.

Coordinating Internet standards

The IETF under the umbrella of the Internet Society, is one of the oldest and most successful Internet coordination processes. Other organisations are also involved in Internet-related standards, including the IEEE, the W3C and the ITU.

Many of the protocols at the heart of today's Internet (for example, TCP, IP, HTTP, FTP, SMTP, Telnet, PPP, POP3, the DNS protocol etc.) were developed through IETF standards activities. The results of the IETF are well engineered and practical open protocol standards that are trusted and open to global implementation with little or no licensing restrictions—they are freely available on the Internet, without cost, to everyone.

The strength of the IETF process lies in its unique culture and talented global community of network designers, network operators, service providers, equipment vendors, and researchers. They all openly contribute their individual technical experience and engineering wisdom in an environment that fosters innovation and the open exchange of ideas. This process, which is open to anyone, helps quickly identify and articulate problems of common interest. It also helps build the trust required to make the further investments necessary for a protocol to be usefully implemented and deployed. Ultimately, however, it is the Internet users themselves that determine whether or not a protocol is valuable and useful enough for widespread use. Here the IETF track record of producing useful, widely deployed protocols is unrivaled.

Coordinating Internet resources: The Internet Registry System

There has always been a need to manage the allocation of Internet resources such as the unique addresses that identify devices connected to the Internet (IP addresses), generic top-level domain names (for example, **`.org`**), country code top-level domain names (for example, **`.ch`**), domain names (such as **`www.isoc.org`**), and the systems that translate domain names into IP addresses (for example, the *Domain Name System* or DNS).

This coordination activity has been handled by long-standing, not-for-profit membership organisations such as the *Regional Internet Registries* (RIRs) and *top-level domain* (TLD) registries.

More recently, coordination at a global level has been supported by the *Internet Corporation for Assigned Names and Numbers* (ICANN). Established in 1998, ICANN is also a not-for-profit organisation. Business, technical, non-commercial, academic, governmental and end-user communities participate in ICANN.

These organisations are a meeting point for bottom-up, consensual, industrial self-regulation by the groups and individuals that use their services and resources.

Coordinating policies preventing misuse of the Internet

As we have seen, organisations such as the RIRs, TLD registries, ICANN and the IETF all have very specific roles. It is neither within their charters, nor within their capabilities, to take on responsibility for all areas of Internet Coordination—particularly that of preventing inappropriate use of the Internet. For example, areas such as "cyber crime" (for example, fraud and child pornography) require coordinated global attention by lawmakers—and not by those responsible for the equitable coordination of the underlying Internet infrastructure. Security matters also need to be addressed by organisations providing Internet access (not only by standards developers), and intellectual property iss ues may best be handled by organisations such as the *World Intellectual Property Organization* (WIPO).

In discussions about these broader Internet policy issues there is cooperation between all the organisations mentioned above. ICANN for example works with WIPO to implement its *Uniform Domain Name Dispute Resolution Policy* (UDRP). And the Internet Society, with technical advice from the IETF, works with Governments and policy makers to explain the effects and possibilities of new Internet technologies.

The way forward: Make your voice heard

Existing consensus-based processes have given us the Internet and have successfully coordinated its phenomenal growth: thousands of new networks, new policy procedures, new top-level domain names, new protocols etc. All of them constantly balance the needs and stability of today's Internet with future demands.

An open debate is now needed to move towards common, globally acceptable policies, processes and technologies to prevent misuse of the Internet. Governments have a vital role to play here as a concerted effort on the part of the Internet community, non-governmental organisations and Governments can help strengthen and extend today's successful coordination processes.

The successful continued development of the Internet for the benefit of everyone can be ensured by participation in these proven processes rather than by attempting to create new untested mechanisms that are inappropriate to the unique characteristics of the Internet.

The Internet Society remains dedicated to providing information and orientation about Internet structures and processes. We encourage broad participation in the activities of each of the organisations involved in Internet coordination. For more information on ISOC, visit: **www.isoc.org**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

March 2004 Volume 7, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

The operational stability of the global Internet (or any network based on TCP/IP technology) is in large part the result of a carefully configured routing system. Routing continues to be one of the most complex topics in Internet engineering. In our first article, Russ White describes some mechanisms for the design of large-scale, stable routing systems. The article is entitled "High Availability in Routing."

Security continues to be a high-priority item in computer networks and in society in general. One aspect of security is the identification system by which an individual is given authorized access to a particular facility, be it physical or virtual. Edgar Danielyan gives us an overview of one key element of identification, namely *biometrics*.

The Internet is "going where no network has gone before." The *National Aeronautics and Space Administration* (NASA) has been working on the *Interplanetary Internet Project* (**http://www.ipnsig.org/**). We hope to bring you an in-depth article about this project in a future issue. An important demonstration of this system took place recently. To quote from the press release:

> "A pioneering demonstration of communications between NASA's Mars Exploration Rover *Spirit* and the *European Space Agency* (ESA) *Mars Express* orbiter has succeeded. On February 6, 2004, while Mars Express was flying over the area Spirit was examining, the orbiter transferred commands from Earth to the rover and relayed data from the robotic explorer back to Earth. The commands for the rover were transferred from Spirit's operations team at NASA's *Jet Propulsion Laboratory* (JPL), in Pasadena, California, to ESA's European Space Operations Centre in Darmstadt, Germany, where they were translated into commands for Mars Express. The translated commands were transmitted to Mars Express, which used them to successfully command Spirit. Spirit used its ultra-high frequency antenna to transit telemetry information to Mars Express. The orbiter relayed the data back to JPL, via the European Space Operations Centre."

We often receive requests for back issues of IPJ. Although we cannot provide paper copies, all of our previously published editions are available in both PDF and HTML format from the IPJ Website: **www.cisco.com/ipj**.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

High Availability in Routing

by Russ White, Cisco Systems

A network is a complex system of interacting pieces, as anyone who has ever worked with a large-scale network "in the wild" can tell you. So, when businesses begin asking for a network that can converge in something under 1 second, especially in a large network, network engineers begin to scratch their heads, and wonder what their counterparts in the business world are thinking about. Just about everyone in the network engineering business knows scale and speed are, generally speaking, contradictory goals. The faster a network converges, the less stable it is likely to be; fast reactions to changes in the network topology tend to create positive feedback loops that result in a network that simply will not converge.

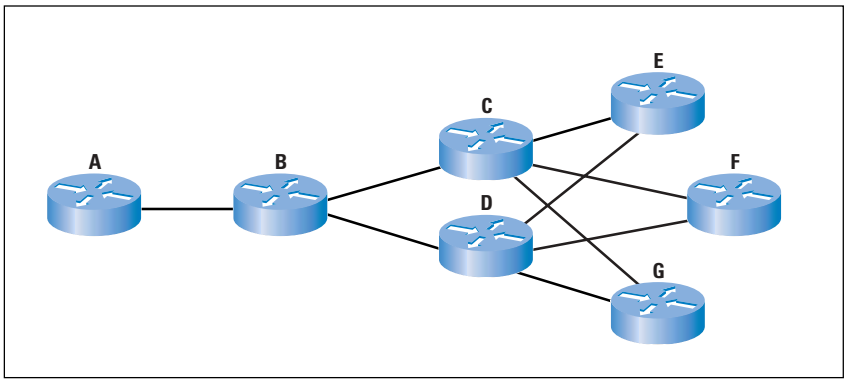
But recent experience has shown that subsecond convergence in a network—even a large network in the wild—is definitely possible. How do we go about building a large-scale network that can converge in times that were, before recently, considered impossible, or improbable, at best? We approach the problem the same way network systems, themselves, are approached. We break the problem down into smaller pieces, and try to solve each piece individually. When we have solved each of the smaller pieces, we recombine them, and see what needs to be adjusted to make it all work together properly.

What pieces of a network do we need to be concerned about when considering subsecond (fast) convergence? Generally, we are concerned with the physical layer (how fast can a down link be detected?), routing protocol convergence (how fast can a routing protocol react to the topology change?), and finally, forwarding (how fast can the forwarding engine on each router in the network adjust to the new paths calculated by the routing protocol?). This article focuses on routing protocols convergence, with some discussion of fast down detection as well, specifically the interior gateway protocols, *Enhanced Interior Gateway Routing Protocol* (EIGRP), *Intermediate System-to-Intermediate System* (IS-IS), and *Open Shortest Path First* (OSPF).

Network Meltdowns

Before beginning to work on a network so it will converge quickly, we need to set some realistic expectations. As mentioned previously, a routing protocol configured to react very quickly to changes in network topology tends to develop positive feedback loops, which result in a network that will not converge at all. Using the following example, consider how a single problem can produce feedback that causes a failure to cascade through the network.

Figure 1: Positive Feedback Loops in a Network



Suppose the link between routers D and G flaps, meaning that it cycles between the down and up states slow enough for a routing adjacency to be formed across the link, or for the new link to be advertised as part of the topology, but too quickly for the link to actually be used. In this situation, the adjacency (or neighbor relationship) between routers D and G forms and tears down as quickly as the routing protocol will allow.

While this is occurring, the routing information at routers E, F, and G is changing as quickly as the adjacency between D and G can form and tear down. This change in routing information is, in turn, passed on to C, which then must process it as fast as it possibly can. It is possible that the routing information presented to router C will overcome the ability of its processor to process the information, causing router C to fail, or drop its neighbor adjacencies.

At the same time, the constantly changing routing information at router B will also cause problems, possibly causing it to periodically drop its adjacencies, specifically with routers C and D. At this point, if the routers B, C, and D are all three consuming a large amount of memory and processing power adjusting to apparent topology changes because of changing adjacency states, the flapping link between routers D and G, which originally caused the problem, can be removed from the network, and the routing protocol will still not converge. This is what network engineers consider a classic *meltdown* in the routing system.

Solving the Meltdown

Typically, when a network engineer faces a network in this condition, the first step is to simply remove routing information from the system until the network "settles." This typically involves removing parallel (redundant) links from the view that the routing protocol has of the topology until the routing protocol converges. At this point, the network would be examined, routers reloaded as needed, and the parallel links brought back up. The network design might then be reviewed, in an attempt to prevent recurrence of a meltdown.

Routing protocol designers and developers would also like to move the point at which a routing protocol "melts" as far along the curve of network design as possible.

Of course, it is impossible to prevent all network meltdowns through protocol design; there are limits in any system where the implementation steps outside the "state machine," and the system will simply fail. But how would a routing protocol designer work around this sort of a problem in the protocol itself? The answer is actually very simple: Slow down.

The main problem here, from a protocol designer's point of view, is that routers D and G are simply reacting too fast to the changing topology. If they were to react more slowly, the network would not fall into this positive feedback loop, and the network would not melt. And, in fact, slowing down is really quite simple. Various methods of slowing down include:

- Not reporting all interface transitions from the physical layer up to the routing protocol. This is called *debouncing* the interface; most interface types wait some number of milliseconds before reporting a change in the interface state.
- Slow neighbor down timers. For instance, the amount of time a router waits without hearing from a given neighbor before declaring that a neighbor has failed is generally on the order of tens of seconds in most routing protocols. The dead timer does not impact down-neighbor detection on point-to-point links, because when the interface fails, the neighbor is assumed to be down, but there are other "slow-down" timers here, as well.
- Slow down the distribution of information about topology changes.
- Slow down the time within which the routing protocol reacts to information about topology changes.

All four of these methods are typically used in routing protocols design and implementation to provide stability within a routing system. For instance:

- In IS-IS, a timer regulates how often an intermediate system (router) may originate new routing information, and how often a router may run the *shortest path first* (SPF) algorithm used to calculate the best paths through the network.
- In OSPF, similar timers regulate the rate at which topology information can be transmitted, and how often the shorter path first algorithm may be run.
- In EIGRP, the simple rule: "no route may be advertised until it is installed in the local routing table" dampens the speed at which routing information is propagated through the network, and routing information is also paced when being transmitted through the network based on the bandwidth between two routers.

It seems like the simplest place to look when trying to decrease the time a routing protocol requires to converge, then, is at these sorts of timers. Reduce the amount of time an interface waits before reporting the transition to a down state, reduce the amount of time a router must wait before advertising topology information, etc. But when we consider implementing such changes, we remove much of the stability we have all come to expect in routing systems—the size a network can be built without melting down decreases below an acceptable threshold, even with modern processors, more memory, and implementation improvements in place.

There is another place to attack this problem: the frequency of changes within the network. This is the same concept—speed—from a different angle. How does looking at it from a different angle help us? By allowing us to see that it is not the speed of the network changes that causes the positive feedback loop, but rather how often the changes take place. If we could report the changes quickly when they occur slowly, and report them more slowly when they occur quickly, or if we could just not report some events at all, routing could converge much faster, and still provide the stability we expect.

The two options we want to examine, then, are not reporting every event, and slowing down as the network speeds up. First we will discuss these two options, and then discuss speeding up the reporting of network events, which plays a large role in decreasing convergence times.

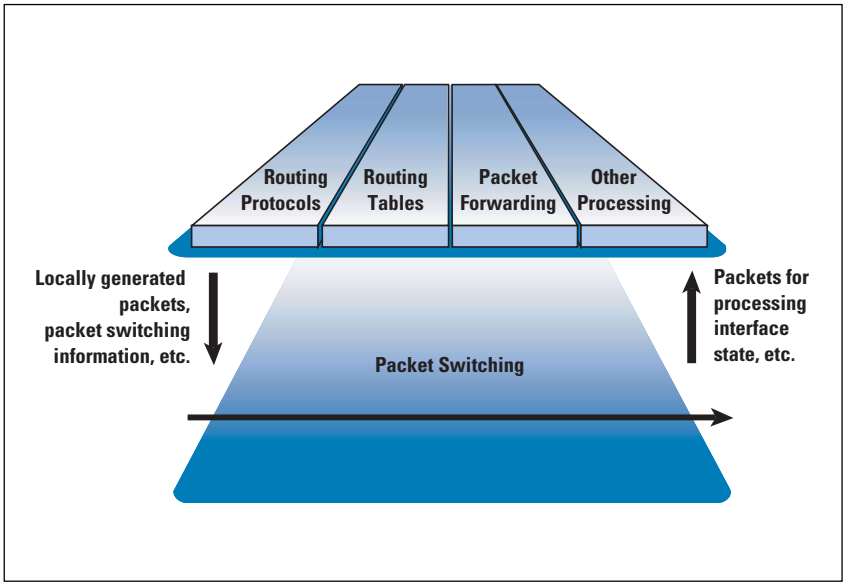
Do Not Report Everything You See (NSF and GR)

It sounds simple just to say that a router should not report every event within the network it is aware of, but it becomes more complicated as we consider the issues involved. What we need to do is sort out which events are important, in some sense, and which are not. For instance, if a router loses contact with an adjacent router because the adjacent router restarted for some reason, do not report the resulting change in topology until you are certain the neighbor is not coming back.

But the classic questions follow: How long do you wait before deciding the problem is real? And what happens to traffic you would normally forward to that neighbor while you are waiting? Finally, how do you reconnect in a way that allows the network to continue operating correctly? A technology recently incorporated in routing protocols called *Graceful Restart* (GR), combined with another technology called *Non-Stop Forwarding* (NSF), can combine to answer these questions.

Let's start at the bottom of the *Open Systems Interconnection* (OSI) model, at the physical and data link layers, and discuss the second question, what happens to traffic that would normally be forwarded while a router is restarting? Normally, this traffic would be dropped, and any applications impacted would need to retransmit lost data. How could we prevent this? We can take advantage of the separation between the control plane and the forwarding plane in a large number of modern routers.

Figure 2: Control and Data Plane Interaction in a Router



In some routers, such as the Cisco® 12000, 10000, 7600, and others, the actual switching, or forwarding, of packets is performed by different processors and physical circuitry than the control plane processes run on (such as routing protocol processes, routing table calculation, and other processes). Therefore, if the control plane fails or restarts for any reason, the data plane could continue forwarding traffic based on the last known good information.

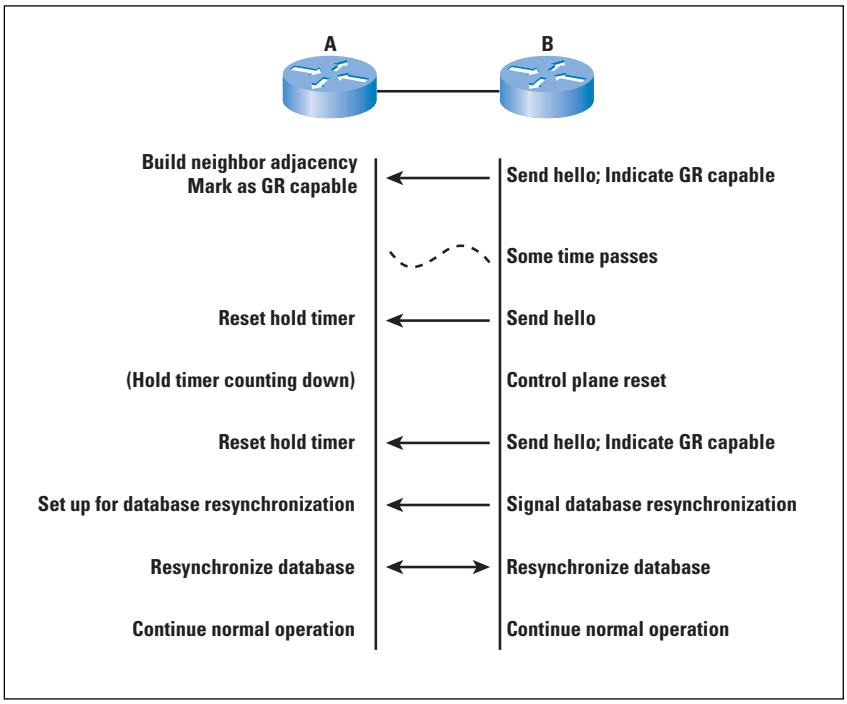
NSF, implemented through *Stateful Switchover* (SSO) and *Stateful Switchover+* (SSO+) in Cisco products, allows this continuous forwarding, regardless of the state of the control plane, to take place. Normally, when the control plane resets, it sends a signal to the data plane that it should clear its tables out, and reset, as well. With NSF enabled, this signal from the control plane simply acts as a signal to mark the current data as stale, and to begin aging the information out.

Now we need to be able to bring the control plane back up, resynchronize the routing protocol databases, and rebuild the routing table, all without disturbing the packets still being switched by the data plane on the router. This is accomplished through GR. GR starts by assuming two critical things:

- The normal hold times are acceptable, within this network environment, for reporting a network event or topology change. In other words, if a router's control plane fails, the event wouldn't be reported until the routing protocol's default hold or dead timers expire, whether or not GR is configured.
- The control plane on the router can reload and begin processing data within the hold or dead time of the routing protocol.

Let's examine how, in principle, GR works, so we can put these two requirements into context, and understand where GR is best deployed in a live network. Consider the following chart to understand how GR works between two peers of any generic routing protocol.

Figure 3: The Process of Graceful Restart



When two routers begin forming an adjacency (or neighbor relationship, or begin peering, depending on which routing protocol is being run between them), they exchange some form of signaling noting that they are capable of understanding GR signaling, and responding to it correctly.

[Note that this does not imply the router is GR-capable, only that it can support a neighboring router performing a GR. For instance, the Cisco 7200 supports switching modes only where the control and data planes are not cleanly separated, so it cannot fully support GR. It can, however, support the signaling necessary for a neighboring router to gracefully restart.]

Assume some time passes, and router B is transmitting Hello packets to router A normally, on a periodic basis. Each time router A receives one of these Hello (or *keepalive*) packets, it resets the hold, or dead, timer on router B, indicating that it should wait that amount of time before declaring router B down if it stops receiving Hellos. Now, at some point, after sending a Hello packet, the router B control plane resets. While the control plane is down, the router A hold timer is still counting down; the routing protocol does not reset the session. This is, in fact, normal routing protocol operation, which normally results in the packets forwarded by router A toward router B to be dropped. Because router B is NSF-capable, however, its data plane is still forwarding this traffic to the correct destination, even though the control plane is down.

If the router B control plane does not come back up within the dead or hold timer allowed by the routing protocol, router A declares the adjacency down, and begins routing around router B. This explains why the router B control plane must come back up within the hold interval of the routing protocol, one of the two assumptions we outlined GR as making at the beginning of this section. For this case, we assume that the router B control plane comes back up before the router A hold timer expires, and router B sends a Hello with no information other than indicating it is restarting.

When router A receives this Hello, it acts as though it has received a normal Hello, and simply keeps its adjacency with router B up. In other words, although router B may not know what the network it is connected to looks like at this point, router A does not report this failure to the rest of the network. Convergence time is, from a network standpoint, effectively reduced to 0.

When the router B control plane completes its reset, it then signals router A to begin resynchronizing their databases. The two routers then use some method specific to each protocol to resynchronize their databases, and begin operating normally, in a stable condition once again.

Slow Down When the Network Speeds Up

The second option we discussed originally was to attack the problem by reducing the frequency, rather than the number, of updates. What we want to do is to slow down the reporting of events when they occur more frequently (or when they occur rapidly), and speed up the reporting of events when they occur less frequently (or when they occur slowly). This is possible through a series of features built into Cisco IOS® Software within the last year or two, applying the concept of the *exponential timers*.

An exponential timer changes the amount of delay between an event occurring and the reporting of that event by the frequency at which the event occurs—possibly not reporting the event at all, in some situations. Two implementations of exponential timers are *exponential backoffs* and *dampening*. Let's examine each of these individually, and then consider where they are implemented in Cisco IOS Software.

Exponential Backoffs

Consider the following figure to examine how exponential backoff works.

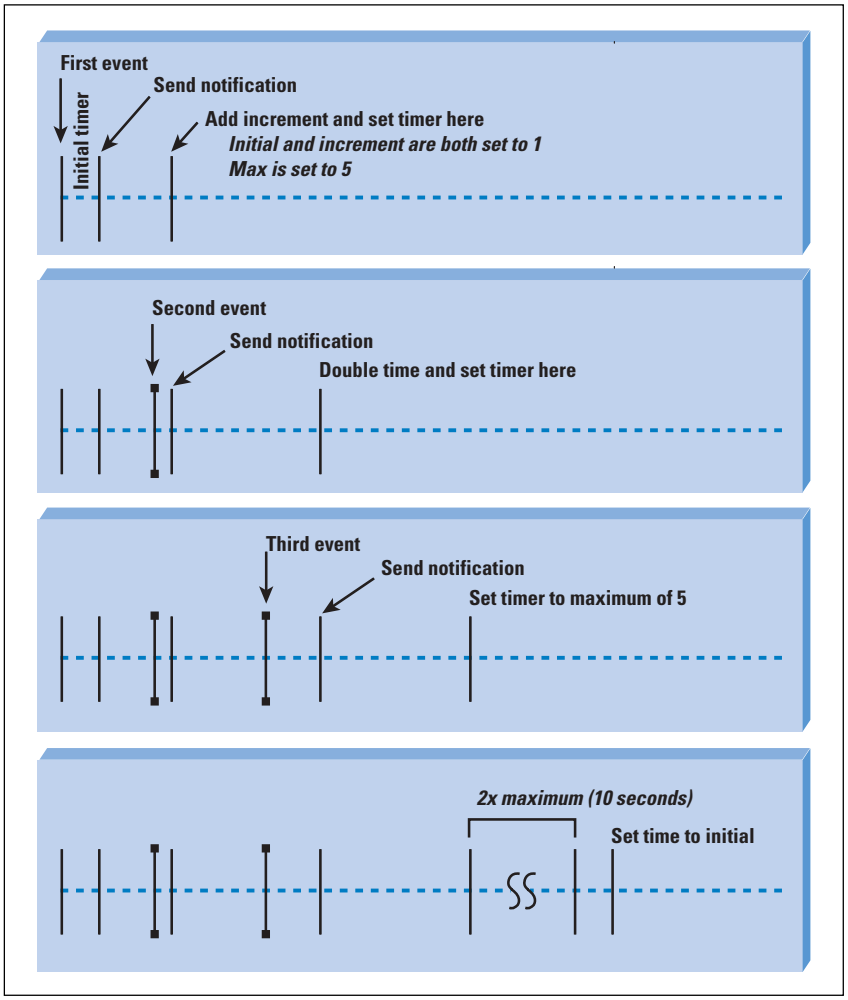


Figure 4: Exponential Backoff

When the first event occurs, a timer is set to the initial time, 1 second in this case, meaning that the router waits for one second before notifying other routers in the network about the event. When the notification is sent, the router adds the initial timer to the increment, and sets a timer for this period. We call this timer the *backoff timer*.

When the second event occurs, the backoff timer is still running; the router waits until this timer expires to send the notification about this event occurring. When this notification is sent, the backoff timer is set to twice the previous setting or the maximum backoff time, whichever one is shorter. In this case, doubling the backoff timer results in 4 seconds, so it is set to 4 seconds.

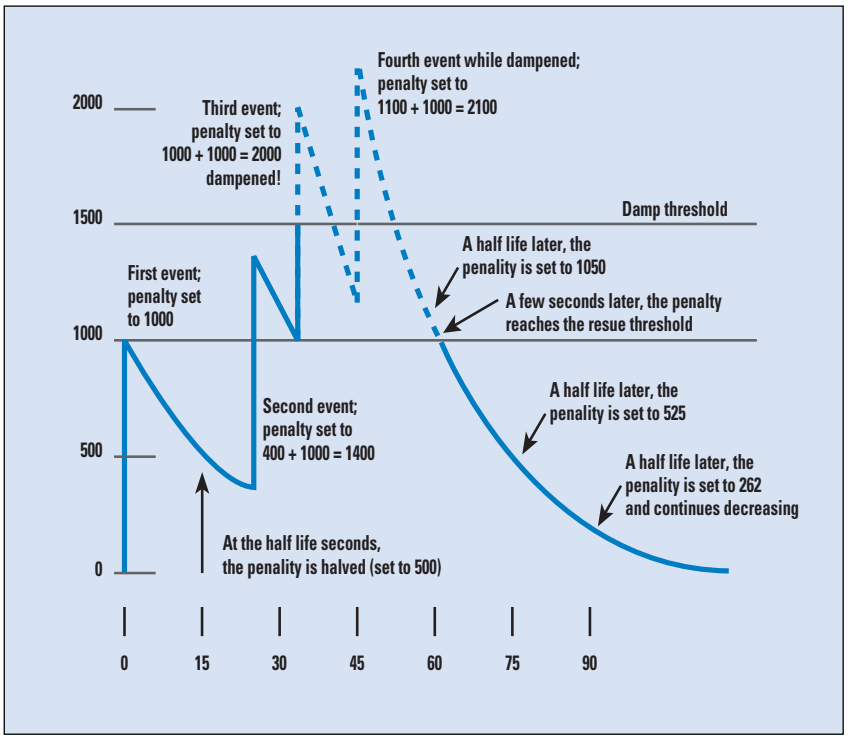
When the third event occurs, the backoff timer is still running; the router waits until the timer expires before sending any notification of the event occurring. Again, the timer is doubled, this time to 8 seconds, and compared to the maximum time, which is 5 seconds. The shorter of the two times is taken, so the backoff timer is now set for 5 seconds.

At this point, any future events will be reported only at 5-second intervals, as long as one event occurs at least every 5 seconds. If no events occur for an interval of 10 seconds, the timers are all reset to their initial condition, so the initial timer is set to 1 second, and the backoff timer is not set at all.

Dampening

Dampening, or damping, is also an exponential backoff mechanism similar to the exponential backoff algorithm we examined previously. The primary difference is that dampening is applied to events that have a Boolean component; a route that is either advertised or withdrawn, an interface that is either up or down, etc. Exponential backoff simply deals with events in general, whereas dampening adds value based on the type of event, as well as the frequency at which the event occurs. Consider the following figure to understand dampening.

Figure 5: Dampening Over Time



In dampening, the desirability of reporting an event is set using the *penalty*; the higher the penalty applied to a given item, such as a route or an interface, the less desirable it is to advertise changes in the state of that item. Dampening always leaves the item in the "off," or "down," state, when it stops reporting state changes; this is called the *dampened* state. A penalty is normally added when transitioning from "down" to "up" in most dampening systems.

Here, we start at time 0, with a penalty of 0; when the first event occurs, a penalty of 1000 is added, making the total penalty 1000. As time passes without another event occurring, the penalty is decreased, based on the *half life*. Each time the half life passes, in this case 15 seconds, the current penalty is halved, so after 15 seconds, the half life is set to 500.

A few seconds later, while the penalty is still decreasing, the second event occurs; 1000 is added to the current penalty, making the total penalty 1400. Again, as time passes, the penalty decays exponentially, reaching 1000 before the third event occurs. When the third event occurs, 1000 is again added to the total penalty, so it reaches 2000—which is above the *damp threshold*, so future events are dampened by simply leaving the interface or route in the down state.

Again, as time passes, the penalty is cut in half for each passing half life, reaching 1100 before the fourth event occurs. When the fourth event occurs, 1000 is again added, making the penalty 2100, and leaving us in the dampened state until the penalty can be reduced again. Over time, the penalty finally drops to 1000 (at around 60 seconds in the example), which is the *reuse threshold*. At this point, state changes in the item being tracked are once again reported as they occur, unless the penalty reaches the dampening threshold at some future point.

So, dampening reacts to events by simply not reporting events if they occur too frequently, whereas exponential backoff reacts to events by reporting each event that occurs, but slowing down the reporting of events as they occur more frequently.

Speeding Up the Reporting of Events

When we have some methods in place to prevent a network meltdown when events occur, we can consider ways to discover events faster. Primarily, these techniques are used in conjunction with exponential backoff and dampening.

There are two ways to detect a down neighbor or link: *polling* and *event driven*. We will briefly discuss each of these, and some various techniques available in both cases.

Polling

One method commonly used for detecting a link or adjacency failure is polling, or periodically sending Hello packets to the adjacent device, and expecting a periodic Hello packet in return. The speed at which Hello packets are transmitted and the number of Hello packets missed before declaring a link or adjacency as failed are the two determining factors in the speed at which polling can discover a failed link or device.

Normally, a neighbor or link is declared down if three Hello packets are lost, meaning that the hold time, or the dead time, will always be about three times the Hello time, or polling interval. Normally, for Layer 2 links and routing protocols, the Hello interval is measured in seconds. For instance:

- EIGRP running over a point-to-point link sends one Hello every 5 seconds, and declares a neighbor down if no Hellos are heard for 15 seconds.
- EIGRP running over a lower-speed point-to-multipoint link sends one Hello every 60 seconds, and declares a neighbor down if no Hellos are received in 180 seconds.

- OSPF normally sends a Hello every 10 seconds, and declares a neighbor down if no Hellos are heard for 40 seconds.
- *Frame Relay Local Management Interface* (LMI) messages, the equivalent of a Hello, are transmitted every 10 seconds. If an LMI is not received in 30 seconds, the circuit is assumed to have failed.
- *High-Level Data Link Control* (HDLC) keepalive messages are transmitted every 10 seconds. If a keepalive message is not received within 30 seconds, the circuit is assumed to have failed.

Fast Hellos can decrease these timers to Hello intervals on the order of 300 milliseconds, and dead timers of around 1 second.

The primary problem with fast Hellos is scaling, particularly in receiving and processing fast Hellos from a large number of neighboring routers. For instance, if a router has 1000 neighbors and is using a Hello interval of 330 milliseconds, the router has to be able to receive and process 3000 Hellos per second and send 1000 Hellos per second. Timers in this range leave little room for processes that consume a router processor for long periods of time, short-term packet loss on a link due to congestion, and other factors.

Event Driven

Rather than polling at a fast interval, event-driven notifications rely on devices within the network that can sense the state of a link through lower layers (electrical, electronic, or optical state) to notify the routers attached to the link when the link has failed. SONET links are probably the best example of media with built-in capabilities for sensing link failures and notifying attached devices. This Tech Note on Cisco Online:

`http://www.cisco.com/en/US/tech/tk482/tk607/technologies_tech_note09186a0080094522.shtml`

... provides information about SONET alarms. There are also techniques that can be used to speed up the reporting of failed links in Frame Relay circuits, and techniques are being developed for allowing switches to notify devices attached to an Ethernet VLAN about a loss of connection to an attached device.

Implementations

Now that we have discussed what exponential backoff and dampening are, we can consider how they are implemented, and how their implementation helps you build highly available networks (through fast convergence) without risking major network instability along the way. We start by examining where dampening is implemented, and then follow that with a discussion about where exponential backoff is implemented. These sections do not provide a great deal of detail on the implementation of these features; vendor documentation and other sources of information (such as the forthcoming book *Designing to Scale*) should be consulted for technical details.

Dampening

Dampening is currently implemented in two places:

- *Border Gateway Protocol* (BGP) route flap dampening
- Interface dampening

BGP route flap dampening is a well-known technology, deployed in the Internet on a wide scale to increase the stability of the Internet routing table.

Interface dampening allows the network engineer to prevent rapidly flapping interfaces from having a wide-ranging impact on the entire network. When an interface fails and comes back up numerous times within a short time period, the interface is placed in the down state from an IP perspective, and not advertised within routing protocols, or used for forwarding packets.

It is important to note that the interface is allowed to change states freely at Layer 2; an interface that continues to change state rapidly continues to accumulate penalties, and continues to show down to the IP subsystem.

Exponential Backoff

Exponential backoff is implemented in several places in link state protocols at this point, including:

- The ability to exponentially back off the amount of time between a change in the network topology being detected and the transmission of a link state packet being transmitted to report the change; exponential backoff has been applied to the link state generation timer.
- The ability to exponentially back off the amount of time between receiving a link state packet reporting a change in the network topology, and running SPF to recalculate the path to each reachable destination in the network; exponential backoff has been applied to the SPF timer.

Fast Hellos

Each routing protocol has a different limit on how Fast Hellos can be transmitted and how often they must be received for a neighbor to be considered alive. OSPF and IS-IS have both implemented the fastest Hellos, with a minimum of 330 millisecond Hellos, and a dead interval of 1 second.

EIGRP can run with Hellos as fast as one per second, with a 3-second dead time. BGP can use similar timers, with a keepalive interval of 1 second.

Caution should be used when configuring Fast Hellos on a network. Congestion, high processor use, and other problems can cause false down indications that may cause higher rates of network failure than would normally occur.

Deploying GR and Fast Convergence Technologies

We now have a full range of options we can use to improve network availability, including GR and NSF, event dampening, and fast convergence techniques. How can we deploy these in a real network to improve network uptime? Generally, the technologies can be placed in one of three categories:

- *Fast reaction to node or link failure, to route around the failure.* We use Layer 2 techniques and Fast Hellos to quickly determine when an adjacent node, or a link to that node, has failed.
- *Slow reaction to node of link failure, combined with routing through the failure.* We rely on moderate speed reactions to node failures to allow resynchronization of routing data while forwarding of traffic continues.
- *Fast recalculation of the best path when a topology change has been reported.*

As we can see, the first two are complementary; we could not deploy both of them in the same place in the network. The third one, fast recalculation, can be deployed with either (or both) fast reaction and slow reaction techniques to increase network availability. The primary question then becomes: which of these two techniques do you deploy in your network, and where?

The basic premise behind making this design decision follows:

- If there is a fast, online backup available to a node or a link, it probably makes more sense to route around any problems that occur as rapidly as possible.
- If any existing backup is going to take a good deal of time to bring online, or there is no backup path (such as a single homed remote office, or a remote office with dial backup), it probably makes more sense to route through any problems.

In general, then, we want to deploy techniques that improve network convergence time everywhere—techniques that bring down the time a network is down when a failure occurs, is detected, and a new path calculated. At the same time, we want to evaluate each point in the network we would like to protect from failure, and determine the best means to protect that point of failure: redundancy with fast down detection, GR, or NSF.

Fast, stable networks are possible with today's techniques in routing; some large networks, with several hundred routers, measure their convergence times in the milliseconds, with 1 second as their outside convergence goal.

RUSS WHITE is on the Cisco Systems Routing DNA Team in Research Triangle Park, North Carolina, specializing in the deployment and architecture of routing protocols. He has coauthored books on routing protocols, network design, and router architecture, regularly speaks at the Cisco Networkers conference, and is active in the Internet Engineering Task Force. Russ can be reached at **riw@cisco.com**. This article offers a high-level overview of material covered in depth in a forthcoming network design book, *Designing to Scale,* being published through Cisco Press.

The Lures of Biometrics

by Edgar Danielyan, Danielyan Consulting LLP

This article introduces biometrics and discusses some of the complex issues associated with use of biometrics for identification and authentication of individuals and its impact on both standalone and networked information systems, as well as on physical security. The agenda is not to show whether biometrics is your best investment or a useless thing—these two polar viewpoints share the same quality of being oversimplifications, to say the least. It also certainly does not purport or try to tell everything there is to tell about biometrics or its applications. Legal and social implications of biometrics are also not discussed in this article because these would differ considerably, depending on the legislation and cultural traditions of countries concerned; we also do not consider the complex performance, design, and implementation questions, because these are of too specialized nature—for more in-depth coverage of these topics a list of biometrics organizations and publications are provided at the end of this article, along with a list of references.

Before we continue, it would be useful to examine the current deployment of biometrics outside testing laboratories and the corporate perimeter. With the U.S. government fingerprinting and taking photographs of some of the visitors coming to the United States beginning January 5, 2004, under the US-VISIT program, biometrics and associated issues such as privacy and personal data protection are bound to get unprecedented levels of publicity[1]. Although it is too early to judge whether this innovation will actually contribute to overall security of the country or rather increase the general confusion surrounding security procedures, it has already resulted in more questions asked than answered. To some of its proponents, biometrics is a magic technology that would contribute to the security of their societies, to others the same technology heralds the coming of a police state and erosion of personal privacy and liberties and discrimination against (potentially not only) foreign citizens. Indeed, that was the opinion of Julier Sebastiao da Silva, a federal judge in Mato Grosso state of Brazil, who ordered similar measures to be taken in the case of U.S. citizens visiting Brazil[2]. Despite the announcement of Brazil's federal police that they may well seek to have this judgment overturned, this is a significant event because it illustrates that the use of biometrics is not only a technical procedure but also has its far-reaching social, legal, and international implications. It is immaterial whether this judgment will be upheld or overruled—it is the fact that introduction of the mandatory use of biometrics at borders resulted in such a response that is important.

Earlier announcement by the U.S. authorities that they expect the visa-waiver countries whose citizens currently may enter the U.S. without visas, simply upon presentation of their passports, to provide biometric data in newly issued passports also resulted in different reactions, ranging from support for the measure to outright condemnation[3].

Aside from the huge technological and logistical work that must be done in order to introduce biometrics into passports, these requirements also pose considerable legal and social issues in countries with strong personal privacy and data protection legislation in place. However, one thing is clear—biometrics ceases to be an exotic and little-used technology and is bound to be increasingly used in one way or another.

This article is organized as follows. First biometrics and related concepts are introduced, along with descriptions of the most widely used and understood physiological and behavioral biometrics. We will also see how biometric systems fail when inadequately designed or implemented. Later we describe the system and design issues of biometrics, such as security, accuracy, speed, resilience, privacy, and cost of biometric identification and verification systems, as well as practical applications of biometrics in network authentication and international travel documents.

Definition of Biometrics

A *biometric* is a physiological or behavioral characteristic of a human being that can distinguish one person from another and that theoretically can be used for identification or verification of identity. For a biometric to be practically useful, ideally it should be unique, universal, permanent, recordable, and acceptable—more on these properties of practical biometrics later.

Authentication in General

Authentication is the second step in the identify-authenticate-authorize process, which is done countless times every day by humans and computers alike. When speaking about human authentication, basically we have three choices: using something we know (such as passwords and passphrases), something we have (such as access tokens, smart cards, and so on) or something we are (biometrics). There is no "best" authentication method; each has its pros and cons, depending on the application, the users, and the environment. Whatever authentication method we use, we can make it stronger by using one or both of the other methods. An example of strong authentication would be a system that requires possession of a smart card, knowledge of a password or *Personal Identification Number* (PIN), and biometric verification. Obviously to steal or fake all three would be much more difficult than to steal or fake any one of these—however, more expensive and laborious to operate as well. The other two factors—the time of access and the location of subject—may also be used for access control, but usually only as auxiliary factors.

What You Know

Unquestionably the most widely used method of authentication, passwords, passphrases, and PINs share both pros and cons with each other. Moreover, an advantage in one situation easily becomes a problem in another—an example being the ease of password sharing. Passwords are easy to change, but are also easy to intercept. Systems can force the use of strong passwords, but the user may respond by storing or transmitting them in such a way that the added security is effectively reduced to nil.

Unauthorized disclosure of a password is not usually detected until after unauthorized access has already taken place. Passwords are also vulnerable to guessing, dictionary, and brute-force attacks. On the other hand, they require no additional hardware, they are an accepted method of authentication, and they are well-understood—even by the most technologically challenged part of human species.

What You Have

Smart cards, access tokens (both challenge-response and time-based), and other "what you have" authentication methods solve some of the problems associated with "what you know" authentication, but they create a set of different problems. Unlike theft of a password, theft of a smart card or access token can, of course, be easily detected. Unlike passwords, smart cards usually cannot be used simultaneously by two or more parties in different places. However, "what you have" authentication devices may be lost, damaged, and stolen. They may also run out of power (if self-powered) or may be prone to power-, synchronization- and time-based attacks if externally powered. They may also be subjected to reverse engineering and other treatment, which may compromise their security.

What You Are: Biometric Authentication

There are two biometric authentication methods: biometric verification and biometric identification of identity. Biometric identification is also sometimes referred to as *pure biometrics* because it is based only on biometric data and is more difficult to design and operate—but alas, pure biometrics is not the most secure, useful, or efficient one. Also, both methods can not always be used with all biometrics—some biometrics can only be used in verification mode because of their intrinsic properties.

Verification

Biometric verification uses entity IDs and a biometric—in this case biometric merely serves to prove identity already declared by the entity—which may be done using something you know (a username) or something you have (a smart card). Biometric (something you are) works to actually complete the authentication process. Hence, the biometric database keeps a list of valid entity IDs (which may be said to serve as primary keys to the database) and corresponding biometric templates, and compares ("matches") the stored template with the biometric provided. The result of this comparison is either an accept or reject decision based on a complex algorithm and system settings (refer to the section "Matching").

Identification

Unlike biometric verification of identity, biometric *identification* is based solely on biometrics. The biometric serves as both the identifier and the authenticator. The biometric database contains the enrolled biometric templates, and they all are compared against the provided biometric to find a match. Biometric identification may be described as "putting all your eggs in one basket," partly because somehow faking or stealing a biometric compromises both the ID and the authenticator.

A biometric identification system may operate in one of the two modes: positive identification or negative identification. In a positive identification biometric system, the provided biometric must be in the database and there must be only one match to positively identify the person. The risks present in a biometric system are false acceptance and false rejection, whereas unauthorized subjects are incorrectly accepted, or authorized ones are denied identification, resulting in a denial of service. A negative identification system, in contrast, works by determining whether the provided biometric is not in the database.

Enrollment

Regardless of the type of a biometric system, *enrollment* is a mandatory part of the process. Biometric enrollment is the registration of subjects' biometrics in a biometric database. Positive enrollment results in a database of recognized persons' biometric templates that may be later used for positive identification or verification. Negative enrollment results in a database of "excluded" persons, a black list if you wish. Security and reliability of the enrollment process and the biometric database are fundamental to the security of the entire system, but in practice they are difficult to achieve because of the myriad of issues that affect collection, transmission, storage, and usage of biometric data (see "Security" and "Privacy," later in this article for an overview of just some of the risks).

Matching

After an individual is enrolled—that is, the individual's biometrics are scanned and registered in the biometric database—*matching* is the next step. Biometric matching is essentially the comparison of the enrolled person's known biometric data stored in the biometric database in the form of biometric templates—binary representation of biometric sample—with the biometric provided by the individual at the identification or verification time. However, biometric matching is a pattern-recognition problem and not a simple bit-by-bit comparison—representation of the same biometric taken by two input sensors or taken at two different points in time does not match bit by bit because of numerous factors such as sensor resolution, system noise, and so on. Therefore, a degree of likeness (usually referred to as the *matching score*) is used to express how like the stored biometric is to the provided biometric. A *threshold level* is used to decide whether the matching score is high enough to be considered a match—if the score is at or below the threshold level, matching fails. This threshold level is one of the many variables that affect the accuracy—and hence security—of biometric authentication systems.

For biometric identification applications, the provided biometric is compared against all entries in the database and should result in only one successful match to result in positive identification. In biometric verification systems, the provided biometric is compared only with the biometric template or templates corresponding to the specified identity. As a result of biometric matching, the following system errors may occur:

- *False match or acceptance:* This occurs when the system decides that the two biometrics (the one stored in the database and the one provided now) are the same, when in reality they are not. The rate of false matches is known as *False Matching Rate* (FMR) or *False Acceptance Rate* (FAR). False acceptance is a confidentiality and integrity risk.
- *False nonmatch or rejection:* This is expressed as *False Rejection Rate* (FRR), and *False Nonmatching Rate* (FNMR). False nonmatch is when the system erroneously decides that biometrics are from different identities while in reality they are from the same person. False rejection is an availability risk.

In practice, both FRR and FAR do not equal zero, and in different applications one of them may be more important than the other. In an application that requires higher security (and hence as low FAR as possible), users may be troubled with high false rejection rates; whereas in an application that can accept somewhat higher false acceptance rates (such as public transport), false rejection rate is of more concern because of convenience and manual processing concerns. When FAR and FRR meet, that is the *Cross-over Error Rate* (CER). The lower the CER, the better—hence it is frequently used to express accuracy of biometric systems (although it is not the infallible measure as some suppose). Additionally, *Failure to Acquire* (FTA) errors occur when an individual does not have the required biometric or the biometric cannot be read by the sensor; and *Failure to Enroll* (FTE) is when a part of the targeted population may not be enrolled for whatever reason (such as a FTA). These errors directly affect the practicality of biometrics and must be accounted for with regard to the projected population of users.

Practicality of Biometrics

Writing in the December 1994 issue of *Information Technology & People* ("Human identification in Information Systems: Management Challenges and Public Policy Issues")[4] ten years ago, Roger Clarke proposed some criteria that should be met in order for a biometric to be practically usable:

- *Universality:* Every relevant person should have an identifier.
- *Uniqueness:* Each relevant person should have only one identifier, and no two people should have the same identifier.
- *Permanence:* The identifier should not change, nor should it be changeable.

- *Indispensability:* The identifier should be one or more natural characteristics, which each person has and retains.
- *Collectibility:* The identifier should be collectible by anyone on any occasion.
- *Storability:* The identifier should be storable in manual and in automated systems.
- *Exclusivity:* No other form of identification should be necessary or used.
- *Precision:* Every identifier should be sufficiently different from every other identifier that mistakes are unlikely.
- *Simplicity:* Recording and transmission should be easy and not error-prone.
- *Cost:* Measuring and storing the identifier should not be unduly costly.
- *Convenience:* Measuring and storing the identifier should not be unduly inconvenient or time-consuming.
- *Acceptability:* Its use should conform to contemporary social standards.

Although some of these criteria may be argued over, this set is nevertheless a useful reference. An interesting point is that no known biometric completely satisfies all of these criteria, perhaps proving that these are not strict "must haves" but instead guidelines to be accounted for.

Types of Biometrics

Two broad categories of biometrics exist: *physiological* biometrics (such as fingerprints, hand geometry, iris recognition) and *behavioral* biometrics (such as signature and voice biometrics). Physiological biometrics is based on direct measurements and data derived from measurements of a part of the human body, whereas behavioral biometrics is based on measurements and data derived from human actions, and indirectly measures characteristics of the human body over a period of time.

Physiological Biometrics

Relatively widely understood and used physiological biometrics are fingerprint recognition, face recognition, hand geometry, and iris recognition. These methods are introduced in the following sections.

Fingerprint Recognition

It is believed that no two persons share the same fingerprints—not even identical twins—because the fingerprint patterns are part of a person's phenotype and do not apparently depend on genetics[5]. Fingerprints have been used to identify humans for a long time—there is some evidence that thousands of years ago ancient Chinese were aware of the uniqueness of fingerprints[6], not speaking about their current use in forensic science and law enforcement. The traditional fingerprint acquisition mechanism—finger into ink and then on to paper—obviously is not usable in many—if not most—noncriminal applications.

Currently there are four known inkless fingerprint acquisition mechanisms considered suitable for use in practical biometrics.

Optical Sensing

Optical fingerprint sensing works by acquiring light reflected from the finger surface through a special prism. The result is an image of the finger surface. The downside of this method is that wet, dirty, or dry finger skin may result in a bad image.[7]

Thermal Sensing

With the thermal sensing method, a thermogram of the finger surface is taken and the resulting image is used.[8]

Capacitance Sensing

Because of differing capacitance of the ridges and valleys of fingers, a *Complementary Metal-Oxide Semiconductor* (CMOS) capacitance sensor can obtain an image of the finger when it is touched. However, like optical sensing, cpacitance sensing may be negatively affected by dry, dirty, or wet skin. [9]

Ultrasound Sensing

Ultrasound sensing works by using an ultrasound beam to scan the skin surface. Ultrasound sensing is not affected much by dry, dirty, or wet skin but takes longer to perform and the ultrasound sensing equipment is usually not compact and consequently not widespread.[10]

In addition to the mentioned issues of wet, dry, or dirty skin, numerous other factors may also affect the quality or the very possibility of taking a fingerprint. For example, although the absolute majority of people have at least one finger, many people may also have damaged skin or skin illnesses that may degrade the quality of fingerprints or render them unusable. Fingerprint matching approaches may be broadly categorized into three classes: feature techniques, imaging techniques, and hybrids of the two. In feature-based fingerprint matching techniques, a symbolic representation of the fingerprint, defined by so-called *minutiae*, is created from the fingerprint image, and it is this representation that is later stored and used to match fingerprints—not the raw fingerprint image itself[11]. Imaging techniques use the fingerprint images directly—image correlation algorithms are then used to compare the fingerprints[12].

The Mighty Fingers

If the defending technology is expensive and complex, it does not mean the attacking technology will also be complex and expensive—this has been proven by many successful security attacks. Tsutomu Matsumoto of the Yokohama National University successfully fooled numerous fingerprint readers into accepting fake fingers made of gelatin with a 80-percent success rate, sending a shock wave among biometrics proponents[13].

In a paper ambiguously entitled "Impact of Artificial Gummy Fingers on Fingerprint Systems," co-authored with H. Matsumoto, K. Yamada, and S. Hoshino and presented at the Optical Security and Counterfeit Deterrence Techniques IV conferene (Proceedings of the *International Society for Optical Engineering,* 2002), Matsumoto describes relatively easy ways to create artificial clones of fingers using cheap and freely available materials such as gelatin, free molding plastic, and photosensitive printed circuit boards.

Not only was he able to create a copy of a live finger that was good enough to fool most fingerprint readers used in the experiment, he also created an artificial finger using a latent fingerprint left on a glass, which was also accepted as genuine. In addition, Matsumoto mentions several other attack vectors against fingerprint systems, including instances where the registered finger is presented by an armed criminal, under duress, or on a sleeping drug; a severed fingertip of the registered finger; or a genetic clone of the registered finger.

Even if we disregard the last possibility as too expensive and unlikely, the others are indeed very real and must be disturbing to current users of fingerprint-based identification or verification systems. After this research was published, Bruce Schneier wrote in the May 2002 issue of his monthly newsletter CRYPTO-GRAM[14]:

> "There's both a specific and a general moral to take away from this result. Matsumoto is not a professional fake-finger scientist; he's a mathematician. He didn't use expensive equipment or a specialized laboratory. He used $10 of ingredients you could buy, and whipped up his gummy fingers in the equivalent of a home kitchen. And he defeated eleven different commercial fingerprint readers, with both optical and capacitive sensors, and some with "live finger detection" features. (Moistening the gummy finger helps defeat sensors that measure moisture or electrical resistance; it takes some practice to get it right.) If he could do this, then any semi-professional can almost certainly do much much more. More generally, be very careful before believing claims from security companies. All the fingerprint companies have claimed for years that this kind of thing is impossible. When they read Matsumoto's results, they're going to claim that they don't really work, or that they don't apply to them, or that they've fixed the problem. Think twice before believing them. "

Face Recognition

One of the most powerful drivers behind the use of face recognition is the fact that we all use face recognition every day to recognize people—so it seems to be one of the most acceptable biometrics we have (unlike, for example, fingerprints, which are often associated with criminal prosecution), not speaking about photographs that have been used for identification for many years[15]. However, despite progress in this area of biometrics, face recognition is still not accurate and dependable enough, and factors such as aging, changing hairstyles, beards, and moustaches only make reliable face recognition more difficult. Bruce Schneier, in his recent book *Beyond Fear,* had the following to say about the usefulness of face recognition systems[16]:

> "I'll start by creating a wildly optimistic example of the system. Assume that some hypothetical face-scanning software is magically effective (much better than is possible today)—99.9% accurate. That is, if someone is a terrorist, there is a 1-in-1000 chance that the software fails to indicate "terrorist" and if someone is not a terrorist, there is a 1-in-1000 chance that the software falsely indicates "terrorist." In other words, the defensive-failure rate and the usage-failure rate are both 0.1%. Assume additionally that 1 in 10 million stadium attendees, on average, is a known terrorist (this system won't catch any unknown terrorists who are not in the photo database). Despite the high (99.9%) level of accuracy, because of the very small percentage of terrorists in the general population of stadium attendees, the hypothetical system will generate 10,000 false alarms for every one real terrorist. This would translate to 75 false alarms per Tampa Bay football game and one real terrorist every 133 or so games."

Of course these issues do not apply exclusively to face recognition systems, but we get the idea—a system that generates so many false alarms and catches so few terrorists is not going to be successful. This was proven on several occasions. First at the Palm Beach International Airport, where a face recognition system failed by providing less than 50-percent recognition rate and generating a large number of false positives, resulting in a decision by the airport not to use the system at all[17]. Almost the same happened in the second case, at a face recognition system trial at the Boston Logan International Airport[18].

Hand Geometry

Features measured and used by hand geometry biometrics typically include length and width of fingers, different aspect ratios of palm and fingers, thickness and width of the palm, and so on[19]. Existing hand geometry systems mostly use images of the hand. Like face recognition, hand geometry is a user-friendly technology that scores higher on the acceptability test than, for example, fingerprints. It is also relatively more easily measurable and recordable than some other biometrics. Several patents have been issued for hand geometry systems, but there is not as much research as on fingerprints[20]. However, because of its biometric properties, hand geometry is not suitable for use in the identification mode.

Iris Recognition

Iris recognition-based biometric systems are believed to be very reliable and accurate[21]. Like fingerprints, the iris image is a part of human phenotype and is believed to be unique in every individual. Perhaps one of the most known cases of deployment of the iris recognition system is the Privium at Amsterdam's Schiphol International Airport. Frequent travelers may enroll in the system to enjoy fast border crossing by simply looking at the iris scanner, which authenticates the person and opens the gate[22]. In February 2004, an iris recognition system will also be piloted at the Frankfurt International Airport, and if the six-months-long trial concludes successfully, the system may be installed and deployed in 18 European countries[33]. Obviously, iris recognition would not work for people who are missing both eyes or who have serious eye illnesses that affect the iris.

Behavioral Biometrics

Two of the most used behavioral biometrics are signature- and voice-based systems. Another behavioral biometric, keystrokes (where the timing between successive key pressings is used), seems to receive increasing attention and use.

Signature

In use for centuries, signatures enjoy a high degree of acceptance, largely because of their everyday use and familiarity, but as a behavioral biometric, signatures lack permanence: they may change at the will of a person, or under influence from such factors as illness, mental state, medicines, emotions, or age. For these and other reasons, signature-based biometric systems function in the verification and not in the identification mode.

Two subtypes of signature verification systems exist: static signature verification systems, where only the graphical representation (image) of the signature is used, and dynamic signatures, where the dynamics, pressure, and speed of the movement of a special pen are used for verification. Although the first method does not require any special hardware, the dynamic signature verification requires the use of special electronic signature readers or high-quality tablets. It is understood that dynamic signature verification is more secure and reliable than static signatures[23]. However, some people do not have consistent signatures, resulting in increased false rejection rates to unacceptable levels and severely affecting the practical use of signature-based biometric systems.

Voice

Voice recognition systems (not to be confused with speech recognition systems, which are concerned with the actual words said and not the identity of the speaker) depend on numerous characteristics of a human voice to identify the speaker. Voice recognition holds much potential because it is acceptable and it does not require expensive input devices, unlike some other biometrics. Like face recognition, voice recognition is something we humans do many times a day; additionally, voice recognition is ideal for many practical and widespread telephony applications, and in theory voice recognition systems may even function in the background without forcing the users to go through a separate identification and verification process, saving us from another password to remember. But as usual, voice recognition systems also have their fair share of potential problems. As we all know, some people with exceptional vocal abilities may skillfully imitate others' voices, potentially defying such systems. Another issue is the ease of sound recording and replay, so any voice recognition system must be designed to withstand "record and replay" attacks.

Voice recognition also is influenced by the usual suspects—illness, mental state, emotions, age—which may substantially modify an enrolled subject's voice to a degree that it does not match the stored templates anymore. Several voice recognition models varying in accuracy and complexity exist.

The *fixed-text* model involves a person saying a word or phrase previously recorded and enrolled in the biometric database. The verification process is the simple comparison, possibly accounting for some allowable differences. However, if this word or phrase can be recorded, the entire system fails, because it is fairly easy to reproduce words and phrases.

Another model is *text-dependent,* meaning the system instructs the person to speak words or phrases—naturally this system is less prone to replay attacks because supposedly the person does not know in advance what words or phrases the system will ask for. A hybrid system, also known as *conversational voice verification,* combines something you are—your voice—and something you know—such as a password—to provide a higher degree of verification accuracy and reliability, and this system may well be the best choice in practice[24], so multimodal biometrics may hold the key to more accurate and practical biometric authentication. Again, we should keep in mind that some people cannot use this biometric for one reason or another.

System and Design Issues

The following is a quick overview of only some of the most important biometric system design and implementation considerations:

Security

Biometrics is invariably associated with security, hence the biometric system itself should be reasonably secure and trustworthy. Not only should the system provide the required functionality, but we also should have a degree of security assurance. Keeping in mind our track record of creating secure complex systems (almost an oxymoron), we should not really have high expectations this time either. If we have learned a lesson, it is that systems fail and malfunction, so recovery and compensating mechanisms should be in place from the beginning, and even the most sophisticated system should be expected to fail sooner or later, one way or another. Some of the biometrics security issues are discussed in the following section.

Rogue Sensors and Unauthorized Acquisition (theft) of Biometric Samples

One of the risks associated with the use of biometrics for identification or verification is that a biometric cannot be changed by definition—your fingerprint is your fingerprint and there is no easy way to change it—so if it is stolen and used to create a fake finger to impersonate you, there is not much you can do about yours. Therefore, the issue of mutual authentication of the individual and the sensor is of much importance. In practice, however, as illustrated by numerous stories about rogue *Automated Teller Machines* (ATM) harvesting unsuspecting victims' card and PINs, this would prove to be a difficult task. Unlike, for example, smart cards, which may use cryptographic protocols to establish with whom they are communicating, we humans have no secure way to ascertain whether the biometric reader attached to a computer somewhere is indeed under control of (let's say) a genuine Internet banking application and will not relay or store our biometric template without authorization.

In contrast, bank customers asked to authenticate themselves at a bank counter may have a reasonable expectation that their biometric will be used by the same bank for lawful purposes only—because of their and the sensor's physical location (so called location-based authentication). Still, unauthorized acquisition and use of biometrics remains one of the issues to be considered in any practical implementation.

The fact that not all biometrics require placing your finger on a fingerprint reader (such as face recognition systems) and that some biometric samples may be obtained without any action on part of the subject is further food for thought because one's biometrics may be acquired without knowledge or authorization.

Communications Security Between Sensors, Matchers, & Biometric Database(s)

Although as important as the previous issue, communications security between sensors, matchers, and biometric databases is easier to provide than to solve the problems of mutual authentication of humans and biometric sensors. Well-designed and well-implemented secure cryptographic protocols may provide the required security for sensitive data exchange between parts of a biometric identification or verification system, and they are unlikely to be the weak link in the biometrics chain.

Accuracy

A biometric system must be reasonably accurate—otherwise why would we need it? The widely used FAR and FRR, and their product, CER, are not really exact measures but often estimates made using assumptions—and these assumptions may not be reasonable in all circumstances.

Speed

Although the question of how fast the system works may not be a pressing issue in, say, a nuclear reactor access control system, it will be a crucial factor at installations such as airports or border crossing points where a large number of people needs to be reliably and quickly identified and authenticated.

Scalability

Biometric verification systems are significantly and inherently more scalable than biometric identification systems particularly because only one-to-one matching is required. A distributed, combined system using smart cards that store the owner's biometric template and compare the provided biometric in card is an example of a scalable distributed biometric verification system. However, as the previously described face recognition system experiences at airports show, system properties such as FRR must be considered in context—one false rejection a month may be acceptable, but a hundred false rejections a day clearly would not. Another scalability issue is the nature of biometrics. A scalable biometric—such as the iris—can theoretically be deployed on a large scale (with thousands or millions of enrolled users), but a biometric with weak scalability could provide acceptable error rates and performance only in small installations. Therefore, scalability is directly linked with the particular type of biometric used, and this seems to be accounted for by the International Civil Aviation Organization (see the section "Biometrics and Passports").

Resilience

A biometric system should be able to handle exceptions. An exception in this context might be a person without the required biometric or a person whose biometric may not be usable for some reason. In many cases exception handling means resorting to a manual process, which of course brings all the issues of human intervention (speed and social engineering, to name only two) with it and may mean life or death for a particular system or application.

Cost

Because laws of economics apply to almost every human activity, a biometric system should be reasonable in cost. Of course reasonableness of cost is a very subjective concept and would vary greatly between different environments and different uses.

Privacy

As mentioned in the beginning of this article, biometrics is argued to be one of the threats to privacy and anonymity in the modern age. The *Electronic Frontier Foundation* (EFF) lists the following as being the most important privacy concerns:

- Biometric technology is inherently focused on individuals and interfaces easily to database technology, making privacy violations easier and more damaging.
- Biometric systems are useless without a well-considered threat model.
- Biometrics are no substitute for quality data about potential risks.
- Biometric identification is only as good as the initial ID.
- Biometric identification is often overkill for the task at hand.
- Some biometric technologies are discriminatory.
- Biometric systems accuracy is impossible to assess before deployment.
- The cost of failure is high.

Indeed it is very depressing to imagine a society—or even worse, a world order—where everyone is forced into a biometric database and total control over all your actions and whereabouts during your entire life is maintained—and where you can never "change your username" or "log out." One cannot help but remember Benjamin Franklin's immortal statement that those who are willing to trade liberty for security deserve neither. However depressing, this image hopefully will not materialize—and to achieve that, biometric systems should provide reasonable privacy and specific use guarantees to the enrolled subjects; in addition, they must have effective systems of checks and balances to audit and assure conformance with these guarantees.

Standards in Biometrics

As Andrew Tanenbaum once supposedly said, the good thing about standards is that there are so many to choose from—regardless of whether he did or not, this statement perhaps does not yet seem to apply to biometrics standards.

- The *Common Biometric Exchange File Format* (CBEFF) describes a set of data elements necessary to support biometric technologies in a unified way, and provides for the exchange of security, processing, and biometric data in a single file. The U.S. *National Institute for Standards and Technology* (NIST) describes CBEFF as facilitating interoperability between different systems or system components, forward compatibility for technology improvement, and software/hardware integration[26].
- *BioAPI and Human Authentication API.* BioAPI and HA-API efforts merged in 1999 under the umbrella of the BioAPI Consortium. The current version of the BioAPI Specification is Version 1.1, which aims to provide a "standardized *Application Programming Interface* (API) that will be compatible with a wide range of biometric applications and a broad spectrum of biometrics technologies"[27].
- The Open Group's *Human Recognition Services* (HRS) is a module of the *Common Data Security Architecture* (CDSA), which in particular is used in Apple's Mac OS X. HRS is compatible with the CBEFF and, thanks to the CDSA modular and layered approach, can use services provided by other CDSA modules[28].
- *Biometrics Management and Security for the Financial Services Industry* (ANSI X9.84-2000) specifies minimum security requirements for effective use of biometrics data in the U.S. financial services industry, including collection, distribution, and processing of biometrics data. In particular, it specifies the security of the physical hardware used throughout the biometric life cycle; the management of the biometric data across its life cycle; the use of biometric technology for verification or identification of bank clients and employees; and other aspects. The data objects specified in X9.84 are compatible with CBEFF[29].
- The *American Association of Motor Vehicle Administrations* (AAMVA) *Driver's License and Identification* (DL/ID) standard provides a uniform way to identify holders of driver license cards within the United States and Canada. This standard specifies identification information on drivers' license and ID card applications, provides for inclusion of fingerprint data, and is compatible with BioAPI and CBEFF[30].
- *ANSI/NIST Data Format for the Interchange of Fingerprint, Facial, Scar Mark, and Tattoo Information* (ANSI/NIST-ITL 1-2000). This standard defines the content, format, and measurement units for the exchange of the specified information that may be used for identification of persons, and it is mainly directed at U.S. law enforcement agencies and government.[31]

Additionally, one of the groups of the *International Organization for Standardization* (ISO) is working toward inclusion of biometrics specifications in the widely used ISO 7816 standard for smart cards (Part 11: personal verification through biometric methods)[32].

Practical Uses of Biometrics

Because there may be as many practical uses of biometrics as users, we address just two of them: the use of biometrics for network authentication and the use of biometrics in international travel documents.

Biometrics for Network Authentication

As we saw earlier in this article, the accepted and widely used *what you know* and *what you have* authentication methods are not always—nor are they necessarily—secure or convenient, and they have their share of weaknesses.

The additional challenge of using biometrics for network authentication is the fact that the subject and the object of access are separated by a (usually uncontrolled, untrusted, and possibly hostile) network, which does not add to the simplicity or security of the system as a whole. As illustrated by the case of gelatin fingers described earlier, the question of whether a live person provided the biometric to a remote biometric sensor is even more important in network authentication applications when there are no preventive or detective controls, such as a watching guard, in place.

Although we have relied mostly on passwords to serve as the only or the main authentication mechanism until today, it has been clear for a while that passwords do not provide strong authentication. Keeping this lesson in mind, a biometric network authentication system should not depend solely on biometrics but should use one of the other authentication methods (what you know or what you have) as well.

The remote biometric sensors required in any biometric network authentication system are one of the most vital parts of the entire system, yet they are most vulnerable ones as well. For our purposes, we define the remote part of a centralized network authentication system as including a human user who needs to be authenticated as being physically present at the site and time of authentication, a general-purpose computer running a general-purpose operating system, and a special-purpose biometric sensor device directly connected to the general-purpose computer. This setup, therefore, includes the following high-level potential points of attack:

1. User
2. Path from the user to the sensor
3. Biometric sensor
4. Path from sensor to the general-purpose computer
5. Network
6. The central database

Even if the central authentication database is left out of the picture, the most simple risk assessment would reveal, among others, the following issues:

1. The user should be accurately identified or the declared identity should be verified; the sensor should be able to differentiate between a live human being providing live biometric and a biometric replica, such as an iris photograph or a gelatin finger. This includes, inter alia, reasonable assurance of the physical presence of the whole individual and not just the particular biometric at a particular point in time (hence, in part, the need for multimodal authentication involving not only what you are but also what you know or what you have).
2. The sensor should be sufficiently tamper-proof to withstand a defined set of attacks by a defined class of attackers, which would of course differ from environment to environment.
3. The communication protocol used between the sensor and the general-purpose computer should be simple, well-defined, and verified.
4. The role of the (untrusted) general-purpose computer and its software in such a system should be kept to a minimum. The biometric data acquired by the sensor should be cryptographically protected (encrypted and signed with the device key, for instance) inside the same sensor, without any dependence on action or inaction of the general-purpose computer. Their only role in this play should be to relay the bits from the sensor to the central authentication server for verification. Confidentiality and integrity of the biometric data should not be affected by a malicious, general-purpose computer or its software; the worst that can happen is the nondelivery of such data to the central authentication database.

An example of this approach would be a tamper-resistant fingerprint reader able to accurately recognize live human fingers (and reject fake ones), extract the required information, append a time stamp from an internal independent time source, encrypt and sign the resulting minutiae + time stamp data block using some digital signature algorithm, and send the resulting information through, for example, a *Universal Serial Bus* (USB) connection to the general-purpose computer. The general-purpose computer may then use the provided token to seek authentication from the central authentication database, provided all other requirements have been met.

Today a variety of network authentication systems that use or can use biometrics are available from numerous vendors. Aside from the objectively subjective information provided by vendors of such systems, little evidence of assurance exists that could enable potential users to evaluate them for their particular environments. The fact that most of these systems run as applications on the most widespread and arguably the least secure of operating systems perhaps speaks for itself.

Biometrics and Passports

For many years now more than 110 nations have issued machine-readable travel documents (mainly passports and visas) that conform to the *International Civil Aviation Organization* (ICAO) standard 9303. ICAO, a United Nations specialized agency, in addition to being responsible for international civil aviation matters, is also mandated to develop and adopt international standards on customs and immigration documents and procedures under the Chicago Convention. These machine-readable travel documents include a two-line area printed in *Optical Character Recognition* (OCR) B format, which contains information usually required for international travel (such as a person's name, date of birth, citizenship, document validity dates, and other information). These documents have greatly reduced the time necessary to check passports and visas by border officials, and have contributed to smoother international travel. In May 2003, the ICAO adopted a set of documents on integration of biometrics into machine-readable passports, choosing three most suitable for these purposes[2.5]. The main biometric chosen was a digitized face image, followed by two optional biometrics: fingerprints and irises. The ICAO also selected high-capacity, contactless smart cards as the storage method for this biometric data and gave other recommendations related to integration and use of biometrics in passports and other documents. It remains to be seen if or how and when 188 member states of the ICAO will integrate biometrics into their passports.

New Biometrics

It would be unreasonable to assume that we are aware of all possible biometrics. It may very well be the case that new biometrics are discovered and possibly, in the fullness of time, considered fit for practical use. An example would be a behavioral biometric proposed by Ross Anderson of Cambridge University, author of the already classic *Security Engineering:*

> "Are there any completely new biometrics that might be useful in some circumstances? One I thought up while writing this chapter, in a conversation with William Clocksin and Alan Blackwell, was instrumenting a car so as to identify a driver by the way in which he or she operated the gears and the clutch."

Summary

Biometrics is a promising and exciting area, where different disciplines meet and provide an opportunity for a more secure and responsible world. However, the same biometrics, if misused or poorly engineered, may instead bring many hassles—if not troubles. Some biometrics are less usable than others, and different environments warrant different biometrics and design considerations. The best advice would be to differentiate between market-ready biometric technologies and technologies that are not yet (if ever) ready for deployment outside testing grounds. However much fervent proponents and keen vendors of biometric solutions market their wares, the guiding factor should be proven reliability and appropriateness of these solutions to specific uses, not marketing hype, which seems at times to dominate this arena.

Organizations and Publications

The following organizations and publications may be useful sources of further information on biometrics and biometric applications:

The International Biometric Society: **www.tibs.org**

Biometric Consortium: **www.biometrics.org**

BioAPI Consortium: **www.bioapi.org**

International Biometrics Industry Association: **www.ibia.org**

International Association for Identification: **www.theiai.org**

Journal of the International Biometric Society: **stat.tamu.edu/Biometrics**

Biometric Digest: **www.biodigest.com**

Biometric Technology Today: **www.biometrics-today.com**

Additionally, the following books may serve as good introductions to biometrics:

Guide to Biometrics, by Bolle, Connell, Pankanti, Ratha, Senior, ISBN 0-387-40089-3, Springer Verlag, 2003

Practical Biometrics, Julian Ashbourn, Springer Verlag, 2003

One of the best publicly available works on security engineering is *Security Engineering: A Guide to Building Dependable Distributed Systems,* by Ross Anderson (Wiley, 2001).

References

[1] **http://www.dhs.gov/dhspublic/interapp/editorial/editorial_0333.xml**

[2] **http://news.bbc.co.uk/2/hi/americas/3358627.stm**

[3] **http://www.usatoday.com/tech/news/techpolicy/2003-08-24-biometrics-travel_x.htm**

[4] **http://www.anu.edu.au/people/Roger.Clarke/DV/HumanID.html**

[5] "On the individuality of Fingerprints. Pankanti," Prabhakar, Jain; *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* December 2001.

[6] "The History and Development of Fingerprinting," Lee, Gaensslen; *Advances in Fingerprint Technology,* CRC Press, 1994.

[7] *Guide to Biometrics,* Bolle et al., Springer Verlag, 2003.

[8] "Fingerchip: Thermal Imaging and Finger Sweeping in a Silicon Fingerprint Sensor," Mainguet, Pegulu, Harris; *Proceedings of AutoID 99,* October 1999.

[9] "Low-power and high-performance CMOS Fingerprint Sensing and Encoding Architecture," Jung, Thewes, Scheiter, Gooser, Weber; *IEEE Journal of Solid-State Circuits,* July 1999.

[10] "Ultrasound Sensor for Fingerprint Recognition," Biez, Gurnienny, Pluta; *Proceedings of SPIE—Optoelectronic and Electronic Sensors,* June 1995.

[11] "A Tree System Approach for Fingerprint Pattern Recognition. Moayer," Fu; *IEEE Transactions on Computers,* C-25(3).

[12] *Guide to Biometrics,* Bolle et al., Springer Verlag, 2003

[13] **http://www.itu.int/itudoc/itu-t/workshop/security/present/s5p4.pdf**

[14] **http://www.schneier.com/crypto-gram-0205.html#5**

[15] "Face Recognition: Features versus Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 12(10), October 1993.

[16] *Beyond Fear: Thinking Sensibly about Security in an Uncertain World,* Bruce Schneier; Copernicus Books, 2003.

[17] **http://www.theregister.co.uk/content/archive/25444.html**

[18] **http://www.theregister.co.uk/content/archive/26298.html**

[19] "A Hand Shape Identification System," Biometric Systems Lab, **http://bias.csr.unibo.it/research/biolab/hand.html**

[20] U.S. Patent 3,576,537; U.S. Patent 3,648,240

[21] "Iris Recognition: An Emerging Biometric Technology," Wildes; *Proceedings of the IEEE,* 85(9), September 1997.

[22] **http://www.schiphol.nl/schiphol/privium/privium_home.jsp**

[23] "Automatic On-line Signature Verification," Nalwa; *Proceedings of the IEEE,* 85(2), February 1997.

[24] "Speaker Recognition," Campbell, in *Biometrics: Personal Identification in Networked Society,* by Jain, Bolle, Pankanti, ISBN 0-7923-8345-1, Kluwer Academic Publishers, 1999.

[25] **http://www.icao.int/mrtd/download/technical.cfm**

[26] **http://www.itl.nist.gov/div895/isis/bc/cbeff/CBEFF010301web.PDF**

[27] **http://www.bioapi.org/**

[28] **http://www.opengroup.org/security/cdsa.htm**

[29] **http://www.ansi.org**

[30] **http://www.aamva.org**

[31] **ftp://sequoyah.nist.gov/pub/nist_internal_reports/sp500-245-a16.pdf**

[32] **http://www.iso.org**

[33] **http://news.com.com/2100-7348_3-5158973.html**

The author of this article does not work for, is not affiliated with, and has no financial interest or shareholding in any vendor of any biometric technology at the time of submission of this article for publication.

EDGAR DANIELYAN, CISSP, is a self-employed consultant, published author, editor, and instructor specializing in information security, UNIX, and internetworking. He is the principal partner at Danielyan Consulting LLP (**www.danielyan.com**), an information security assurance consultancy, and a member of ACM, IEEE, ISACA, USENIX, and the British Computer Society's Information Security Specialist Group.
E-mail: **edd@danielyan.com**

Book Reviews

The Unicode Standard

The Unicode Standard, Version 4.0, by The Unicode Consortium, ISBN: 0-321-18578-1, Addison Wesley Professional, 2003.

The Unicode 4.0 book is a thick, heavy one, but it is good. If you work with the Unicode character set, you should have this book on your bookshelf.

This book consists of four parts:

- Background and explanation of terms (103 pages)
- Implementation guidelines (29 pages)
- Technical specifications (60 pages)
- The Unicode Character Tables (1150 pages)

A review must describe each of these sections by itself, because they are important for different reasons. Unfortunately, however, the sections in the book are not clearly divided into sections as I outlined, so you don't necessarily know where to start. You don't need to read the characters section—just the sections you are interested in.

You should read the "Preface" (Section 0), because this section describes the rest of the book. It starts on page xxxi (before chapter 1).

You can then immediately go to the section you are interested in. Each section more or less stands by itself, and the book is easy to read. If something is not clear, you should look for text in another section that describes the subject. Reading from start to finish is possible, but I use this book as reference material, like an encyclopedia (except for the characters).

The background material is easy to read. It covers basic concepts such as differences between *characters* and *glyphs,* definition of terms such as *equivalence,* character encoding schemes and implication of things such as bidirectional text (mixed right-to-left and left-to-right text). Knowing how these things work is essential for anyone who either implements text engines of any kind or works on developing protocols or standards. This background material is easier understood read on paper and not electronically. It also is the part of the book I return to most often.

The second very good part concerns implementation guidelines. Even though it is (relatively) short, it is very important material. It discusses selection algorithms and other user interface guidelines, as well as other algorithms needed for, for example, comparison (what is called "Normalization"). I like this section as well, because it really describes the details you need to know when implementing anything Unicode related.

Unicode is a *large* character set. You see that in the more-than-1000 pages of "just characters." Of course, the tables themselves can be found on the Unicode Consortium Web site, but this book gives you a good overview. Part of this overview is a description of the *scripts* that Unicode covers, one at a time before the *codepoints* that come from those scripts. Still, this is the part that makes this book heavy, and a version without the codepoints would have been interesting by itself.

The book ends with more technical material, consisting mostly of references to, for example, *Unicode Technical Notes* and other standards documents that the Unicode Consortium produces, in addition to the Unicode Standard itself.

Useful reference

In summary, the first 130 pages (well, starting at page 40) in the book are very good. If you work at all with Unicode, you should read those pages. The rest of the book is good reference material.

Even though I have been working with Unicode for almost 10 years now, and for the last 8 years have weekly reviewed Unicode-related standards in the Internet Engineering Task Force, I see myself opening this book now and then. There is always something I need to check, and to be honest, I like encyclopedias on paper.

As reference material, this is a must-have item. If you want to read only the 140 interesting pages once, well, the book is possibly overkill.

—Patrik Fältström, Cisco Systems
paf@cisco.com

---

iSCSI: The Universal Storage Connection

iSCSI: The Universal Storage Connection, by John L. Hufferd, ISBN 0-201-78149-X, Addison-Wesley, 2003.

I have to come clean straightaway and say that when I received this book to review I had never even heard of *Internet Small Computer System Interface* (iSCSI) and, to be honest, I have never heard it mentioned by anyone again since the day the book arrived. This is, of course, not a criticism of this book, just a comment on the current state of penetration of iSCSI into everyday computing discourse. In fact, if you search Google for "iscsi," you get only 465,000 hits—very few indeed these days, though this does have the decided advantage that the links you get are generally pretty useful. I'm sure that this will change because there are lots of big names behind the protocol, and certainly when vendors start really selling kit that uses it. *Storage Area Networks* (SANs) are important (though also not yet at the forefront of most computing people's minds)—and iSCSI will probably make them bigger.

However, to the book. And, really, if you want to know pretty well everything about iSCSI and don't want to read lots of Web sites, then this book is for you. It covers everything from the background behind the protocol, to how and where it might be applied, to all the low-level information that most of us hope that we never need to see. I'm not going to list it all and go into detail: the whole thing is here, from soup to nuts.

As to the presentation of the material, it is excellent—clear diagrams and useful tables. The layout is spacious without huge amounts of wasted white space on every page—making a change from many textbooks you see today.

The writing is clear too, though I did find myself becoming a bit bogged down in all the abbreviations (no, not acronyms—most of them are not words), which seem to pile up in the sentences. I got a bit tired of seeing iSCSI everywhere after a while too. I wasn't keen on the end-of-chapter summaries, finding them a bit redundant.

Good Reference

All in all, if you are in a position where you need to know about iSCSI and may have to be involved in working with it at a low level, this book is a good reference. I doubt that there is anything more comprehensive or better written at the present time.

—Lindsay Marshall, University of Newcastle upon Tyne
lindsay.marshall@newcastle.ac.uk

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

NRO Comments Concerning ICANN and WSIS

The *Number Resource Organization* (NRO) is the coalition of *Regional Internet Registries* (RIRs) which operate in the world today. The NRO is an organization representing the collective experience of individual RIRs and their communities. While the prime subject of its work are matters of joint interest relating to Internet numbering resources, the NRO provides an efficient interface to other parties interested in these issues. As the Internet continues to evolve, the NRO will ensure continuity of the operational infrastructure of Internet number resource allocation.

The RIRs are responsible for distribution of *Internet Number Resources* [IPv4 and IPv6 addresses and *Autonomous System Numbers*]. These number resources are the most fundamental of the identifiers on which the Internet relies: the Internet can operate without domain names; but it cannot operate without numbers. The RIRs have carried the responsibilities associated with managing these critical resources collectively for over 10 years, since well before the start of ICANN. This has been done very effectively through the entire "modern history" of today's Internet which includes both the "dot com boom" and the "dot com bust."

The RIRs have participated in the *World Summit on the Information Society* (WSIS) processes for over a year, including regional Prepcoms and the Summit itself. This is probably longer than any other Internet organization. The RIRs have attended as observers, and as subject matter experts with a genuine aim to assist in debates and discussions around issues related to Internet Number Resources in general and to IP addresses in particular.

The RIRs participated in the WSIS Phase I process as full supporters of ICANN as the model which represents not only the fundamental and critical aspects of Internet development to date, but also the means of community self-regulation to administer and manage Internet Number Resources. It must be understood that this is not given by the RIRs as mere components of ICANN, dependent upon it for support; but rather as independent components of the broader Internet administrative framework which ICANN itself is intended to support.

In the second round of WSIS, the NRO speaking for the collective RIRs will assert an active role vis-à-vis ICANN in order to aid that organization to address the genuine questions that it faces. The principle of these issues within the WSIS context is that of the independence and genuine internationalization of ICANN.

Therefore the NRO calls on ICANN to continue its work in this area, not by building a multinational organization, but rather by including and gaining the genuine support of its significant base of core stakeholders, namely those in the DNS, IP address, and protocol communities. Furthermore, the NRO calls on ICANN to work with the US Government to demonstrate a genuine and unambiguous plan for its independence and to commit to this plan before the conclusion of the second phase of the WSIS.

Finally, the NRO rejects any concept of an alternative Internet administrative model located within any governmental or intergovernmental structure. The NRO acknowledges that there is a valid role for governments in the administration of the Internet but this must be in the context of the current model. There is a need for the continual improvement of the current model of industry self-regulation to the extent that the ultimate solution may look little like today's ICANN.

http://www.apnic.net/index.html
http://www.arin.net/index.html
http://www.lacnic.net/
http://www.ripe.net/index.html

Upcoming Events

INET/IGC 2004 will be held in Barcelona, Spain, May 10–14, 2004. INET, which is the annual conference of the *Internet Society* (ISOC), will this time be held jointly with Spain's *Internet Global Congress* (IGC). For more information, visit: **http://www.isoc.org/inet04/**

The *North American Network Operators' Group* (NANOG) will meet in San Francisco, May 23–25, 2004. For more information see: **http://nanog.org/**

The *South Asian Network Operators Group* (SANOG) will meet 23–30 July, 2004 in Kathmandu, Nepal. More info at: **http://www.sanog.org/**

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in Kuala Lumpur, Malaysia, July 19–23, 2004, and in Cape Town, South Africa, December 1–5, 2004. For more information see: **http://www.icann.org**

The *Internet Engineering Task Force* (IETF) will meet in San Diego, CA, August 1–6, 2004 and in Washington, DC, November 7–12, 2004. For more information, visit: **http://ietf.org**

The *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will be held Feburary 16–25, 2005 in Kyoto, Japan and February 15–24, 2006 in Bangalore, India. For more information visit: **http://www.apricot.net/**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

June 2004 Volume 7, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

The Internet Protocol Journal continues to be a forum for discussion of current and emerging technologies. In this issue, we first look at *content networking*. One can describe the Internet as a system of interconnected devices, but equally as a collection of information, called *content*, that resides on a distributed set of *servers* and is accessed by numerous *clients*. Our first article is by Christophe Deleuze.

Engineers are hard at work planning for an eventual transition to the next version of IP — IPv6. We've published several articles about IPv6 in previous editions. This time, François Donzé describes the automatic address configuration feature of IPv6. Of note is also the increasing global support for IPv6 deployment, (refer to "Fragments" on page 31).

Our final article returns to our recurring theme: adding security to existing Internet protocols. Because many malicious attacks on the Internet are perpetrated by "spoofing" information in one form or another, it makes sense to look at the *Domain Name System* (DNS), a critical component of the Internet infrastructure. Today, it is possible to create systems which provide fake answers to DNS queries. Miek Gieben explains what is being done to address this issue in his tutorial on DNSSEC, the secure version of the DNS protocols.

Please take a moment to renew or update your subscription to this journal. You can do so by visiting **www.cisco.com/ipj** and clicking on the "Subscription Information" link on the left. You will need to supply your subscription ID and e-mail address in order to gain access to your database record. If you have any questions, please send a note to **ipj@cisco.com**.

This is the 25th edition of IPJ. The journal now has more than 32,000 subscribers world-wide, and is available on paper and electronically on our Website in PDF and HTML format. The Website, located at **www.cisco.com/ipj**, contains all our back issues, and will soon offer a cumulative index in ASCII format that will make it easier to find particular articles. As always, we welcome your feedback.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Content Networks

by Christophe Deleuze

The Internet is constantly evolving, in both usage patterns and underlying technologies. In the last few years, there has been a growing interest in *content-networking* technologies. Various differing systems can be labelled under this name, but they all share the ability to access objects in a location-independent manner. Doing so implies a shift in the way communications take place on the Internet.

The Classic Internet Model

The Internet protocol stack comprises three layers, shown in Figure 1. The network layer is implemented by IP and various routing protocols. Its job is to bring datagrams hop by hop to their destination host, as identified by the destination IP address. IP is "best effort," meaning that no guarantee is made about the correct delivery of datagrams to the destination host.

The transport layer provides an end-to-end communication service to applications. Currently two services are available: a reliable ordered byte stream transport, implemented by the *Transmission Control Protocol* (TCP), and an unreliable message transport, implemented by the *User Datagram Protocol* (UDP).

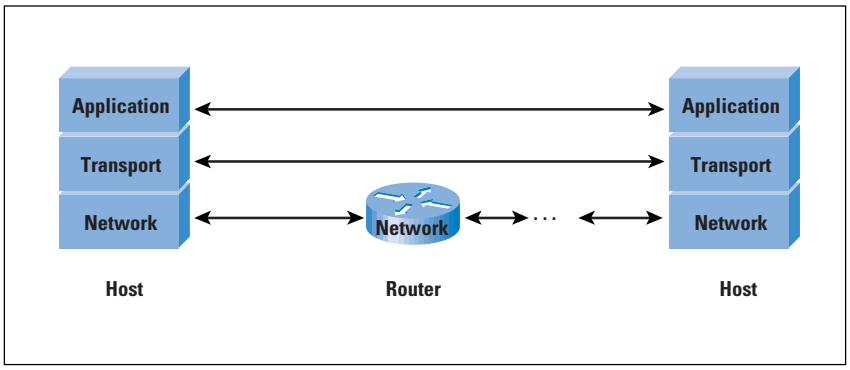


Figure 1: The Three Layers of the Internet Protocol Stack

Above the transport layer lies the application layer, which defines application message formats and communication semantics. The Web uses a client-server application protocol called *Hypertext Transfer Protocol* (HTTP)[10].

A design principle of the Internet architecture is the "end-to-end principle," which states that everything that can be done in the end hosts should be done there, and not in the network itself[8]. That is why IP service is so crude, and transport and application layer protocols are implemented only in the end hosts.

Application objects, such as Web pages, files, etc. (we will simply call those "objects") are identified by URLs. (Actually URLs identify "resources" that can be mapped to different objects called "variants." A variant is identified by a URL and a set of request header values, but in order to keep things simple, we will not consider this in the following.) URLs for Web objects have the form **`http://host:port/path`**. This means that the server application lives on a host with *hostname* (or possibly IP address) on port *N* (with default value of 80), and knows the object under the name *path*. Thus URLs, as their name implies, tell where the object can be found. To access such an object, a TCP connection is open to the server running on the specified host and port and the object named path is requested.

Content Networks

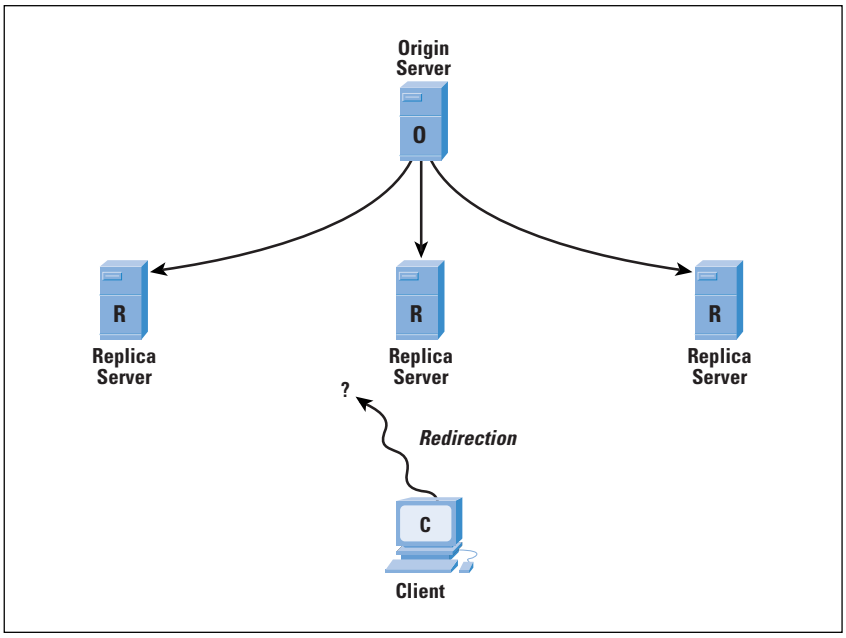
Content networks aim to provide location-independent access to an object, most commonly because they handle some kind of (possibly dynamic) replication of the objects. By design, URLs are not suited to identify objects available on several places on the network.

Handling such replication and location-independent access usually involves breaking the end-to-end principle at some point. Communication is no more managed end to end: intermediate network elements operating at the application layer (whose most common types are "proxies") are involved in the communication. (Content networks are not the only case where this principle is violated.)

In the same way that IP routers relay IP datagrams (that is, network layer protocol data units), routing them to their destination according to network layer information, those application layer nodes relay application messages, using application layer information (such as content URLs) to decide where to send them. This is often called *content routing*.

So the goal of a content network is to manage replication, handling two different tasks: *distribution* ensures the copying and synchronization of the instances of an object from an *origin server* to various *replica servers,* and *redirection* allows users to find on instance of the object (possibly the one closest to them.) (By "replica," we mean any server of any kind other than the origin that is able to serve an instance of the object. This term often has a narrower meaning, not applying, for example, to caching proxies.) This is illustrated in Figure 2.

Figure 2: Elements of a Content Network



Various kinds of content networks exist, differing in the extent to which they handle these tasks and in the mechanisms they use to do so. There are many possible ways to classify them. In this article, we use a classification based on who owns and administers the content network. We thus find three categories: content networks owned by network operators, content providers, and users.

Network Operators' Content Networks

Network operators (also called *Internet Service Providers*, or ISPs) often install caching proxies in order to save bandwidth[11]. Clients send their requests for objects to the proxy instead of the origin server. The proxy keeps copies of popular objects in its cache and can answer directly if it has the requested object in cache. (To be precise, such a caching proxy does not cache objects, but server responses.) If this is not the case, it gets the object from the origin server, possibly stores a copy in its cache, and sends it back to the client.

This caching proxy scheme can be used recursively, making those proxies contact parent proxies for requests they cannot fulfill from their local store. Such hierarchies of caching proxies actually lead to constructing content-distribution trees. This makes sense if the network topology is tree-like, although there are some drawbacks, including the fact that less popular objects (those not found in any cache) experience delays, which increase with the depth of the tree. Another problem is with origin servers whose closest tree node is not the root.

The Squid caching proxy[5] can be configured to choose the parent proxy to query for a request based on the domain name of the requested URL (or to get the object directly for the origin server). This allows setting up multiple logical trees on the set of proxies, a limited form of content routing.

Such manual configuration is cumbersome, especially because domain names do not necessarily (and actually most do not) match network topology. Thus the administrator must know where origin servers are in the network to use this feature effectively.

The same effects can be achieved, to some extent, in an automatic and dynamic fashion using ICP, the *Internet Cache Protocol* [16, 15]. ICP allows a mesh of caching proxies to cooperate by exchanging hints about the objects they have in cache, so that a proxy missing an object can find a close proxy that has it. One advanced feature of ICP allows you to select among a mesh of proxies the one that has the smallest *Round-Trip Time* (RTT) to the origin server.

One design flaw of ICP is that it identifies objects with URLs. We mentioned previously that a URL actually identifies a resource that can be mapped to several different objects called variants. Thus information provided by ICP is of little use for resources that have multiple variants. However, in practice most resources have only one variant, so this weakness does little harm.

Users normally configure their browsers to use a proxy, but automatic configuration is sometimes possible. Multiple proxies can be used by a client with protocols such as the *Cache Array Routing Protocol* (CARP)[14]. To avoid configuration issues, a common trend is for ISPs to deploy *interception proxies*. Network elements such as routers running the Cisco *Web Cache Communication Protocol* (WCCP)[6,7] redirect HTTP traffic to the proxy, without the users knowing. The proxy then answers client requests pretending to be the origin server. This poses numerous problems, as discussed in [12].

Caching proxies have limited support for ensuring object consistency. Either the origin server gives an expiration date or the proxy estimates the object lifetime based on the last modification time, using an heuristic known as *adaptive TTL* (time to live).

Content Providers' Content Networks

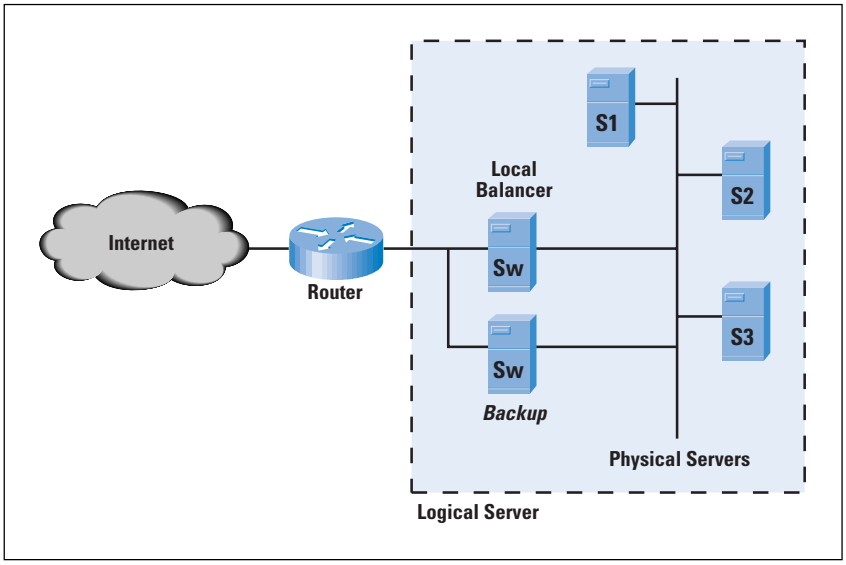
Contrary to ISPs whose main goal is to save bandwidth, content providers want to make their content widely available to users, while staying in control of the delivery (including ensuring that users are not delivered stale objects). We can again roughly classify such content networks in three subcategories:

- *Server farms:* Locally deployed content networks aimed at providing more delivery capacity and high availability of content
- *Mirror sites:* Distributed content networks making content available in different places, thus allowing users to get the content from a close mirror
- *Content-Delivery Networks* (CDNs): Mutualized content networks operated for the benefit of numerous content providers, allowing them to get their content replicated to a large number of servers around the world at lower cost.

Server Farms

Server farms are made of a load-balancing device (we will call it a *switch*) receiving client requests and dispatching them to a series of servers (the *physical* servers). The whole system appears to the outside world as a single *logical* server. The goal of a server farm is to provide scalable and highly available service. The switch monitors the physical servers and uses various load metrics in its dispatching algorithm. Because the switch is a single point of failure, a second switch is usually set up in a hot failover standby mode, as shown in Figure 3.

Figure 3: Server Farm



Some switches are called *Layer 4 switches* (4 is the number of the transport layer in the *OSI Reference Model*), meaning they look at network and transport layer information in the first packet of a connection to decide to which physical server the incoming connection should be handed. They establish a state associating the connection with the chosen physical server and use it to relay all packets of the connection. The exact way the packets are sent to the physical servers varies. It usually involves some form of manipulation of IP and TCP headers in the packets (like *Network Address Translation* [NAT] does) or IP encapsulation. These tricks are not necessary if all the physical servers live on the same LAN.

More complex *Layer 7 switches* (7 is the number of the application layer in the OSI Reference Model) look at application layer information, such as URL and HTTP request headers. They are sometimes called *content switches*. On a TCP connection, application data is available only after the connection has been opened. A proxy application on the switch must thus accept the connection from the client, receive the request, and then open another connection with the selected physical server and forward the request. When the response comes back, it must copy the bytes from the server connection to the client connection.

Such a splice of TCP connections consumes much more resources in the switch than the simple packet manipulation occurring in Layer 4 switches. Bytes arrive at one connection and are handed to the proxy application, which copies them to the other connection—all of this involving multiple kernel mode-to-user mode memory copy operations and CPU context switches. Various optimizations are implemented in commercial products. The simplest one is to put the splice in kernel mode. After it has sent the request to the physical server, the proxy application asks the kernel to splice the two connections, and forgets about them. Bytes are then copied between the connections directly by the kernel, instead of being given to the proxy application and back to the kernel.

It is even possible to actually merge the two TCP connections, that is, simply relay packets at the network layer to establish a direct TCP connection between the client and the physical server. This requires manipulating TCP sequence numbers (in addition to addresses and ports) when relaying packets, because the two connections will not have used the same initial sequence numbers. This can be much more complex (or even impossible) to perform if TCP options differ in the two connections.

Mirror Sites

In such a content network, a set of servers are installed in various places in the Internet, and they are defined as *mirrors* of the master server. Synchronization is most commonly performed periodically (often every night), using FTP or specialized tools such as *rsync*[4].

Redirection is performed by the users themselves for most sites. The master server, to which the user initially connects, displays a list of mirrors with geographic information and suggests that users choose a mirror close to themselves, by simply clicking on the associated link.

This process can be automated sometimes. One trick is to store the user's choice in a *cookie*, such that the next time the user connects to the master site, the information provided in the cookie will be used to issue an *HTTP redirect* (an HTTP server response asking the client to retry the request on a new URL) to the previously selected site.

Other schemes involve trying to find which of the mirrors is closest to the user based on information provided in the user request (such as preferred language) or indicated by network metrics. Such schemes were not very common for simple mirror sites, but today many commercial products allowing for this kind of "global load balancing" are available.

In any case (except if redirection is automatic and *Domain Name System* [DNS] based—this is discussed in the next section) the URLs of objects change across mirrors.

CDNs

Most content providers cannot afford to own numerous mirror sites. Having servers in different places around the world costs lots of money. Operators of CDNs own a large replication infrastructure (Akamai, the biggest one, claims to have 15,000 servers) and get paid by content providers to distribute their content. By mutualizing the infrastructure, CDNs are able to provide very large reach at affordable costs.

CDN servers do not store entire sites of all the content providers, but rather cache a subset according to local client demand. Such servers are called *surrogates*. They manage their disk store like proxies do, and serve content to clients like mirrors do (that is, contrary to proxies, they act as the authoritative source for the content they deliver).

Because the number of surrogates can be so large, and because of the argument that "no user configuration is necessary," CDNs typically include complex redirection systems that allow them to perform automatic and user-transparent redirection to the selected surrogate. The selection is based on information about surrogate loads and on network metrics collected by various ways such as routing protocol information, RTTs measured by network probes, etc. The client is made to connect to the selected surrogate either by sending it an HTTP redirect message, or by using the DNS system: when the client tries to resolve the host name of the URL in an IP address to connect to, it is given back the address of the selected surrogate instead. Using the DNS ensures that the URL is the same for all object copies. In this case, CDNs actually turn URLs into location-independent identifiers.

In addition to proxy-like on-demand distribution, content can also be "pushed" in surrogates in a proactive way. Synchronization can be performed by sending invalidation messages (or updated objects) to surrogates.

CDN principles are also being used in private intranets for building *Enterprise CDNs* (ECDNs).

Users' Content Networks

User-operated content networks are better known as *Peer-to-Peer* (P2P) networks. In these networks, the costly replication infrastructure of other content networks is replaced by the users, who make some of their storage and processing capacities available to the P2P network. Thus, no big money is needed, and no one has control over the content network.

One advantage P2P networks have over other content networks is that they are usually built as overlay networks and do not strive for transparent integration with the current Web. Thus they are free to build new distribution (some of them allow downloading files from multiple servers in parallel) and redirection mechanisms from scratch, and even to use their own namespace instead of being stuck with HTTP and URLs.

P2P networks basically handle the distribution part of replication in a straightforward way: the more popular an object is, the more users will have a copy of it, thus the more copies of the object will be available on the network. More complex mechanisms can be involved, but this is the basic idea.

The redirection part of replication is more problematic with most current P2P networks. It can be handled by a central directory as in *Napster:* every user first connects to a central server, updates the directory for locally available objects, and then looks up the directory for locations of objects the user wants to access. Of course, such a central directory poses a major scalability and robustness problem.

Gnutella and *Freenet,* for example, use a distributed searching strategy instead of a centralized directory. A node queries neighbors that themselves query neighbors, and so on until either one node with the requested object is found or a limit on the resources consumed by the search has been hit. Although there is no single point of failure, such a scheme is no more scalable that the central directory. It seems easy to perform denial-of-service attacks by flooding the network with requests. Additionally, you can never be sure you have found the object even if someone has it.

These examples are primitive and have serious flaws, but much research work is being performed on this topic; refer to [13] for a summary.

Although they are currently used mainly for very specific file-sharing applications, P2P networks do provide new and valuable concepts and techniques. For example, *Edge Delivery Network* is a commercially available software-based ECDN inspired by Freenet. Various projects use a *scatter/gather* distribution scheme, useful for very large files: users download several file chunks in parallel from other currently downloading users, thus refraining from using server resources for long periods of time.

Some projects attempt to integrate P2P principles in the current Web architecture and protocols. Examples are [3] and [1].

Conclusion

Current networks have been designed and deployed as ad-hoc solutions of specific problems occurring in the current architecture of the network. Caching proxies lack proper means to ensure consistency, but CDNs tricks the DNS to turn URLs into location-independent identifiers. P2P networks are mostly limited to file-sharing applications.

Content networks implement mechanisms to ensure distribution of content to various locations, and redirection of users to a close copy. They often have to break the end-to-end principle in order to do so, mainly because current protocols assume each object is available in only one statically defined location.

Probably the first step in building efficient distribution and redirection mechanisms for providing an effective replication architecture is the setting up of a proper replication-aware namespace. Applications would pass an object name to a name resolution service and be given back one or more locations for this object. The need for such a location-independent namespace was anticipated a long time ago. URLs are actually defined as one kind of *Uniform Resource Identifier* (URI), another one being *Uniform Resource Names* (URNs) intended to provide such namespaces. A URN IETF working group [2] has been active for a long time, and recently published a set of RFCs (3401 to 3406).

Work on the topic of content networking has also been performed by the now closed *Web Replication and Caching* (WREC) IETF working group, which issued a taxonomy in [9]. An interesting survey of current work on advanced content networks is [13].

References

[1] BitTorrent: **`http://bitconjurer.org/BitTorrent/`**

[2] IETF URN Working Group: **`http://www.ietf.org/html.charters/urn-charter.html`**

[3] Open Content Network: **`http://www.open-content.net`**

[4] Rsync: **`http://rsync.samba.org`**

[5] Squid Internet Object Cache: **`http://www.squid-cache.org`**

[6] M. Cieslak and D. Forster, "Web cache coordination protocol v1.0," Expired Internet Draft, **`draft-forster-wrec-wccp-v1-00.txt`**, Cisco Systems, July 2000.

[7] M. Cieslak, D. Forster, G. Tiwana, and R. Wilson, "Web cache Coordination Protocol v2.0," Expired Internet Draft, **`draft-wilson-wrec-wccp-v2-00.txt`**, Cisco Systems, July 2000.

[8] David D. Clark, "The design philosophy of the DARPA Internet protocols," *Computer Communication Review,* Volume 18, No. 4, August 1988. Originally published in Proceedings of SIGCOMM'88.

[9] Ian Cooper, Ingrid Melve, and Gary Tomlinson, "Internet Web Replication and Caching Taxonomy," RFC 3040, January 2001.

[10] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol — HTTP/1.1," RFC 2616, June 1999.

[11] Geoff Huston, "Web Caching," *The Internet Protocol Journal,* Volume 2, No. 3, September 1999.

[12] Geoff Huston, "The Middleware Muddle," *The Internet Protocol Journal,* Volume 4, No. 2, June 2001.

[13] H. T. Kung and C. H. Wu, "Content Networks: Taxonomy and New Approaches," 2002. **http://www.eecs.harvard.edu/htk/publication/2002-santa-fe-kung-wu.pdf**.

[14] Vinod Valloppillil and Keith W. Ross, "Cache array routing protocol v1.0," Expired Internet Draft, **draft-vinod-carp-v1-03.txt**, February 1998.

[15] D. Wessels and K. Claffy, "Application of Internet Cache Protocol (ICP), Version 2," RFC 2187, September 1997.

[16] D. Wessels and K. Claffy, "Internet Cache Protocol (ICP), Version 2," RFC 2186, September 1997.

CHRISTOPHE DELEUZE holds a Ph.D. degree in computer science from Université Pierre et Marie Curie, Paris. He worked on quality-of-service architectures in packet networks, and then spent three years in a start-up company designing CDN systems. He has also been a teacher. E-mail: **christophe.deleuze@free.fr**

IPv6 Address Autoconfiguration

by François Donzé, HP

Since 1993 the *Dynamic Host Configuration Protocol* (DHCP)[1] has allowed systems to obtain an IPv4 address as well as other information such as the default router or *Domain Name System* (DNS) server. A similar protocol called DHCPv6[2] has been published for IPv6, the next version of the IP protocol. However, IPv6 also has a stateless autoconfiguration protocol[3], which has no equivalent in IPv4.

DHCP and DHCPv6 are known as *stateful* protocols because they maintain tables within dedicated servers. However, the stateless autoconfiguration protocol does not need any server or relay because there is no state to maintain.

This article explains the IPv6 stateless autoconfiguration mechanism and depicts its different phases.

Scope of IPv6 Addresses

Every IPv6 system (other than routers) is able to build its own unicast global address. A *unicast* address refers to a unique interface. A packet sent to such an address is treated by the corresponding interface—and *only* by this interface. This type of address is directly opposed to the multicast address type that designates a group of interfaces. Most of this article deals with unicast addresses. For simplicity, we will omit the unicast qualifier when there is no ambiguity.

Address types have well-defined destination scopes: *global*, *site-local* and *link-local*. Packets with a link-local destination must stay on the link where they have been generated. Routers that could forward them to other links are not allowed to do so because there has been no verification of uniqueness outside the context of the origin link.

Similarly, border-site routers cannot forward packets containing site-local addresses to other sites or other organizations. The IETF is currently working on a way to remove or replace site-local addresses. Hence, this article will refrain from any other reference to this address type. Finally, a global address has an unlimited scope on the worldwide Internet. In other words, packets with global source and destination addresses are routed to their target destination by the routers on the Internet. A fundamental feature of IPv6 is that all *Network Interface Cards* (NICs) can be associated with several addresses.

At minimum, a NIC is associated with a single link-local address. But in the most common case a NIC is assigned a link-local and at least one global address. The following command displays the configuration of network interface **`eth1`** on a Red Hat system. This interface is associated with two IPv6 addresses. One of them starts with **`fe80::`** and the other with **`3ffe:`**. The scope of the first one is the link and the second has a global scope.

```
root# ip address list eth1
3: eth0: <BROADCAST,MULTICAST,UP mtu 1500 qdisc pfifo_fast qlen 100
link/ether 00:0c:29:c2:52:ff brd ff:ff:ff:ff:ff:ff
inet6 fe80::20c:29ff:fec2:52ff/10 scope link
inet6 3ffe:1200:4260:f:20c:29ff:fec2:52ff/64 scope global
```

Creation of the Link-Local Address

An IPv6 address is 128 bits long. It has two parts: a *subnet prefix* representing the network to which the interface is connected and a *local identifier*, sometime called token. In the simple case of an Ethernet medium, this identifier is usually derived from the EUI-48 *Media Access Control* (MAC) address using an algorithm described later in this article. The subnet prefix is a fixed 64-bit length for all current definitions. Because IPv4 manual configuration is a well-known pain, one could hardly imagine manipulating IPv6 addresses that are four times longer. Moreover, a DHCP server is not always necessary or desired; in the case of a remote control finding the DVD player, a DHCP environment is not always suitable.

Because the prefix length is fixed and well-known, during the initialization phase of IPv6 NICs, the system builds automatically a link-local address. After a uniqueness verification, this system can communicate with other IPv6 hosts on that link without any other manual operation.

For a system connected to an Ethernet link, the build and the validation of the link-local address is the following:

1. An identifier is generated, supposedly unique on the link.
2. A tentative address is built.
3. The uniqueness of this address on the link is verified.
4. If unique, the address from phase 2 is assigned to the interface. If not unique, a manual operation is necessary.

Although a local policy can decide to use a specific token, the most common method to obtain a unique identifier on an Ethernet link is by using the EUI-48 MAC address and applying the modified IEEE EUI-64 standard algorithm. A MAC address (IEEE 802) is 48 bits long. The space for the local identifier in an IPv6 address is 64 bits. The EUI-64 standard explains how to stretch IEEE 802 addresses from 48 to 64 bits, by inserting the 16 bits **`0xFFFE`** at the 24th bit of the IEEE 802.

By doing so, transforming MAC address **`00-0C-29-C2-52-FF`** using the EUI-64 standards leads to **`00-0C-29-FF-FE-C2-52-FF`**. Using IPv6 notation, we get **`000C:29FF:FEC2:52FF`**. Recall that the notation of IPv6 addresses requires 16-bit pieces to be separated by the character "**`:`**". Then, it is necessary (RFC 3513) to invert the universal bit ("u" bit) in the 6th position of the first octet. Thus the result is: **`020c:29ff:fec2:52ff`**.

Universal uniqueness of IEEE 802 and EUI-64 is given by a "u" bit set to 0. This global uniqueness is assured by IEEE, which delivers those addresses for the entire planet. Inverting the "u" bit allows ignoring it for short values in the manual configuration case, as explained in paragraph 2.5.1 of RFC 3513[4].

The second phase of creating automatically a link-local address is to prepend the well-known prefix **`fe80::/64`** to the identifier resulting from phase one. In our case we obtain **`fe80::20c:29ff:fec2:52ff`**. This address is associated with the interface and tagged "tentative." Before final association, it is necessary to verify its uniqueness on the link. The probability of having a duplicate address on the same link is not null, because it is recognized that some vendors have shipped batches of cards with the same MAC addresses.

This is the goal of the third phase, called *Duplicate Address Detection* (DAD). The system sends ICMPv6 packets on the link where this detection has to occur. Those packets contain *Neighbor Solicitation* messages. Their source address is the undefined address "**`::`**" and the target address is the tentative address. A node already using this tentative address replies with a *Neighbor Advertisement* message. In that case, the address cannot be assigned to the interface. If there is no response, it is assumed that the address is unique and can be assigned to the interface.

We are reaching the last step of the automatic generation of a link-local address. This phase removes the "tentative" tag and formally assigns the address to the network interface. The system can now communicate with its neighbors on the link.

Global Prefixes

In order to exchange information with arbitrary systems on the global Internet, it is necessary to obtain a global prefix. Usually (but not necessarily), the identifier built during the first step of the automatic link-local autoconfiguration process is appended to this global prefix.

However, before assigning this global address, the system verifies again that no duplicate address exists on the link. DAD is performed for all addresses before they are assigned to an interface, because uniqueness in one prefix does not automatically assure uniqueness in any other available prefixes.

Generally, global prefixes are distributed to the companies or to end users by *Internet Service Providers* (ISPs).

Random Identifiers

The EUI-48-to-EUI-64 transform process is attractive because it is simple to implement. However, it generates a privacy problem. Global unicast as well as link-local addresses may be built with an identifier derived from the MAC address. A Website tracking where a node frequently attaches can collect private information such as the time spent by employees in the enterprise or at home.

Because a MAC address follows the interface it is attached to, the identifier of an IPv6 address does not change with the physical location of the Internet connection. Hence it is possible to trace the movements of a portable laptop or *Personal Digital Assistant* (PDA) or other mobile IPv6 device.

RFC 3041[5] allows the generation of a *random* identifier with a limited lifetime. Because IPv6 architecture permits multiple suffixes per interface, a single network interface is assigned two global addresses, one derived from the MAC address and one from a random identifier. A typical policy for use of these two addresses would be to keep the MAC-derived global address for inbound connections and the random address for outbound connections. A reason for not using it for inbound connections is the need to update the DNS just as frequently as it is changes.

Such a system, with two different global addresses—one of which changes regularly—becomes very difficult to trace.

By default, Microsoft enables this feature on Windows XP and Windows Server 2003. The random-identifier-based global addresses of Microsoft systems have the address type "temporary." EUI-64 global addresses have type "public." Those types as well as other information can be displayed in a **`cmd.exe`** DOS-box with the command line:

```
netsh interface ipv6 show address
```

IPv6 Routers

By definition, a router is a node that forwards IP packets not explicitly addressed to it. IPv6 routers are certainly compliant with this definition but, in addition, they regularly advertise information on the links to which they are connected—provided they are configured to do so. These advertisements are *Internet Control Message Protocol Version 6* (ICMPv6) *Router Advertisement* (RA) messages, sent to the multicast group **`ff02::1`**. All the systems on a link must belong to this group, and nodes configured for autoconfiguration, among other things, analyze the option(s) of those messages. They might contain any routing prefix(es) for this segment.

Router Solicitation

Upon reception of one of those RA messages and according to local algorithm policy, an autoconfiguring node not already configured with the corresponding global address will prepend the advertised prefix to the unique identifier built previously.

However, the advertisement frequency, which is usually about ten seconds or more, may seem too long for the end user. In order to reduce this potential wait time, nodes can send *Router Solicitation* (RS) messages to all the routers on the link. Nodes that have not configured an address yet use the unspecified address "**`::`**". In response, the routers must answer immediately with a RA message containing a global prefix. This router solicitation corresponds to ICMPv6 messages of type RS, sent to the all-router multicast group: **`ff02::2`**. All routers on the link must join this group.

Thus, a node soliciting on-link routers in such a way is able to extract a prefix and build its global address. Note that this method using an advertised prefix is possible only for end nodes. Today IPv6 routers are usually manually configured. The reason is obvious: a stateless automatic configuration requires the advertisement of a prefix. This prefix is sent by a router. The router sending the prefix must be fully configured to do so. The easiest way to break this seemingly unsolvable problem is to manually configure IPv6 routers. However, some automatic methods are being developed[6].

Conclusion

Stateless address autoconfiguration is a new concept with IPv6. It gives an intermediate alternative between a purely manual configuration and stateful autoconfiguration. In addition to ease of use with no dedicated server or relay, this mechanism removes problems that have not been discussed here, such as the mismatch between the DCHP server and the router (prefix topology) or the IPv4 need to readdress subnets that have outgrown their prefix. Moreover, automatic renumbering (prefix change) is also possible on nodes using stateless autoconfiguration.

References

RFCs can be found at **`http:://www.ietf.org/rfc/`**

[1] Droms, R., "Dynamic Host Configuration Protocol," RFC 1531, October 1993.

[2] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., Carney, M., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3315, July 2003.

[3] Thomson, S., Narten, T., "IPv6 Stateless Address Autoconfiguration," RFC 2462, December 1998.

[4] Hinden, R., Deering, S., "Internet Protocol Version 6 (IPv6) Addressing Architecture," RFC 3513, April 2003.

[5] Narten, T., Draves, R., "Privacy Extensions for Stateless Address Autoconfiguration in IPv6," RFC 3041, January 2001.

[6] Prefix delegation:
`http://www.ietf.org/internet-drafts/draft-ietf-dhc-dhcpv6-opt-prefix-delegation-06.txt`

FRANÇOIS DONZÉ studied at the University of Utah in Salt Lake City. In 1989 he joined Digital Equipment Corporation as a UNIX and network teacher. He is now a technical consultant at HP, based in Sophia-Antipolis, France, promoting IPv6 and other leading-edge technologies. The author of several internal articles, he also publishes in French magazines. E-mail: **`francois.donze@hp.com`**

DNSSEC: The Protocol, Deployment, and a Bit of Development

by Miek Gieben, NLnet Labs

> "One Key to rule them all,
> one Key to find them,
> one Key to bring them all
> and in the Resolver bind them."
> —Modified from *Lord of the Rings*.

The *Domain Name System* (DNS) (RFCs 1034 and 1035) is a highly successful and critical part of the Internet infrastructure. Without it the Internet would not function. It is a globally distributed database, whose performance critically depends on the use of caching.

Unfortunately the current DNS is vulnerable to so-called *spoofing attacks* whereby an attacker can fool a cache into accepting false DNS data. Also various man-in-the-middle attacks are possible. The *Domain Name System Security Extension* (DNSSEC) is not designed to end these attacks, but to make them detectable by the end user. Or more technically correct: detectable by the security-aware *resolver* doing the work for the end user. This saves users from doing online banking on the wrong server even if a secured connection is used and the address in the browser looks correct.

DNSSEC is about protecting the end user from DNS protocol attacks. In order to make it work, zone owners (such as **`.com`**, **`.net`**, **`.nl`**, etc.) need to deploy DNSSEC in their zones. End users then need to update their resolvers to become security-aware (that is, understand DNSSEC) and add some trusted keys. These keys are called *anchored keys;* they are configured in the resolver and cannot be changed or updated very easily. If this is all configured, the end user will (finally) be able to detect attacks.

DNSSEC, as defined in (hopefully soon-to-be-obsoleted) RFC 2535, adds data origin authentication and data integrity protection to the DNS. The *Public Key Infrastructure* (PKI) in DNSSEC may be used as a means of public key distribution, which may be used by other protocols. *IP Security* (IPSec) and the *Secure Shell* (SSH) protocol, for example, are already considering the use of DNSSEC to carry their keying material.

In the course of early-deployment experiments carried out by various organizations, it became evident that RFC 2535 introduced an administrative key-handling and maintenance nightmare. This in turn would mean the DNSSEC deployment would never start (or be successful, for that matter).

The IETF DNSEXT working group decided to fix this problem, and to incorporate all drafts and RFCs written since RFC 2535 into a new DNSSEC specification.

This (still ongoing) effort became known as the *RFC 2535bis* DNSSEC specification. This work has resulted in three drafts, each handling a specific part of the new specification. These drafts follow:

1. dnssec-intro[1] provides an introduction into DNSSEC.
2. dnssec-records[2] introduces the new records for use in DNSSEC.
3. dnssec-protocol[3] is the main document, which details all the protocol changes.

The documents are now almost ready (July 2004) to be submitted to the *Internet Engineering Steering Group* (IESG) for review. It is hoped that soon after this is done the drafts will become RFCs. It could be that 2004 will be the year of DNSSEC.

In this article I use the terms *domain* and *zone*. These are important concepts in the DNS and in DNSSEC. The difference between a zone and a domain is worth highlighting. A domain is a part of the DNS tree. A zone contains the domain names and data that that domain contains *except* for the domain names and data that are delegated elsewhere. Also refer to [4].

Consider, for instance, the **`.com`** *domain,* which includes everything that ends in **`.com`**. **`CNN.com`** is in the **`.com`** domain. The **`.com`** *zone,* however, is the entity handled by VeriSign.

One other important concept in DNS is the *Resource Record* (RR) and the *Resource Record Set* (RRset). An RR in DNS is, for instance:

```
www.example.org.  IN   A    127.0.0.1
```

... where **`www.example.org`** is the "ownername" or "name." **`IN`** is the class (IN stands for Internet). **`A 127.0.0.1`** is the type (together with its rdata). **`A`** stands for "address." This 3-tuple (name, class, type) together make up the resource record. RRset are all the RRs that have an identical name, class and type. Only the rdata is different. Thus:

```
www.example.org.  IN   A    127.0.0.1
www.example.org.  IN   A    192.168.0.1
```

... together form a RRset, but:

```
www.example.org.  IN   A    127.0.0.1
www.example.org.  IN   MX   mail.example.org.
```

... do not (their type is different). In the DNS an RRset is considered *atomic* and the smallest data item. In DNSSEC each RRset gets a signature.

What Is DNSSEC?

DNSSEC adds data origin authentication and data integrity to the DNS. To achieve this, DNSSEC uses public key cryptography; (almost) everything in DNSSEC is digitally signed.

Public key cryptography uses a single key split in two parts: a private and a public component. The *private* component, also known as the *private key,* must be kept secret. The *public* component (the *public key*) can be made public. Both these keys can be used for cryptographic operations, albeit with different goals.

If a message is scrambled with the public key, it can be decrypted only with the private key. This is called *encryption* of the message and it ensures that only the holder of the private key can read the original message. When the private key is used to scramble a message, everybody can use the available public key to decipher the message. This last operation is called (digitally) *signing* a message (for increased speed usually a hash of the message is signed). In this case you know where the message comes from (*authenticated data origin* in cryptographic jargon). An added benefit of signing messages is that when the data is mangled during transport the signature is no longer valid. This last property is called *authenticated data integrity*. A more lengthy introduction on public key cryptography can be found at [10]. In DNSSEC only digital signatures (signing) are used, and nothing is ever encrypted.

For every secure zone there must be a public key in the DNS for use by DNSSEC. Each zone administrator generates a key to be used for securing a zone. The private key is (of course) kept private and is used in the "signing process" to create the signatures. The public key is published in DNSSEC as a DNSKEY record, which is the zone key. The generated signatures are published as RRSIG records.

If RRsets in DNSSEC do not have a valid signature, they are labeled bogus by the resolver. Bogus data should not be trusted, because probably somebody is trying to conduct a spoof attack. DNSSEC further distinguishes between:

- Verifiable secure—The data has signatures that are valid.
- Verifiable unsecure*—The data has no signatures.
- Old-style DNS—A non-DNSSEC lookup is done.

* Yes, Unsecure. This word has somehow evolved from "insecure."

Verifiable secure data is data that has valid signatures, and the key used to create those signatures is trusted (anchored in the resolver). Verifiable unsecure data is data for which we know for sure we do not need to do signature validation. Old-style DNS is the current (insecure) method of getting DNS data.

The signing of data in DNSSEC is comparable to the *Gnu Privacy Guard* (GPG) signing of e-mail. If I trust a public key from someone, I can use that key to verify the GPG signature and authenticate the origin of the e-mail.

The problem with both DNSSEC and GPG lies in the "...If I trust the public key from someone." GPG solves this with public key servers, key signing parties at various events and thus the creation of a web of trust. For DNSSEC such solutions are impractical. DNSSEC uses a different, but very elegant mechanism called the *chain of trust*.

The chain of trust makes it possible to start with a root zone key, the highest possible key in the DNS tree, and following cryptographic pointers to lower zones. Each pointer is validated with the previous validated zone key. (The root key is the key used in the root zone of the Internet; it is the key used in the **.** (dot) zone. It could take a while before the root is signed.)

By using this mechanism only the root key is needed to validate *all* DNSSEC keys on the Internet. With these DNSSEC keys the DNS data in each zone can then be validated. So, unlike GPG, we need to distribute only one key. This can be done by publishing it on the World Wide Web or in a newspaper or putting an ad on TV, etc.

One of the current items in the DNSSEC community is to outline procedures and guidelines on how to update this root and other keys.

Chain of Trust

To start securely resolving in DNSSEC, a root key must be anchored in the resolver at your local computer or nameserver. Only when a resolver knows and trusts a zone key can it validate the signatures belonging to that zone. Because of the chain of trust, a resolver has to carry only a few zone keys to be able to validate DNSSEC data on the Internet.

The chain of trust works by following "secured pointers," which are called *secured delegation* in DNSSEC. A special, new record called the *Delegation Signer* (DS) record delegates trust from a parental key to a child's zone key.

The DS record holds a hash (*Secure Hash Algorithm 1* [SHA-1]) of a child's zone key. This DS record is signed with the zone key from the parent. By checking the signature of the DS record, a resolver can validate the hash of the child's zone key. If this is successful, the resolver can compare this (validated) hash with the (yet-to-be-validated) hash of the child's zone key. If these two hashes match, the child's real zone key can be used for validation of data in the child's zone. Note: by successfully following a secured delegation, the amount of trust a resolver has in the parental key is transferred to a child's key. This is the crux of the chain of trust.

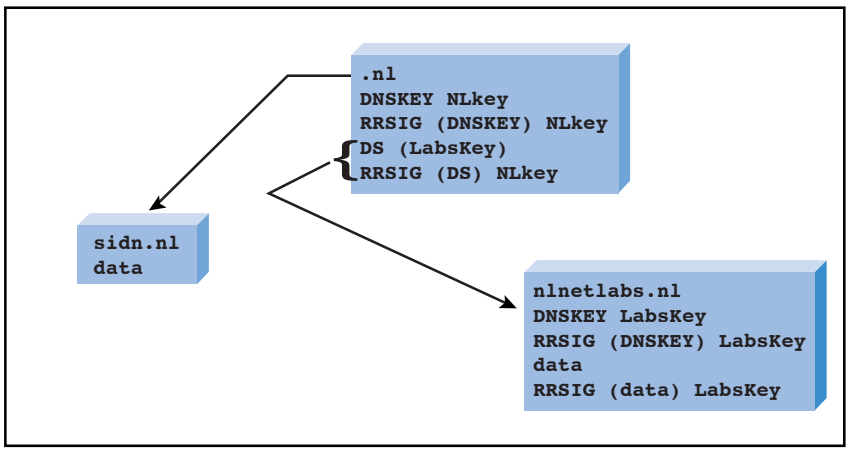


Figure 1: **nlnetlabs.nl** *is a secured delegation under* **.nl**. *RTSIG(x)y denotes that a signature over a data x is created with key y.*

In Figure 1 the following takes place.

The **.nl** zone contains the following:

```
nl.   IN    SOA (soa-parameters)
; the zone key
nl.   IN    DNSKEY NLkey
nl.   IN    RRSIG(DNSKEY)NLkey
nl.   IN    RRSIG(SOA)NLkey

nl.   IN    NS ns5.domain-registry.nl.
; this NS is authoratitive
nl.   IN    RRSIG(NS) NLkey

nlnetlabs.nl.   IN   NS open.nlnetlabs.nl.
; no RRSIG here (nonauthoritative data is not signed)

; DS record with a hash of the child's zone key
nlnetlabs.nl.   DS  hash(LabsKey)
; The signature of the parent
nlnetlabs.nl.   RRSIG(DS)NLkey
```

Note: It is important to see that we now have linked a parental signature to something that is *almost* the key of the child.

And the **nlnetlabs.nl** zone has the following:

```
nlnetlabs.nl.    IN    SOA (soa-parameters)
; The zone key
nlnetlabs.nl.    IN    DNSKEY LabsKey
nlnetlabs.nl.    IN    RRSIG(SOA)Labskey
; The (self) signature of the zone key
nlnetlabs.nl.    IN    RRSIG(DNSKEY)Labskey
nlnetlabs.nl.    IN    NS open.nlnetlabs.nl.
nlnetlabs.nl.    IN    RRSIG(NS)LabsKey
```

So the chain of trust looks like the following:

```
.nl DNSKEY –> nlnetlabs.nl DS –> nlnetlabs.nl DNSKEY
```

... and with that last key we can validate the data in the **nlnetlabs.nl** zone.

With this "trick" all keys from all the secure **.nl** zones can be chained from the **.nl** "master" key. So instead of one million (the number of zones in **.nl** currently) we need to configure only one key.

As you might have guessed, getting the root zone signed as soon as possible will make it possible to have one key that validates all other keys on the Internet.

We can also look at it from the resolver side. A resolver wants to get an answer. With DNSSEC it has to deal with signatures, keys, and DS records, but those are "side issues"; it still wants an answer.

Suppose **.nl** is secured and a secure delegation to **nlnetlabs.nl** exists. Our resolver has the key of **.nl** anchored. The nameservers of the root zone are also known to the resolver. We further assume the root is not signed. The resolver wants to resolve the address (A record) of **www.nlnetlabs.nl**. What does the actual resolving process look like in DNSSEC? Numerous steps need to be performed:

1. Go to a root server and ask our question.
2. The root server does not know anything about **www.nlnetlabs.nl**, but it *does* know something about **.nl**. The root nameserver refers us to the **.nl** nameservers. This kind of answer is called a *referral*.

3a. Notice that we have a key for **.nl** anchored.

3b. Go to the **.nl** nameserver and ask the **.nl** DNSKEY.

4a. Compare the two DNSKEYs. Continue with the secure lookup only if they match.

The **.nl** DNSKEY is now validated.

4b. Optionally, the RRSIG on the DNSKEY also can be checked.

5. Ask a **.nl** nameserver our question.
6. The **.nl** nameserver is also oblivious about **www.nlnetlabs.nl**, but it does know something about **nlnetlabs.nl**. It returns a secure referral consisting of a DS record plus the RRSIG and some nameservers.
7. The resolver now checks the signature on the DS record. If the signature is valid, the hash of the **nlnetlabs.nl** zone key is ok. The nameservers in the referral do not have any signatures on them.

   The hash of the **nlnetlabs.nl** DNSKEY is validated with the **.nl** DNSKEY.

8. Go to the nameserver as specified in the referral and ask for the **nlnetlabs.nl** DNSKEY.
9. Hash the DNSKEY of **nlnetlabs.nl** and compare this hash with the hash in the DS record. If they match continue with the secure lookup.

   The **nlnetlabs.nl** DNSKEY is now validated.

10. Ask the nameserver of **nlnetlabs.nl** our question.

11. The nameserver now responds with an answer consisting of the A record of **`www.nlnetlabs.nl`** and an RRSIG made with the **`nlnetlabs.nl`** DNSKEY.
12. The resolver now uses the already validated **`nlnetlabs.nl`** DNSKEY to check the RRSIG. If that signature is valid the RR with the answer is ok and can be given to the application.
13. After these steps we find out that the address of **`www.nlnetlabs.nl`** is 213.154.224.1. We also know it is not a spoofed answer.

This looks like a lot of work and it is—a recursive resolver is a complicated piece of software. Keep in mind, though, that only steps 3ab, 4ab, 7, 8, 9, and 12 are needed for DNSSEC; the rest is how resolving is done in the DNS today.

Deployment

As mentioned earlier, each zone owner generates its own key. To make the secure delegation actually work, this key must somehow be securely transferred to the parent, which is usually the local registry. The registry must have procedures in place to determine whether or not the uploaded key really belongs to the domain it claims to come from. During the *Secure Registry* (SECREG) experiment[5] NLnet Labs has researched the impact DNSSEC has on registries.

But even before the key can be actually uploaded to the parent, a zone administrator still has to do some work; the DNS zone must be signed. This process, called *zone signing,* turns a DNS zone into a DNSSEC zone.

The signing is done offline; first you sign, and then you load the zone. This setup was chosen because at the time (late 1990) computers were not fast enough to generate the signature in real time. Currently it would be possible to do this, but having a server sign every answer it gives is a *Denial-of-Service* (DoS) attack waiting to happen. Especially root servers will be unable to do this.

In DNSSEC a zone can have multiple keys. The signed zone then has multiple signatures per RRset (one for each key). There is no protocol limit on the number of keys. Here we sign with only one zone key. Also signatures in DNSSEC have a start and end date, that is, before and after a certain date interval the signature can no longer be used for validation.

If you use DNSSEC, you must re-sign your zone to generate new signatures with a new validity interval.

The signing of a zone consists of the following steps:

1. The zone key is added to the zone file.
2. The zone file is sorted.

3. Each owner name (for example, a host name) in the zone gets a *Next SECure* (NSEC) record. (Refer to the section "Authenticated Denial of Existence.")
4. For each secured delegation, a DS record is added.
5. The entire zone is then signed with the private key of the zone. Each authoritative RRset gets a signature, including the newly generated NSEC records.

Berkeley Internet Name Domain (BIND)[6] version 9—a popular implementation of the DNS protocols—contains a tool *dnssec-signzone,* which does steps 2 through 5 automatically; we only (manually) need to add the zone key to the zone file. The net result is that we have a bigger, signed, DNSSEC zone. A typical DNSSEC zone is 7 to 10 times larger than its DNS equivalent.

Experiments have shown that this does not pose much of a problem, even for such so-called country code *Top Level Domains* (ccTLDs) as **`.nl`**. The signed **`.nl`** zone was 350 megabytes, slightly more than a half a CD-ROM. And even if scaling problems are occurring, 64-bit machines would certainly help.

A few years ago there was much concern about the signing time. There was fear that it would be impossible to sign large zones, such as **`.com`**.

Experiments disproved this fear. Furthermore, a zone can be split up in pieces and each piece can be signed on a different machine. Later all the signed pieces can be put back together. Signing DNS zones is a highly parallel process.

After signing the zone, it can be loaded in the nameserver. If a resolver is DNSSEC-aware and has been configured with a trusted key that has a chain of trust to the zone key, it can validate the answers. If an answer does not validate, something is wrong and the DNS data must not be used.

The actual Internet-wide deployment of DNSSEC can happen incrementally. Each zone can decide to join independently. It is expected that initially DNSSEC is deployed in subsections of the Internet. These so-called *Islands of Trust* can appear anywhere on the Internet or even in intranets. The only requirement is that the key of the island of trust is distributed to the resolver. Resolvers configured with the key of a certain island of trust are called the *resolvers of interest.* Of course when DNSSEC is widely deployed on the Internet all resolvers are resolvers of interest and will have that key preconfigured.

Authenticated Denial of Existence

As mentioned previously, all records are signed offline. When a nameserver receives a query it looks up the answer plus the signature and returns the two (RRSIG + RRset) to the resolver. The signature is thus not created in real time. How can a secure-aware nameserver then respond to a query for something it does not know (that is, give an NXDOMAIN answer)? The only way to have offline signing and NXDOMAIN answers work together is to somehow sign the data you do not have.

In DNSSEC this is accomplished by the *Next SECure* (NSEC) record. This NSEC record holds information about the next record; it spans the nonexistence gaps in a zone, so to say. For this to work, a DNSSEC zone must be sorted (this is where that requirement stems from). To clarify this, consider an example.

We have a DNS zone, with (for the sake of clarity only the NSEC records are shown):

a.nl
d.nl
e.nl

Next we generate (with the signer) our DNSSEC zone:

a.nl
a.nl NSEC d.nl (span from **a.nl** to **d.nl**)

d.nl
d.nl NSEC e.nl (span from **d.nl** to **e.nl**)

e.nl
e.nl NSEC a.nl (loop back to **a.nl**)

1. If a resolver asks information about b.nl, the nameserver tries to look up the record fails. Instead it finds **a.nl**. It must then return: **a.nl NSEC d.nl** together with the signature. The resolver must then be smart enough to process this information and conclude that **b.nl** does not exist. If the signature is valid, we have an *authenticated denial of existence*. These NSEC records together with their signatures are the major cause of the zone size increase in DNSSEC.

Road to the DS Record

This section briefly considers the history of DNSSEC and, in particular, why the DNSEXT working group has invented this peculiar DS record, which can only exist at the parent side of a zone cut.

In RFC 2535 the DS record did not exist, and this is the reason that the key management in RFC 2535-DNSSEC is very, very cumbersome. In 2000 NLnet Labs ran its first experiment to test deployment of DNSSEC in the Netherlands. Because **.nl.nl** was chosen as the zone under which the secure tree would grow, this experiment became known as the *nl-nl-experiment*. With this experiment it was shown that the current DNSSEC standard (the soon-to-be-obsoleted RFC 2535) was difficult to deploy[7].

An update of a zone key in a child zone required up to 11 (coordinated and sequential) steps with the parent zone. The **.nl** zone now has more than 1 million delegations, so updating all the child zones would require more than 11 million steps. Because these updates could be quite frequent (once a month is typical), this is clearly an administrative nightmare.

Worse yet, if **.nl** lost its private key, all child-zone administrators would have to be notified and they would have to resubmit their public key for re-signing with the new **.nl** key. And because under these conditions the DNS may have been hacked and is thus untrusted, **.nl** is limited in its communication through the Internet; e-mail may not be the preferred method. A telephone call would be more safe, but what kind of organization can make up to one million phone calls in a few days ..?

After various failed attempts (sig@parent[8]) to fix this behavior, the DS record was introduced[1,3]. With this record the administration nightmare is solved, because DS introduces an indirection from the parent zone to a child's zone key.

If **.nl** loses its private key, it can easily resign its own zone, *without* contacting all its children. The DS to child key indirection is still valid, and only the signature of the DS record needs to be updated. This is a local operation.

To test this new DNSSEC specification, a new experiment was set up, which would build a shadow DNSSEC tree in the **.nl** zone. This experiment, called *SECREG,* was to test the new procedures in DNSSEC and, of course, the new DS record. Detailing the conclusions of this experiment is beyond the scope of this article, but in short the conclusion was that the new DNSSEC procedures do not pose much difficulty. At some point, more than 15,000 zones were delegated from the secure tree. A writeup of the experiment and the conclusions can be found in "DNSSEC in NL"[5].

Settings and Parameters in DNSSEC

DNSSEC brings many new parameters to the DNS, including cryptographic ones such as key sizes, algorithm choices, and key and signature lifetimes. Because DNS never has involved cryptography, the best values for these parameters are still open for debate. There is, however, some documentation and knowledge available on this topic (refer to [9] for instance).

One of the major issues is how large (bit length) to make a zone key and how often to re-sign a zone file. The current view is that a parent zone should use larger keys and re-sign more often than a child zone. Also the signature lifetime should be shorter in a parent zone.

Because a parent zone has a DS record (and signature) of a child's zone key, it can decide how long this DS RRSIG must be valid. The shorter this validity interval is, the better protected the child. If a cracker steals a child's zone key, it can forge DNS data. This data looks genuine because the cracker has access to the private key. As long as there is a valid chain of trust to this hijacked key, the child is vulnerable. This chain of trust is broken as soon as the RRSIG of the DS record expires. This argues in favor of a very short parental RRSIG over the DS record.

However, making this interval too short opens the door for accidental mishaps. If a child zone makes an error and somehow the chain of trust is broken, it has until the RRSIG expires to fix the problem. This would recommend a longer signature lifetime. In DNSSEC these and other trade-offs have to be made.

The IETF DNSOP working group is currently addressing these parameters and their trade-offs. The current data came (and comes) from workshops and early test deployments.

Outlook and Prospects

Because DNSSEC requires some additions to the (cc/g)TLD registration process, it could be a while before ccTLDs are capable of deploying DNSSEC. If the protocol is completed this year (2004), it will probably take a few years before registries can advertise DNSSEC domain names.

It is important to consider what DNSSEC actually wants to accomplish; it makes spoofing attacks in the DNS visible—and nothing more. It is not a PKI with all the extra features because key revocation is, for instance, not implemented in DNSSEC. Seen in this light, the protection of private keys in DNSSEC is important, but when a private key is compromised we are just back to plain old DNS.

On the other hand, because DNSSEC does introduce cryptographic material in the DNS and allows for the addition of other (non-DNS) keys, some interesting possibilities emerge. Many technologies on the Internet want to have some kind of simple key distribution mechanism in place; for example: SSH and IPSec. What DNSSEC promises is a system in which we can validate the SSH key from an unknown host with only one key. If the validation is successful, we are quite certain the SSH host key comes from the host from which it claims to come. We get this without any extra effort or cost (from a client's perspective at least). The possibilities are probably endless.

References

[1] Roy Arends, Rob Austein, Dan Massey, Matt Larson, and Scott Rose, "DNS Security Introduction and Requirements," Work In Progress,
http://www.ietf.org/internet-drafts/draft-ietf-dnsext-dnssec-intro-10.txt

[2] Roy Arends, Rob Austein, Dan Massey, Matt Larson, and Scott Rose, "Resource Records for the DNS Security Extensions," Work In Progress,
http://www.ietf.org/internet-drafts/draft-ietf-dnsext-dnssec-records-08.txt

[3] Roy Arends, Rob Austein, Dan Massey, Matt Larson, and Scott Rose, "Protocol Modifications for the DNS Security Extensions," Work In Progress,
http://www.ietf.org/internet-drafts/draft-ietf-dnsext-dnssec-protocol-06.txt

[4] DNS and BIND Talk Notes:
http://www.tfug.org/helpdesk/general/dnsnotes.html

[5] R. Gieben, "DNSSEC in NL,"
http://www.miek.nl/publications/dnssecnl/index.html

[6] BIND9, Berkeley Internet Name Domain, Version 9:
http://www.isc.org/sw/bind/

[7] R. Gieben, "Chain of Trust: The parent-child and keyholder-keysigner relations and their communication in DNSSEC," NIII report CSI-R0111:
http://www.cs.kun.nl/research/reports/info/CSI-R0111.html
http://www.miek.nl/publications/thesis/CSI-report.ps

[8] R. Gieben and T. Lindgreen, "Parent's SIG over Child's KEY,"
http://www.nlnetlabs.nl/dnssec/dnssec-parent-sig-01.txt

[9] O. Kolkman and R. Gieben, "DNSSEC Operational Practices," Work In Progress,
http://www.ietf.org/internet-drafts/draft-ietf-dnsop-dnssec-operational-practices-01.txt

[10] Netscape Communications Corporation, "Introduction to Public-Key Cryptography,"
http://developer.netscape.com/docs/manuals/security/pkin/contents.htm

MIEK GIEBEN graduated in Computer Science in 2001 from the University of Nijmegen (Netherlands) on the subject of DNSSEC. He has been employed by NLnet Labs since that time. He has been using Linux and the Internet since 1995. Currently he is involved in DNSSEC deployment and has co-written parts of NSD2 (which is now fully DNSSEC aware). His personal home page can be found at **http://www.miek.nl/**. The home page of NLnet Labs can be found at **http://www.nlnetlabs.nl/**.
E-mail: **miekg@atoom.net**

Book Review

Network Management

Network Management, MIBs and MPLS by Stephen B. Morris, ISBN 0131011138, Prentice Hall, June 2003.

Few people would question the need for good network management, and books about the *Simple Network Management Protocol* (SNMP) have been circulating for more than ten years now. But the key differentiator of this book is well recognized in its title—it's about SNMP in the context of a *Multiprotocol Label Switching* (MPLS) network. MPLS is now recognized as the convergence technology, and an increasing number of mission-critical services are being deployed over it. World-class network management is vital to keep these services running to the "five nines" level we've all come to expect.

Organization

In this book, Stephen Morris offers a very approachable and comprehensive look at SNMP and the methodology behind the all-important *Management Information Base* (MIB). The first chapter gives the obligatory justification for network management and sets the scene nicely for the rest of the book.

It's amazing to think that SNMP has been around since the late 1980s, and yet if you ask any MPLS operations person, the odds are that person is still using a *Command-Line Interface* (CLI) to actually configure boxes. CLI is a man-machine interface, not a machine-machine interface like SNMP. Even centralized provisioning platforms, such as the former Orchestream (now Metasolve) VPN Manager, simply created a friendly *Graphical User Interface* (GUI) front end for the provisioning procedure, and then ran CLI scripts frantically in the background. The drawbacks of CLI configuration are too numerous to list here, but the basic solution to the problem is to create a scalable and secure machine-to-machine interface. In the IP world the candidate technology for this is SNMPv3, and Morris discusses both the MIB structure (the key to scalability) and the security model in Chapter 2. Because premium MPLS-based services demand secure and robust provisioning, SNMPv3 is the technology of choice.

Chapter 3 describes what Morris calls the "Network Management Problem," although in fact this is described as a whole set of problems, some of which are caused by deficiencies in the SNMP architecture, whereas others are caused by the scale and pace of operations in a modern network. A specific problem that Morris addresses very sensibly is the way that the rapid pace of network technology development impacts the ability to manage these networks. In other words, new technologies tend to appear too quickly for management mechanisms to be optimized for these protocols. To solve this problem, Morris (a software engineer by training) presents a series of "Linked Overviews" (these describe the properties of a given network technology—MPLS, *Asynchronous Transfer Mode* (ATM), etc.—in a procedural framework. In essence this is a kind of recipe for the software developer. In addition, the text is liberally sprinkled with "Developers Notes" that I'm sure will provide invaluable help for people trying to write management system code.

Chapter 4 then takes the approach of solving the "Network Management Problem" to a higher, and perhaps longer-term level, with the proposed development of smarter network management components and more integrated data frameworks. This culminates in a description of *Directory Enabled Networking,* a technology that seemed to flower briefly in the context of network management a few years ago, but then was buried when the telecom recession hit the industry. My own feeling is that the time is right for a rebirth of this approach in modern, converged networks.

Chapter 5 looks at some real *Network Management System* (NMS) issues, using the HP OpenView Network Node Manager as a worked example. Morris is quick to point out that this is not an endorsement of the product, but because it is the most well-known and widely used product in this class, it is the logical choice.

Chapters 6 and 7 look at software components, and Morris's background in software development shines through here in the level of detail, coupled with well-structured explanations.

Chapter 8 describes a very useful case study of using SNMP to provision a tunnel through an MPLS network—a task that is typically performed today using crude CLI techniques.

Chapter 9 contrasts theory and practice in network management, and deals with the loose ends of various topics such as end-to-end security and the integration of a third-party *Open Source Software* (OSS) using standardized northbound *Element Management System* (EMS) interfaces.

Recommended

Overall this is an excellent book that really does deliver what it claims—a comprehensive and practical look at the latest SNMP technologies and techniques. In this regard it stays highly focused, and doesn't waste time with irrelevant discussion on other topics. For example, at first I was disappointed to note that only a page or two of brief explanation is devoted to topics such as *Common Object Request Broker Architecture* (CORBA) and *Extensible Markup Language* (XML). But in the context of what this book is trying to tell us, it makes perfect sense. Each of these topics really needs it own book to cover the topic in similar detail to Morris's work.

Similarly, if you're expecting a description of emerging IP/MPLS *Operations, Administration, and Maintenance* (OA&M), then this book is not for you. Again, I would defend Morris's use of Occam's Razor because OA&M protocols are usually demanded by network staff, and not by OSS operatives. In my own opinion, this situation will gradually change in the next few years, as OA&M is recognized as the "eyes and ears" of the OSS. Perhaps this would be a good place for Mr. Morris to start his next book.

—Geoff Bennett, Heavy Reading
bennett@heavyreading.com

Fragments

Cooperative Support for Global IPv6 Deployment

The *Regional Internet Registries* (RIRs), the *IPv6 Task Forces* and the *IPv6 Forum* are working in cooperation to support global IPv6 deployment.

The four RIRs, APNIC, ARIN, LACNIC and the RIPE NCC, are responsible for the management of global Internet numbering resources, including IPv4 and IPv6 address space, throughout the world. The RIRs confirm their commitment and continued support towards the deployment of IPv6 in cooperation with the IPv6 Task Forces and with the support of the IPv6 Forum.

The IPv6 Task Forces are focused on rapid IPv6 deployment. They see the adoption of IPv6 by industry, governments, schools and universities is particularly important. The extra address space offered by IPv6 will facilitate the deployment of widespread "always-on" Internet services including broadband access for all. In addition, IPv6's built-in encryption will help improve Internet security and is promoted by many government institutions globally.

The cooperation among the RIRs and the IPv6 Task Forces includes key aspects such as:

- Supporting awareness, education and deployment of IPv6;
- Disseminating information on the progress of IPv6 deployment;
- Encouraging dialogue and ensuring the necessary cooperation between all involved parties;
- Benchmarking IPv6 deployment progress;
- Supporting the adoption of Domain Name Service infrastructure necessary for IPv6;
- Encouraging the participation of all those who are interested in the IPv6 policy development process.

This cooperative effort between the RIRs and the IPv6 Task Forces recognises that while IPv4 address space will be available for many years, new users and usages of the Internet have the potential to rapidly increase the utilisation of IPv4 address space. With the advent of multiple always-on devices, wireless handhelds and 3G mobile handsets, the Internet community needs to prepare for a sharp increase in IP address space utilisation. In order to prevent future operational problems, the global rollout of IPv6 is essential for enabling the development and adoption of new applications and services.

The rollout of IPv6 on this scale requires significant preparation, particularly in terms of training and planning. The RIRs and the IPv6 Task Forces encourage early evaluation by network operators and industry players, in order to promote the necessary technical dialogue and to facilitate widespread adoption. *Internet Service Providers* (ISPs) can already deploy IPv6 in non-disruptive ways that do not require additional investment while providing added value to their customers.

"The RIPE NCC has supported IPv6 from an early stage. We are committed to ensuring that IPv6 resources are provided to RIPE NCC members whenever they are required. We will continue to use the long-established system of address distribution where IP addresses are allocated according to demonstrated need wherever that need is demonstrated," stated Axel Pawlik, Managing Director of the RIPE NCC. "The RIPE NCC is already providing IPv6 training to our members and other tools required to facilitate IPv6 deployment," he added.

Jordi Palet, Founding Member of the EU IPv6 Task Force and co-chair of the IPv6 Forum's Awareness and Education Working Group, sees the formalisation of this cooperative support of IPv6 deployment as an important development. "This cooperative effort ensures the global recognition of the strategic importance of IPv6 in enabling the continued development of the Internet and the worldwide information society. This ongoing coordination will have a positive global benefit for end users and the industry, by reinforcing the resilience of the Internet while allowing for the development of ever-improving applications and services," he said.

Paul Wilson, APNIC Director General, noted that significant advances have been taking place in all the RIR regions with respect to IPv6 allocation and policy. "The RIRs are already working with the IANA and large ISPs to facilitate the delegation of large blocks of IPv6 address space," he stated. "In the Asia Pacific region, a number of countries are taking the lead in terms of IPv6 deployment, and APNIC will continue to offer its support in these areas, and elsewhere, to allow the entire region to benefit from IPv6."

"In the ARIN region, we have received clear direction from the community to make all necessary preparations for IPv6 deployment. This includes work on the allocation policies and procedures, as well as making our own services available via IPv6," stated John Curran, Acting President of ARIN

"LACNIC is involved in the formation of the Latin American and Caribbean IPv6 Task Force and is active in encouraging the participation of its members and the community in IPv6 deployment and policy, and our services are already available over IPv6," said Raúl Echeberría, CEO of LACNIC.

"This global cooperation signals another historic milestone to further accelerate take-up of IPv6 for the global good," applauded Latif Ladid, President of the IPv6 Forum.

"The North American IPv6 Task Force supports the worldwide collaboration with the RIRs to further support the deployment of IPv6 and the next generation Internet mobile society using IPv6," stated Jim Bound, Chair NAv6TF and IPv6 Forum CTO.

As an IPv6 Forum Board member and an ICANN Address Council member, Takashi Arano of the Asia Pacific IPv6 Task Force steering committee supports this collaboration. "Address management, which the RIRs are in charge of, is one of the crucial components for the commercial deployment of IPv6 and its stable operation."

"I hope collaboration between IPv6 Task Forces and the RIRs will result in the advent of an IPv6-powered 'everything-everywhere-every time' networking world," he stated.

IPv6 is a new version of the data networking protocols on which the Internet is based. The *Internet Engineering Task Force* (IETF) developed the basic specifications during the 1990s. The primary motivation for the design and deployment of IPv6 was to expand the available "address space" of the Internet, thereby enabling billions of new devices (PDAs, cellular phones, appliances, etc.), new users and "always-on" technologies (xDSL, cable, Ethernet-to-the-home, fibre-to-the-home, Power Line Communications, etc.).

The existing IPv4 protocol has a 32-bit address space providing for a theoretical 232 (approximately 4 billion) unique globally addressable network interfaces. IPv6 has a 128-bit address space that can uniquely address 2^{128} (340,282,366,920,938,463,463,374,607,431,768,211,456) network interfaces.

The *European IPv6 Task Force* is a volunteer organisation, with over 500 members, open to all the interested parties in advancing the IPv6 deployment in the European region, in cooperation with the rest of the world and other related entities. Further information is available on the IPv6 Task Forces website: **http://www.ipv6tf.org**

Four RIRs exist today. They provide number resource allocation and registration services that support the operation of the Internet globally. The RIRs are independent, not-for-profit organisations that work together to meet the needs of the global Internet community. They facilitate direct participation by all interested parties and ensure that the policies for allocating Internet number resources (such as IP addresses and *Autonomous System Numbers*) are defined by those who require them for their operations.

The RIRs ensure that number resource policies are consensus-based and that they are applied fairly and consistently. The RIR framework provides a well-established combination of bottom-up decision-making and global cooperation that has created a stable, open, transparent and documented process for developing number resource policies.

The RIR framework contributes to the common RIR goal and purpose of ensuring fair distribution, responsible management and effective utilisation of number resources necessary to maintain the stability of the Internet. The RIRs currently consist of:

APNIC: *Asia Pacific Network Information Centre*
http://www.apnic.net

ARIN: *American Registry for Internet Numbers*
http://www.arin.net

LACNIC: *Latin American and Caribbean Internet Addresses Registry*
http://www.lacnic.net

RIPE NCC: *RIPE Network Coordination Centre*
http://www.ripe.net

The *IPv6 Forum* is a world-wide consortium of over 160 leading Internet service vendors, National Research & Education Networks and international ISPs, with a clear mission to promote IPv6 by improving market and user awareness, creating a quality and secure New Generation Internet and allowing world-wide equitable access to knowledge and technology. The key focus of the IPv6 Forum today is to provide technical guidance for the deployment of IPv6. IPv6 Summits are hosted by the IPv6 Forum and staged in various locations around the world to provide industry and market with the best available information on this rapidly advancing technology. **http://www.ipv6forum.org**

The *North American IPv6 Task Force* is an all-volunteer non-vendor/service/provider or other entity interest with the IPv6 mission of assisting the North American geography as sub task force of the IPv6 Forum for deployment, education, awareness, technical analysis/direction, transition analysis, political/business/economic/social analysis support and other efforts as required. The members see IPv6 as more important than their own self-interests. **http://www.nav6tf.org**

Upcoming Events

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in Kuala Lumpur, Malaysia, July 19–23, 2004, and in Cape Town, South Africa, December 1–5, 2004. For more information see: **http://www.icann.org**

ICANN and *The International Telecommunications Union* (ITU) will be jointly hosting a workshop on *country code Top Level Domains* (ccTLDs), in Kuala Lumpur on 24 July. The purpose of this joint ICANN/ITU-T open workshop is to focus on the operation and practical operational issues facing the ccTLDs and to give the opportunity for ccTLD operators and ITU Member States to share their experiences. The Workshop is not a policy meeting, but rather it is intended as a forum for the exchange of views and discussions. Written presentations are encouraged, but not required. Written presentations can be submitted to **ICANN-ITU-T-Workshop@icann.org**. Additional information can be found at the ITU-T website: **http://www.itu.int/ITU-T/worksem/cctld/kualalumpur0704/index.html**

The IETF will meet in San Diego, CA, August 1–6, 2004 and in Washington, DC, November 7–12, 2004. For more information, visit: **http://ietf.org**

Useful Links

The following is a list of Web addresses that we hope you will find relevant to the material typically published in the IPJ.

- The *Internet Engineering Task Force* (IETF). The primary standards-setting body for Internet technologies. **http//:www.ietf.org**
- *Internet-Drafts* are working documents of the IETF, its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are not an archival document series.

These documents should *not* be cited or quoted in any formal document. Unrevised documents placed in the Internet-Drafts directories have a maximum life of six months. After that time, they must be updated, or they will be deleted. Some Internet-Drafts become RFCs (see below). **http://www.ietf.org/ID.html**

- The *Request for Comments* (RFC) document series. The RFCs form a series of notes, started in 1969, about the Internet (originally the ARPANET). The notes discuss man aspects of computer communication, focusing on networking protocols, procedures, programs, and concepts but also including meeting notes, opinion, and sometimes humor. The specification documents of the Internet protocol suite, as defined by IETF and its steering group the IESG, are published as RFCs. Thus, the RFC publication process plays in important role in the Internet standards process. **http://www.rfc-editor/org/**
- The *Internet Society* (ISOC) is a non-profit, non-governmental, international, professional membership organization. **http://www.isoc.org**
- The *Internet Corporation for Assigned Names and Numbers* (ICANN) "...is the non-profit corporation that was formed to assume responsibility for the IP address space allocation, protocol parameter assignment, domain name system management, and root server system management functions." **http://www.icann.org**
- The *North American Network Operators' Group* (NANOG) "...provides a forum for the exchange of technical information, and promotes discussion of implementation issues that require community cooperation." **http://www.nanog.org**
- The *Regional Internat Registries* (RIR) provides IP address block assignments for Internet Service Providers and others. See page 33 for links to APNIC, ARIN, LACNIC and RIPE NCC.
- The *World Wide Web Consortium* (W3C) "...develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential as a forum for information, commerce, communication, and collective understanding." **http://www.w3.org**
- The *International Telecommunication Union* (ITU) "... is an international organization within which governments and the private sector coordinate global telecom networks and services." **http://www.itu.int**

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

September 2004 Volume 7, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Network Address Translators (NATs) were designed to allow multiple devices in a private address realm to dynamically share a single public IP address. NATs are widely deployed in today's Internet. They provide an effective way of IPv4 address conservation while simultaneously offering some level of security because individual IP addresses on the "inside" are hidden from the "outside," or global Internet. But NATs also present a challenge to existing Internet applications that may depend on globally unique IP addressing for proper operation. To further complicate matters, not all NATs are created equal, leading to unpredictable behavior. This edition of IPJ is almost entirely devoted to an in-depth look at NATs. Geoff Huston looks inside the NAT, and explains the complexities behind each variation of NAT implementation. It seemed only natural that he would name such an exposé "Anatomy."

Many IPJ subscriptions had an official expiration date of September 30, 2004, but I am pleased to report that all these subscriptions have been extended for another year. You should still make sure your delivery address and e-mail is up-to-date in our database by using the link at **www.cisco.com/ipj** or sending e-mail to **ipj@cisco.com** with your updated information.

If you're hungry for even more networking-related reading material, look at the Internet Society's publication page at **http://isoc.org/pubs/**. Here you will find The ISP Column, Member Briefings, Articles of Interest, and links to other material.

We didn't have room for a book review in this issue, but we have several in store for future editions. If you'd like to contribute a book review for publication in IPJ, please contact me.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Anatomy: A Look Inside Network Address Translators

by Geoff Huston, APNIC

Over the past decade numerous IP-related technologies have generated some level of technical controversy. One of these is the *Network Address Translator,* or NAT. This article describes the inner workings of NATs in some detail, and then looks at the issues that have accompanied the deployment of NATs in the Internet that appear to have fueled this technical controversy. NATs are a very widespread feature of today's Internet, and this article attempts to provide some insight as to how they operate, why there is such a level of technical controversy about NATs, and perhaps some pointers to what we have learned about technology and the process of standardization of technology along the way.

NAT Motivation

The first RFC document describing NATs was by Kjeld Egevang and Paul Francis in 1994[1]. The original motivation behind the NAT work was based on efforts in the early 1990s associated with a successor protocol to IPv4. The overall effort of a successor protocol to IPv4 was to devise a protocol that would directly address the issues of accelerating address consumption in IPv4 that appeared to be leading to the prospect of imminent address exhaustion. Although IPv4 was capable of uniquely addressing some 4.4 billion devices, it was evident by as early as 1992 that the world was heading down a path of very intensive deployment of devices that included communications capabilities, and that IPv4 was not going to be able to extend across the full range of future device deployment. The objective with NAT was to define a mechanism that allowed IP addresses to be shared across numerous devices. In addition, it was intended that NATs could be deployed in a piecemeal fashion within the Internet, without causing changes to hosts or other routers. Other forms of address-sharing technologies relied on intermittent connectivity, whereas NATs were intended to allow a collection of connected devices to share an address pool dynamically. The original RFC portrays this approach as being a measure that can "provide temporarily relief while other, more complex and far-reaching solutions are worked out."

So, as documented, the original intent of NATs was to be a possible short-term response to address exhaustion while longer-term solutions were being devised. NATs were also intended to be unmanaged devices that are transparent to end-to-end protocol interaction, requiring no specific interaction between the end systems and the NAT device.

A decade later NATs are attaining a status of near-ubiquitous deployment across the Internet, and although IPv6 has been defined and deployment is commencing, NATs appear to be a very well-entrenched part of the network landscape. And, for the most part, NATs continue to function as unmanaged devices.

They can be transparent to some forms of protocol interaction, but, as the voice-over-IP folks are finding out, they can be very obvious to the point of being highly disruptive to other forms of protocol operation.

NAT Operation

The operation of NATs is deceptively easy to describe in general terms. They are active units placed in the data path, usually as a functional component of a border router or site gateway. NATs intercept all IP packets, and may forward the packet onward with or without alteration to the contents of the packet, or may elect to discard the packet. The essential difference here from a conventional router or a firewall is the discretional ability of the NAT to alter the IP packet before forwarding it on. NATs are similar to firewalls, and different from routers, in that they are topologically sensitive. They have an "inside" and an "outside," and undertake different operations on intercepted packets depending on whether the packet is going from inside to outside, or in the opposite direction.

NATs are IP header translators, and, in particular, NATs are IP *address translators*. The header of an IP packet contains the source and destination IP addresses. If the packet is being passed in the direction *from* the inside *to* the outside, a NAT rewrites the source address in the packet header to a different value, and alters the IP and TCP header checksums in the packet at the same time to reflect the change of the address field. When a packet is received *from* the outside destined *to* the inside, the destination address is rewritten to a different value, and again the IP and TCP header checksums are recalculated (Figure 1). The "inside" does not use globally unique addresses to number every device within the network served by the NAT. The inside (or "local") network may use addresses from private address blocks, implying that the uniqueness of the address holds only for the site. Let's look at this using an example.

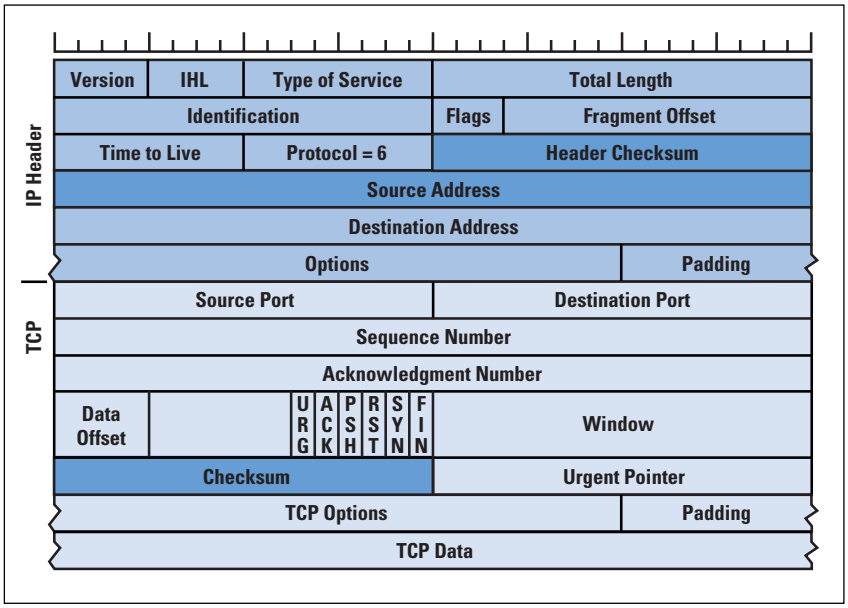
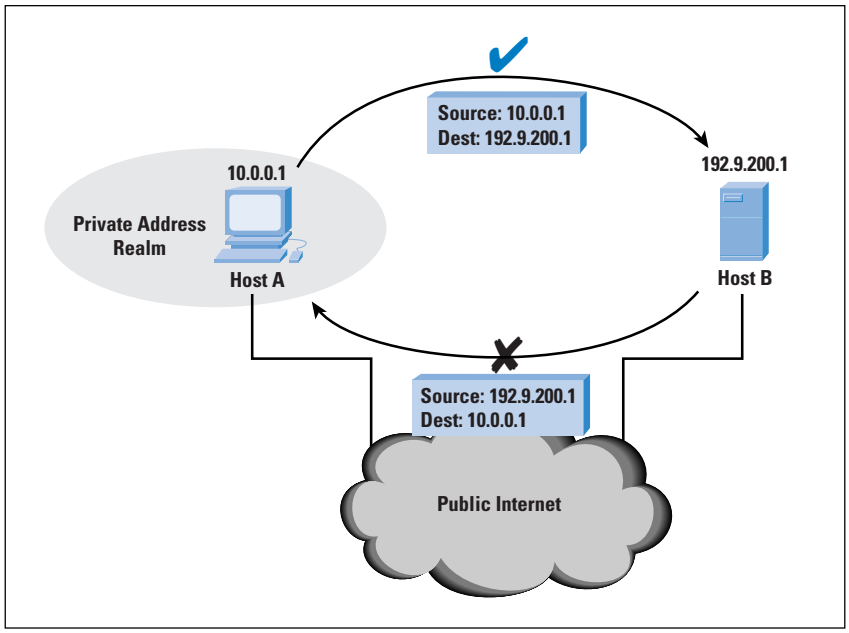


Figure 1: TCP/IP Header Fields Altered by NATs (Outgoing Packet)

As shown in Figure 2, how can local (private) host A initiate and maintain a TCP session with remote (public) host B? Host A first uses the *Domain Name System* (DNS) to find the public IP address for host B, and then creates an IP packet using host B's address as the destination address and host A's local address as the source, and passes the packet to the local network for delivery. If the packet was delivered to host B without any further alteration, then host B would be unable to respond. The public Internet does not (or should not at any rate!) carry private addresses, because they are not globally unique addresses.

Figure 2: Public/Private Communication



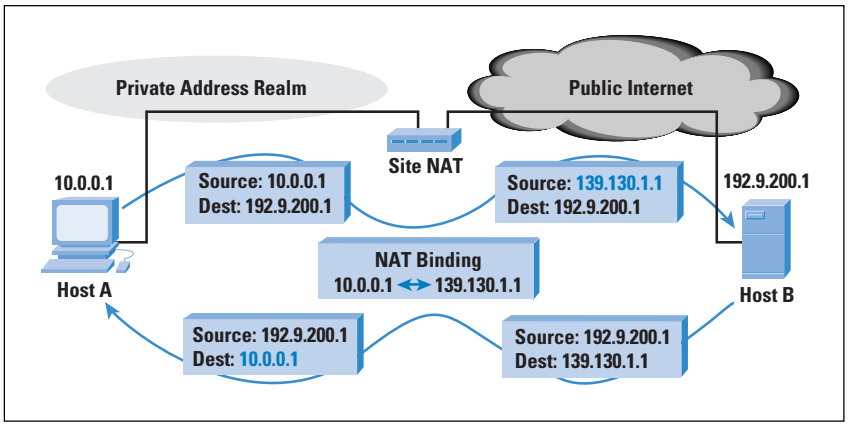
With a NAT between hosts A and B, the NAT intercepts host A's outgoing packet and rewrites the source address with a public address. NATs are configured with a pool of public addresses, and when an "inside" host first sends an outbound packet, an address is drawn from this pool and mapped as a temporary alias to the inside host A's local address. This mapped address is used as the new source address for the outgoing packet, and a local session state is set up in the NAT unit for the mapping between the private and the public addresses.

After this mapping is made, all subsequent packets within this application stream, from this internal address to the specified external address, will also have their source address mapped to the external address in the same fashion.

When an incoming packet arrives on the external interface, the destination address is checked. If it is one of the NAT pool addresses, the NAT box looks up its translation table. If it finds a corresponding table entry, the destination address is mapped to the local internal address, the packet checksums are recalculated, and the packet is forwarded. If there is no current mapping entry for the destination address, the packet is discarded.

The mode of operation of a NAT is shown in Figure 3. So, continuing our example, the local host at address A is directing packets to the external server host at address B. Because the NAT is in the path, the NAT has altered the packets so that address A is translated to address X. Host A is aware that it is communicating with host B, and from host A's perspective this is a normal session. Host B believes that it is communicating with a host at address X, and is entirely unaware of address A. From host B's perspective this is a normal session with a host at address X.

Figure 3: NAT Traversal

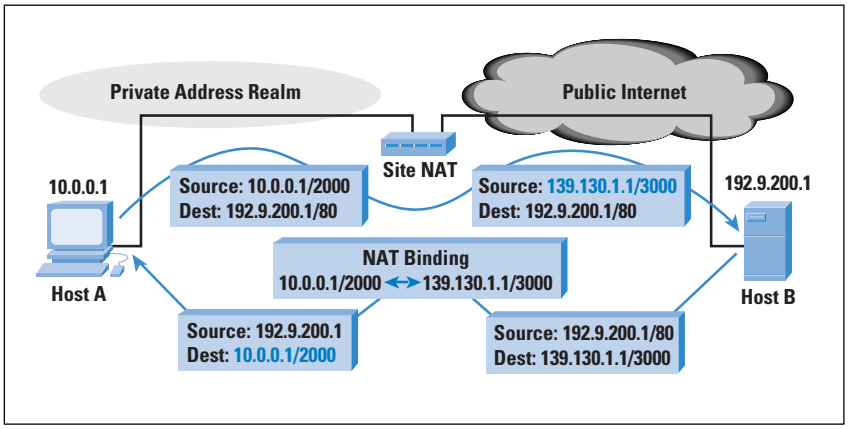


Dynamically created mapping entries (or "bindings") are typically maintained by the NAT with a *timer*. If no packets that use the mapping are received by the NAT within a certain time window, then the binding is removed from the NAT and the public address is returned to the NAT pool.

NAPTs

A variant of the NAT is the *Port-Translating NAT*, or NAPT. This form of NAT is used in the context of TCP and *User Datagram Protocol* (UDP) sessions, where the NAT maps the local source address and source port number to a public source address and a public-side port number for outgoing packets. Incoming packets addressed to this public address and port pair are translated to the corresponding local address and port. Again, the binding is maintained by a NAT idle timer, and upon expiration of the timer the public address and port pair are returned to the NAT pool (Figure 4).

Figure 4: NAPT Traversal



Again the NAPT is attempting to be transparent in terms of providing a consistent view of the session to each end, using a symmetric binding of a local address and port pair to an external address and port pair.

A reasonable question to ask is: Why should NAPTs bother with port translation? Are straight address translations not enough? Surprisingly, NATs can be relatively profligate with addresses. If each TCP session from the same local host is assigned a different and unique external pool address, then the peak address demands on the external address pool could readily match or exceed the number of local hosts, in which case the NAT could be consuming more public addresses than if there were no NAT at all! NAPTs allow concurrent outgoing sessions to be distinguished by the combination of the mapped address and mapped port value. In this way each unique external pool address may be used for up to 65,535 concurrent mapped sessions.

For a while the terminology distinction between NATs and NAPTs was considered important, but this has faded over time. For the remainder of this article we use current terminology, and look at NATs and NAPTs together and refer to them collectively as "NATs."

NAT Behavior

The use of NATs involves two basic issues: One is that NATs make applications "brittle" in that NATs support a particular style of application operation, and if the application deviates in any way from this style then the application no longer works. The second is of much more concern, and that is that NATs differ from each other in quite fundamental ways. What works across one NAT may not work at all for another class of NAT. It has also been reported that NATs differ not only on a vendor-by-vendor basis, but even on a model-by-model basis within a single vendor's range of NAT units. The implication here is that such differences of behavior become a matter for discovery by applications rather than something applications can predict in advance. This section explores this behavioral aspects of NATs in further detail.

Symmetry and Sessions

NATs can manage address mapping in numerous ways, and many implementations of NATs use a form of binding termed a "symmetric" binding.

A *symmetric* binding is where the mapping of a local address to a public address is exclusively tied to the destination address used in the initial trigger outgoing packet for the lifetime of the binding. Incoming external packets with the mapped public address as their destination are translated to the local address only if the source address of the incoming packet matches the destination address of the original mapping. Multiple sessions to different public hosts may use the same mapped public address, or may use different public addresses for each session. This mapping is "endpoint" sensitive. Symmetric NATs represent a restricted model of operation, where each NAT binding represents a window through the NAT that is visible only to the destination host (Figure 5).

By comparison, a *full-cone* NAT allows any external host to use this opened window, where all incoming packets addressed to the mapped external address are translated to the mapped internal address and forwarded through the NAT. Symmetric NATs represent the most restrictive form of behavior, whereas full-cone NATs represent a far more permissive mode of operation.

In the context of NATs, this symmetric mode of operation refers to the session state 5-tuple, made up of Transport Protocol, the local IP address and port number, and the destination IP address and port number. When a session is opened from the local host to a remote service port on a remote host, then only that remote service can pass packets back through the NAT to the local host on that port. As with NATs, a full-cone NAT allows any remote service entity to direct packets back through the port window.

NATs can be further refined by having different behaviors for TCP and UDP transports. A NAT may behave in a symmetric manner for TCP sessions, and operate in a full-cone mode for UDP transactions. The variations in NAT behavior has led to an exercise in categorizing NAT behaviors and developing a discovery protocol whereby a pair of cooperating systems can discover if one or more NATs is on the network path between them, as well as attempting to establish the type of NAT.

Discovering NAT Behaviors and STUN

NAT behavior has not been the topic of any industry standardization efforts, and it should not be surprising to learn that, given that a range of possible NAT behaviors exist under certain conditions, the market contains NAT offerings that cover the full spectrum of possibilities. In the absence of common specifications or standards, implementers have been placed in the position of having to make some creative guesses as to what the "right" behavior should be under such circumstances. This is a significant problem for the application designer, given the prospect that in today's Internet any popular application must have a means of being able to function correctly in the face of one or more NATs on the path between two hosts that are communicating using the application.

One of the more pressing problems here is that NATs commonly enforce an application model where the local "hidden" host must initiate a transaction in order to create a window in the NAT to allow the packets of the remote host back into the local network.

Some applications may wish to undertake "referral," where the correspondent host on the external side may want to pass the externally presented address and port details of the local host to a third party in order to commence a further part of the transaction. Other application transactions may simply want to be initiated from the external side. Although this may have been thought of as a relatively obscure condition, it was brought into the forefront of attention when various forms of voice-over-IP and peer-to-peer applications gained popularity. In particular, the question of "how can the external side initiate a packet flow in the presence of a NAT?" has become increasingly important.

Given that the application needs to perform some additional gymnastics in such a case, there is the additional question that the application must answer, namely: "How does the application learn that there are NATs in the path in the first place?"

At this point the application is placed in the role of performing a forensic exercise of establishing whether or not its packets are being altered by one of more NATs when it attempts to establish an end-to-end packet transaction. If so, what types of implementation decisions have been made by the NAT in terms of the way in which packets are being systematically modified? In others words, what is the anatomy of the particular NATs that have been discovered along the path? This anatomy exercise is further complicated by the observation that NATs are silent devices, so the application cannot directly interrogate the NAT to establish its behavior. All that is left is a somewhat unsatisfying guessing game for the application. It is forced to send particular types of test packets through the NAT to some pre-defined counterpart on the other side. The application must then compare the self-view of the IP address and port number of the local host to the remote view of its IP address and port number, and then attempt to guess the nature of the systematic transforms that the NAT is applying.

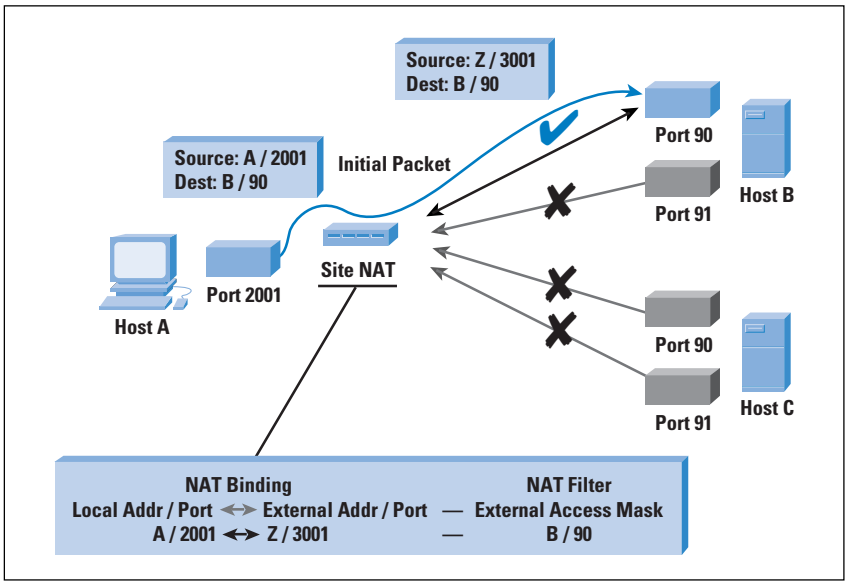
In the case of TCP it appears that the prevalent NAT behavior is that of a symmetric NAT based on address and port bindings. This implies that when the local host opens up a TCP session with a remote host, the NAT address and port bindings for the local host are coupled with the address and port of the destination host. Only packets with a source field of the destination host can pass packets back through the NAT to the TCP session of the local host. In other words, when a TCP session has been established within a NAT, only the two endpoints of the TCP session can access the NAT bindings, and attempts by others to direct packets to the external-side presented address and port meet with the NAT discard response. The fine-grained behavior of NATs with respect to TCP sessions can vary according to the amount of TCP state maintained by the NAT. At a basic level, the NAT can maintain a binding based on the local address and port and the remote address and port. The NAT also can keep the binding timer at a high value until a **`FIN`** exchange is observed, or until the session is reset through the **`RST`** flag being set, at which point the binding timer can be reduced to a very short interval. The NAT can also track the sequence number windows of the two sides and associated window sequence number scaling values and not adjust the binding timer of the session for TCP packets with sequence numbers outside the sequence number window with their **`FIN`** or **`RST`** flags set.

These NAT behaviors are based on the explicit signaling of changes in session state within the TCP packet exchange, and the consequent ability of the NAT to track the session state and adjust the associated binding timer in response to this state information. UDP is not so straightforward, because there is no explicit session state within a UDP packet exchange, and various NATs behave differently with respect to UDP-based bindings.

Various classes of NAT behavior relate to how UDP bindings are managed within a NAT. These have been classified into four types of behaviors[11]:

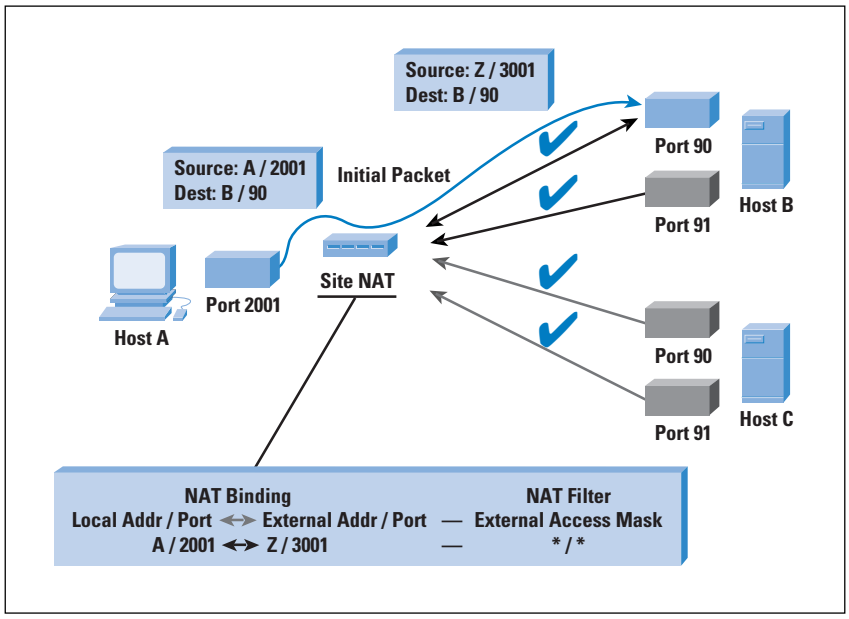
- *Symmetric:* We have already encountered the symmetric NAT, where the NAT mapping refers specifically to the connection between the local host address and port number and the destination address and port number and a binding of the local address and port to a public-side address and port. Any attempts to change any one of these fields requires a different NAT binding. This is the most restrictive form of NAT behavior under UDP, and it has been observed that this form of NAT behavior is becoming quite rare, because it prevents the operation of all forms of applications that undertake referral and handover.

Figure 5: Symmetric NAT



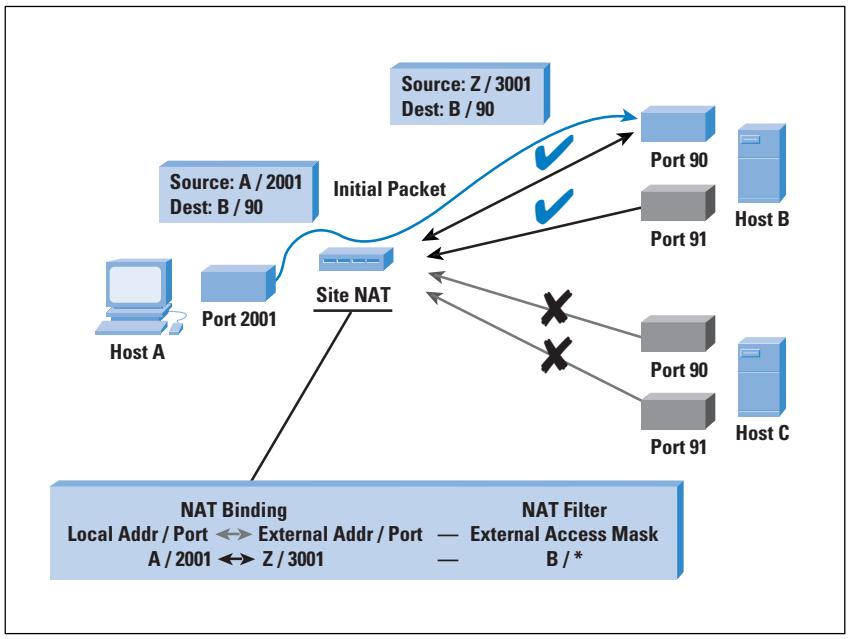
- *Full-cone:* A full-cone NAT is the least restrictive form of NAT behavior, where the binding of a local address and port to a public-side address and port, when established, can be used by any remote host on any remote port address. (Refer to Figure 6.)

Figure 6: Full Cone NAT



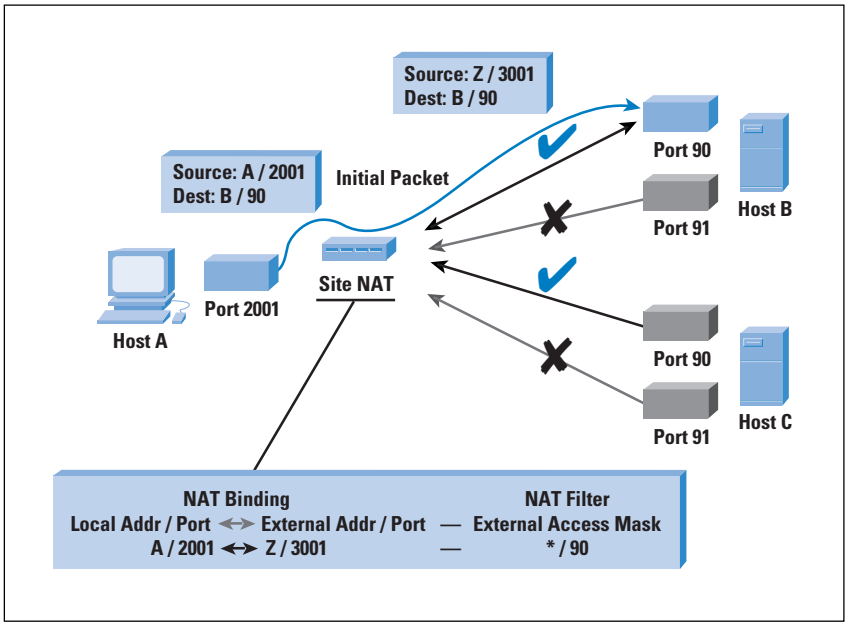
- *Restricted-cone:* A restricted-cone NAT is one where the NAT binding is accessible only by the destination host, although in this case the destination host can send packets from any port address after the binding is created. (Refer to Figure 7.)

Figure 7: Restricted-Cone NAT



- *Port-restricted-cone:* A port-restricted-cone NAT is one where the NAT binding is accessible by any remote host, although in this case the remote host must use the same source port address as the original port address that triggered the NAT binding. (Refer to Figure 8.)

Figure 8: Port-Restricted-Cone NAT



So can an application tell if one or more NATs are in the path, and, if so, what form of behavior the NAT is using? For this purpose the *Simple Traversal of UDP through NATs* (STUN) protocol has been developed[11]. STUN is a probe system that examines the interchange between a STUN client that may lie behind a NAT and a STUN server that is positioned on the public side of the NAT. The STUN-server host must be configured with two IP addresses, and the STUN itself should respond to queries on two UDP port numbers. The protocol is a simple UDP request-response protocol that uses embedded addresses in the data payload, and compares these addresses with header values in order to determine the type of NAT that may lie in the path between client and server.

The basic operation of STUN is a request-response protocol, using a common request of the form: "Please tell me what public address and port values were used to send this query to you."

STUN can be used to discover if a NAT is on the path between a client and server, and attempt to discover the type of NAT by a structured sequence of requests and responses. The client sends an initial request to the STUN server. If the public address and port in the returned response are the same as the local address, then the client can conclude that there is no NAT in the path between the client and the server. If the values differ, the client can conclude that there is a NAT on the path. STUN then uses subsequent requests to determine the type of NAT. One critical additional item of information returned by the STUN server in the initial response is an alternate IP address and port number that can also reach the same STUN server.

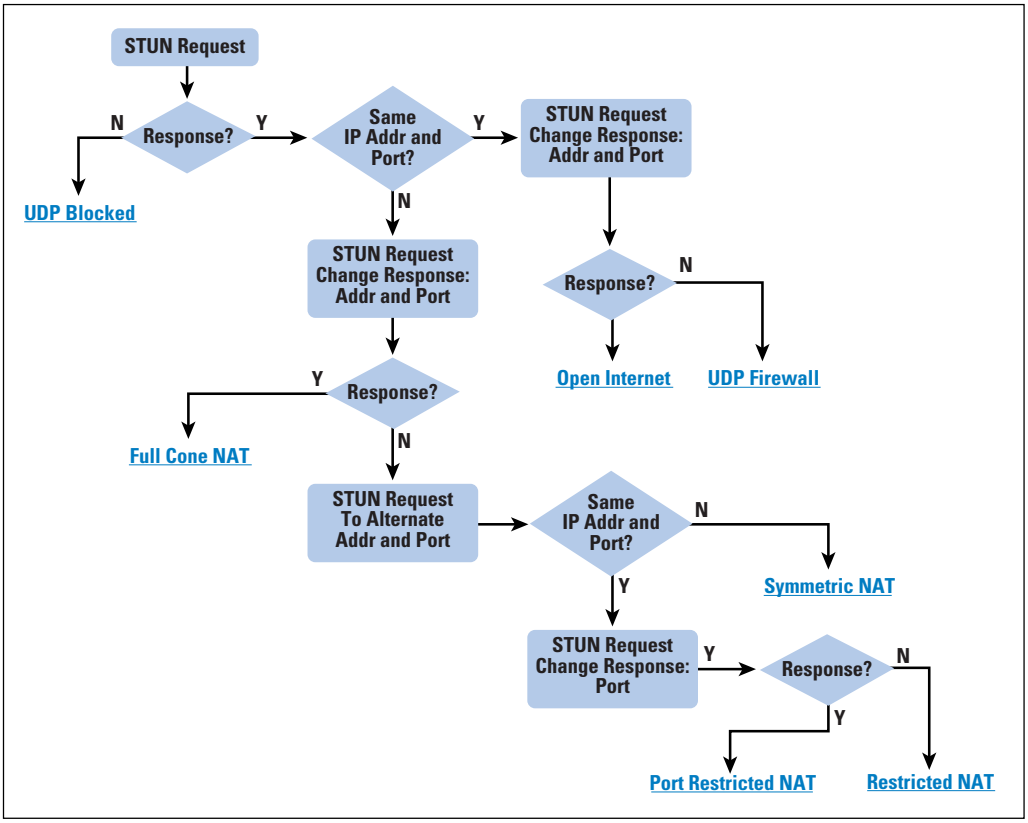
The second STUN request is directed to the same address and port as the initial request, but this time the request includes a control flag that requests the STUN server to respond using its alternate source address and port values. If the STUN client receives this alternate-sourced response, then it can conclude that it is behind a full-cone NAT. This is because the initial NAT binding of the local host address to the external presentation address can evidently be accessed by third-party external hosts.

If no response is received to the second request, then the STUN client sends the original probe request, but this time the request is addressed to the alternate destination address and port pair for the STUN client. If the returned address and port values relating to the new NAT binding are different from those of the first request, then the client can conclude that it is behind a symmetric NAT.

If the values are unaltered, then a further request can be made to determine the form of restricted-cone behavior. This fourth request includes a control flag to direct the STUN server to respond using the same IP address, but with the alternate port value. A received response indicates the presence of a port-restricted cone, and the lack of a response indicates the presence of a restricted cone.

Periodic exchanges between the STUN client and server can also discover the timer used by the NAT to maintain address bindings. Additional components of STUN are intended to provide some reasonable level of integrity in the packet exchange. A flowchart of a STUN-based NAT discovery process is shown in Figure 9.

Figure 9: NAT Discovery Process Using STUN



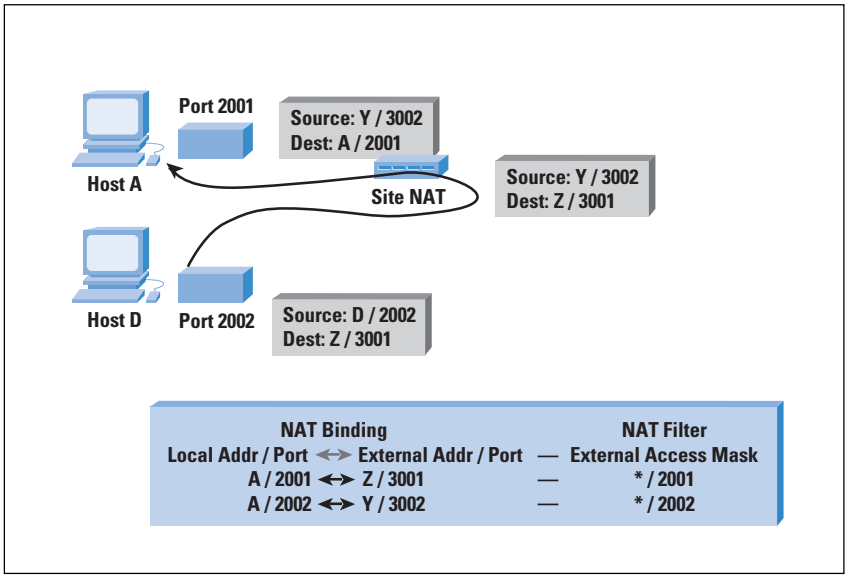
Further Behaviors: Hairpins and Determinism

It would be good if NAT behavior remained that simple. However, it does not, and some further tests on NATs reveal further differences in various NAT implementations[16].

The first area of difference is whether the NAT supports the so-called *hairpin* operation, where a local host directs a packet to the public address and port of an already mapped local host, or even to its own mapped address and port. If successful, then the NAT supports hairpin operation, where the NAT bindings, when created, are available to either side of the NAT. (Refer to Figure 10.)

Furthermore, the NAT may generate a binding for this operation—or not—thereby presenting the hairpin packet with an external address and port, indicating that an outbound binding has been performed in conjunction with the inbound binding, or with an internal address and port, indicating that only an inbound binding is being performed.

Figure 10: Hairpin NAT Operation



The second is in the general class of NAT determinism. Nondeterministic NATs change their binding behavior when a binding conflict of some sort occurs in the NAT. This is further based on the classification of whether "primary," "secondary," or even "tertiary" NAT behaviors differ. To explain primary, secondary, and tertiary behaviors, it is first noted that some NATs attempt to preserve the port address in the binding, so that the local source port and the externally bound port are the same whenever possible. This is the "primary" binding of the NAT. If another local host obtains a NAT binding using the same source port number, then the behavior of the NAT for this conflicting port binding may differ from that where the port number is preserved. The first conflict of port allocations in bindings is the "secondary" binding. In some cases the primary behavior is that of a full cone, or a restricted cone, while the NAT behaves in a symmetric fashion for the secondary instance where the port number has been mapped to a new value by the NAT.

A tertiary behavior occurs when a third binding is added to the NAT, because, again, the behavior of the NAT may be different for this binding.

It is also possible that the NAT may elect to preserve the binding in any case, and remove the current binding and replace it with a new binding that refers to the most recent packet that the NAT has processed.

All these behaviors can be classified as *nondeterministic,* in that the NAT behavior becomes one that is determined by the order of outbound traffic. The implication is that repetitions of the same STUN test at different times may produce different classifications of the type of NAT. The inference is that if an application uses STUN to determine the type of NAT in the path, and then selects a certain behavior based on this STUN-derived knowledge of the NAT type, nondeterministic NATs may behave differently between the STUN test and the application. The NAT response for a particular binding cannot be predicted in advance, and even when a binding state is established it may be disrupted or altered by subsequent traffic.

Another Approach to Classifying NATs

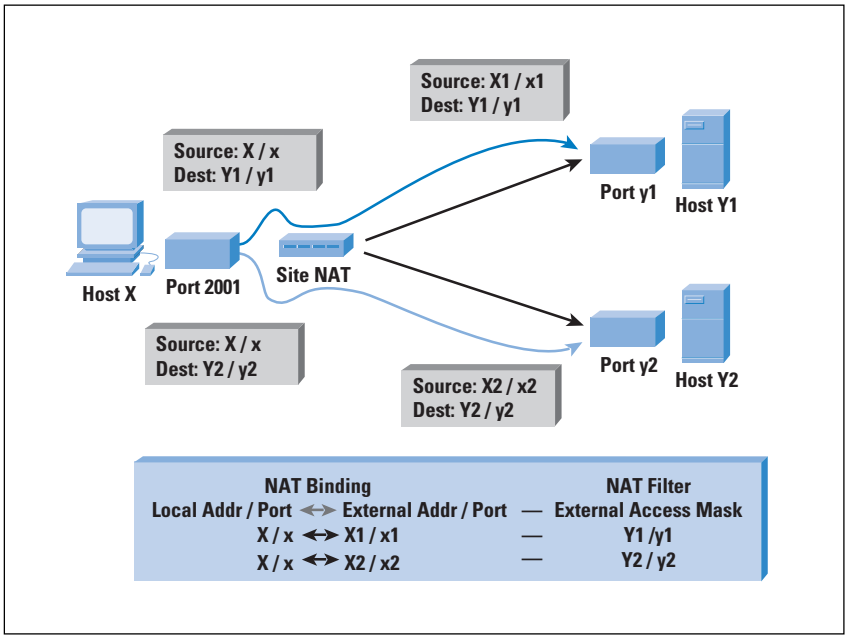
Further tests on NATs reveal that the various behaviors are yet more complex, and that different sequences of tests across a NAT will lead the test routine to come to different conclusions as to the type of NAT[13]. The key observation here is that NATs are the conjunction of two distinct behavior sets:

- *Binding,* or context-based packet translation: Detecting those packets that can be associated with a current binding and using that binding in a manner according to the logical direction of the packet to perform packet header transforms
- *Filtering,* or packet discard: Discarding those packets that cannot be associated with current bindings and discarding them

If a STUN-like test sequence was for a local host to send a packet to one destination and obtain a response of what NAT binding was used, and then to send a packet to a second destination and compare the results, the observation of the NAT using a different binding for each request may lead the tester to conclude that the NAT is a fully symmetric NAT. If the test sequence is for the NAT to send one packet to a destination and have the destination respond using a different source address, then the observation that the response packet is successfully delivered through the NAT back to the originating local host may lead the tester to the conclusion that the same tested NAT is some form of cone NAT.

The STUN approach classifies NAT behaviors on the basis of a single binding being established by the local host when contacting an external host, and then considers what constraints are placed on third-party external hosts as they attempt to access this initial binding. An adjunct to this approach is based on the local host establishing two bindings to two distinct external hosts, and looking for any relationship between these two bindings. (See Figure 11).

Figure 11: Outbound Connections from a Common Source



The behaviors of NATs under this condition can be classified under numerous behavioral aspects.

Binding

Binding behavior can be seen as the amalgam of three somewhat distinct design decisions, namely the manner in which a binding is generated, the behavior of the NAT in managing external ports used in bindings, and the manner in which expiration timers that govern the continued existence of the binding are refreshed.

NAT Binding Behavior:

- *Endpoint independent:* The NAT reuses the port binding for subsequent sessions initiated from the same internal IP address and port to any external IP address and port. This is analogous to a full-cone NAT.
- *Endpoint address dependent:* The NAT reuses the port binding for subsequent sessions initiated from the same internal IP address and port only for sessions to the same external IP address, regardless of the external port. This is a looser form of symmetric NAT, where the binding is created on the basis of the external address, rather than the external address and port.
- *Endpoint address and port dependent:* The NAT reuses the port binding for subsequent sessions initiated from the same internal IP address and port only for sessions to the same external IP address and port. This is a more precise form of UDP symmetry where the binding is available only to a single session, where a session is the 5-tuple of protocol, source address, source port, destination address, and destination port.

Port Binding Behavior:

- *Port preservation:* In addition to the differences in the binding between the two cases, the NAT may attempt to preserve the local port number, if possible. The terminology proposed here is port preservation to describe this NAT action.
- *Port overloading:* Some NATs attempt to undertake port preservation at all times, so that when a different local host establishes a binding using a port that is already being preserved, the new binding will usurp the existing binding. This behavior is proposed to be termed port overloading.
- *Port multiplexing:* The alternative to port overloading is use of the external entity to perform the demultiplexing of the port. In this case if two local systems use the same source port to send packets to two different external hosts, the NAT preserves the source port in the two bindings. If the NAT is using a single external address, the external view is two packets with the same source address and source port, sent to two different external addresses. The reverse packets have the same destination address and port, and the NAT determines the appropriate binding based on the source address and port in the reserve packets. This requires an endpoint address and port-dependant binding behavior. If two internal hosts are directing packets to the same external endpoint using the same source port addresses, then it is necessary for one of the sessions to use a binding with an altered port number. This could be considered as nondeterministic behavior.

Binding Timer Refresh:

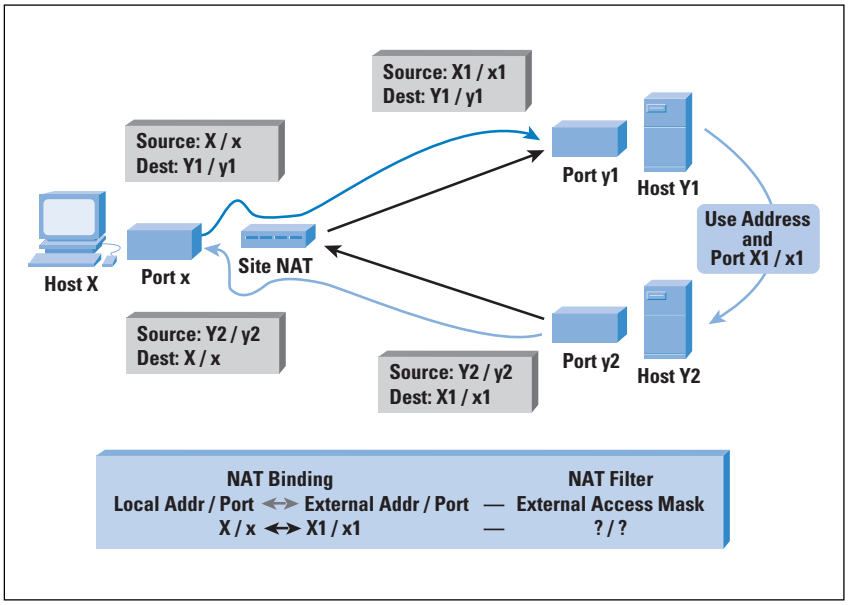
- *Bidirectional:* The NAT does not keep the binding active indefinitely, and normally removes the binding if there are no further packets that use the binding within a certain time period. However, there are variations in the classification of packets that the NAT considers as packets that reset the timer. In the case of bidirectional binding timer refresh, packets from either the local hosts or an external host that uses the NAT binding cause the NAT binding expiration time to be reset.
- *Outbound:* An outbound binding timer refresh NAT resets the expiration timer only when packets pass from the local host to the external host within the context of the binding. The implication is that a local host may have to use some form of keepalive operation to maintain a NAT binding in the face of an inbound UDP unidirectional traffic flow. Additionally, the expiration timer may be on a per-session basis, or may be on a per-binding basis if multiple sessions are associated to a single binding in the NAT.
- *Inbound:* As the name suggests, this is the opposite of the previous case, where only inbound packets cause the expiration timer of the binding to be refreshed.

- *Transport Protocol state:* Although these forms are useful in the case of UDP-based sessions, when the binding is based on a transport session (such as TCP), the NAT can base its binding timer refresh on the transport session state. For TCP this would infer a binding refresh time that is refreshed by any session packet in either direction (bidirectional), with the exception of packets with the TCP **`RST`** or **`FIN`** flags set. Although it would be an option to drop the NAT binding state when such packets are seen, this makes the NAT vulnerable to denial-of-service attacks by third-party injection of TCP **`RST`** packets, so there is some merit in using the binding timer for TCP sessions.

Filtering

The second phase of the test has two external hosts directing a probe to the same binding address, and classifying the behaviors based on what packets are filtered and discarded by the NAT (Figure 12).

Figure 12: Inbound Test



External Filtering:

- *Endpoint independent:* The NAT does not filter and discard packets that are addressed to the external part of the binding, irrespective of the source values in the packet. This is analogous to a full-cone NAT.
- *Endpoint address dependent:* The NAT filters and discards packets that are addressed to the external part of the binding, unless the source address of the packet matches the destination address used in the binding. This is analogous to a restricted-cone NAT.
- *Endpoint address and port dependent:* The NAT filters and discards packets that are addressed to the external part of the binding, unless the source address and port number of the packet matches the destination address used in the binding. This is analogous to a port-restricted-cone NAT or a symmetric NAT.

External Filtering Timer Refresh:
As with binding timers, these timers can be refreshed bidirectionally, inbound or outbound.

NAT Behaviors

The approach of carefully identifying the areas where NAT behaviors differ and classifying these behavioral differences in a methodical manner is one that has the potential to at least allow us to use the same sets of words when we talk about NAT behaviors, and hopefully also refer to the same set of actual behaviors when we use the same descriptions. The original approach with the STUN work used the terms *symmetric, full-cone,* and forms of *restricted-cone* to describe variations of NAT behaviors. Experience with this form of classification has exposed further variations in NAT behaviors, and this has led to a form of NAT classification that first uses a delineation of binding and filtering behaviors, and then classifies the various ways in which these bindings and filters are maintained within the NAT. Additional classification attributes include whether the NAT supports hairpin connections or not and whether it operates in a deterministic or nondeterministic manner.

This exercise is not another study in comparative taxonomies. A NAT has no standard way in which to advertise its presence, nor does it have any standard way in which to advise protocols or applications of the particular behaviors it applies to packets being passed through the NAT. In the absence of such explicit advertisements of the presence of a NAT, it is left to the application to make the necessary adjustments that allow it to function in the presence of NATs. The aim of behavioral classification is to associate test sequences that expose the presence of a NAT, and to determine its behavior. This allows applications to invoke a test procedure that exposes a particular choice of behaviors of a NAT implementation, and then allows the application to invoke a mode of operation that can operate across the particular NAT.

The choices available to application environments include the use of *agents* as session initiation intermediaries, where the endpoints make initial contact through agents, who then assist in passing binding information to the endpoints, allowing them to directly communicate. Other forms of application behavior need to be invoked when the NAT is endpoint address and port dependant for both binding and filtering. Different application responses are applicable when one endpoint is behind a NAT and when both endpoints are behind NATs. A typical application response in this latter case where both endpoints are behind highly restrictive NATs is for the endpoints to use agents as session intermediaries, so that the application payload is then passed through the intermediaries because an end-to-end pair of NAT bindings cannot be established.

Living in a NAT World

It would be a reasonable conclusion to draw from the previous sections that we are left in the somewhat unsatisfying position of observing that there is near-universal deployment in today's Internet of NAT devices that do not conform to any particular well-defined behavior set. NAT behavior varies across implementations, and NATs have no ability to disclose their particular behaviors to applications that are attempting to compensate for their presence in the path. It is extremely challenging for applications to reliably predict the behavior of the NATs that lie in the path, and more so in the face of multiparty applications, such as interactive game environments, where the application is attempting to understand the level to which this silent intermediary is capable of supporting a relatively promiscuous NAT binding state in terms of external entities that wish to send packets to the local host, and communicate between themselves about the local host as a single entity.

NATs, Client-Server, Peer-to-Peer, and Multiparty Applications

NATs, as a class of devices, have strong associations with a client-server model of communications. As long as all the servers have a consistent external visibility, with stable addresses in terms of an IP address and port number, and as long as clients initiate connections with servers in a fixed two-party communications model using TCP as a transport protocol, and refrain from turning on *IP Security* (IPSec), then NATs generally behave in a relatively stable and unobtrusive manner. Applications that operate conservatively in this limited mode can be unaware of the presence of NATs in their path. The relatively widespread deployment of NATs and the continued use of client-server-based applications on the Internet attests to the capability of the NAT to perform transparently and effectively within the strict confines of this particular mode of communication.

However, peer-to-peer applications are more problematic for NATs, because they have extended the model of a NAT beyond its original realm of capability. If the desire is to continue to support the NAT dynamic binding, but also allow external parties to initiate a communication to a local host, then the NAT ceases to be transparent and unobtrusive, and in this extended environment the NAT transforms itself into an application-visible network element. It is overly presumptuous to claim that NATs have led to the increasing deployment of multiparty applications on the Internet, but certainly multiparty applications have been seen to be useful in circumventing some of the more aggravating shortcomings of NATs in various peer-to-peer realms.

In this latter context, the local party is forced to advertise its willingness to participate in a peer-to-peer realm by communicating with an external agent. The local agent performs a NAT discovery test, and then selects a mode of operation that is consistent with the discovered behaviors of a NAT that may be on the path between the client and the agent. The agent then advertises itself as the local party's intermediary to other peers within the application realm. Attempts to initiate a connection with the local party are directed to the external agent, who then undertakes to perform a rendezvous function in order to establish a session.

Depending on the NATs that may exist between the two parties, the rendezvous function may need to perform a convoluted handshake process, or, in some instances, may not be able to set up a peer-to-peer session at all. This topic of establishing connectivity in the face of NATs in the path is sufficiently complex to warrant a separate examination, and the various techniques and approaches are not examined in this article other than providing some suggestions for further reading.

The salient general observation is that NATs have fueled a new generation of applications that use intermediaries and rendezvous protocols. This shift in application behavior has implied greater attention to security frameworks for applications, because intermediaries represent an additional active element in the trust model. This, in turn, has implied that the application level has to turn to other chains of derivation of trust, because the basic Internet model of some form of persistent identity as being an attribute of an IP address is no longer a workable proposition in the face of NATs. The position we are reaching here is that identity and trust need to be derived from other attributes of the end host and the application that it has invoked.

ICMP

If an *Internet Control Message Protocol* (ICMP) message is passed through NAT, there is not only the outer IP header to consider, but also the ICMP payload. Most ICMP messages contain part of the original IP packet in the body of the message, so for the NAT to behave as transparently as possible, the IP address of the IP header contained in the data part of the ICMP packet should be modified according to the NAT binding state, as well as the IP header Checksum field of this inner packet header.

NATs and IP Fragmentation

NATs that use bindings that include both address and port values do not have a clear and uniform response to fragments of an IP packet. The TCP or UDP header is resident only in the initial IP fragment, and subsequent IP packet fragments do not contain a copy of the transport layer packet header.

Some NATs attempt packet reassembly as if they were the end host, and they perform the NAT translation only when the original IP packet has been reassembled. Of course the reassembled packet may be too large to be forwarded onward, and the NAT may be forced to further fragment the packet. The interplay between this behavior and various forms of path *Maximum Transmission Unit* (MTU) discovery become a source of frustration.

Other NAT packet fragmentation behaviors do not attempt packet reassembly, but rely on a stored packet fragment translation state that directs the translation to be performed on subsequent packet fragments after the initial packet header translation has been performed on the initial IP packet fragment.

This form of behavior has weaknesses in terms of out-of-order fragments, when following fragments are received by the NAT prior to the initial IP packet fragment, and in such cases the NAT often has little choice but to silently discard the out-of-order fragment as untranslatable.

NATs and Application Level Gateways

This brings up one of the more vexing questions regarding NAT behavior, namely, should the NAT include knowledge of the payload of certain applications? Numerous applications, including FTP and the DNS resolution protocol, include IP addresses within the payload of the application. In an effort to achieve complete transparency of operation, some NATs have included *Application Level Gateway* (ALG) functionality for certain applications so that this use of IP addresses in the payload can be detected and altered according to the current NAT translation bindings.

The case of ICMP represents one of the simpler forms of gateway functionality, because it can be performed in the same manner as the basic NAT transform, on a per-packet basis while attempting to maintain retained session state. Payload transformations in the case of a TCP-based application have implications in terms of requiring subsequent alteration of TCP sequence numbers, length fields, and even the repacketization of the payload data stream, given that the data transform required by the address change may imply a change of payload length.

Some units attempt to combine the functionality of a NAT with that of an ALG, such that the NAT is an active intermediary in the transport session. This allows the NAT/ALG to perform "deep" inspection of the packets, and use both application protocol knowledge and per-application-session retained state in order to apply the NAT binding transforms to the application payload as well as to the outer IP packet header.

The most widely deployed application that can use IP addresses in the payload is FTP, where IP addresses are passed in the payload of the control channel in order to allow data sessions to be initiated on distinct transport sessions. The variability and reliability of FTP ALG support in NATs has led to the widespread use of the passive mode of FTP operation, where the data flow is passed within the control session.

A related question is that of the use of IPSec and NATs. IPSec with *Authenticated Header* protection attempts to protect what it believes is the fixed part of the IP packet header, including the source and destination addresses. The NAT changes to the IP packet invalidate the Authentication Header integrity check. Also the NAT changes the IP and UDP or TCP checksums, and this disrupts the *Encapsulating Security Payload* (ESP) function of IPSec. The implication is that IPSec needs to operate upon a TCP or UDP payload, as in the IPSec operating tunnel model, or IPSec carried as a payload within other types of tunnel operation.

It is also the case that NATs today are heavily enmeshed with the UDP and TCP transport protocols. Other transport protocols exist, including the *Streams Control Transport Protocol* (SCTP) and the *Datagram Congestion Control Protocol* (DCCP), and doubtless more transport protocol offerings will follow over time. In each case it is a matter of individual choice how NAT implementations define NAT responses to such additional transport protocols. Although it is tempting to propose that NATs should fall back to an address-only form of binding that was not address-and-port based, this does not appear to be practical guidance. Another aspect of today's NAT deployment is that the most common scenario appears to be that of a single external address and mapping each locally initiated session into a binding that uses this common external IP address and a variable external port number. This means that NATs need to be able to identify and transform port addresses from the Transport Protocol section of the IP header.

Another salient factor here is the common association of NATs and firewalls into a single unit, and the coupling of address utilization compression properties of the NAT with its associated packet-filtering actions. Deploying a NAT at the external interface of a site does lead to more restrictive site filtering outcomes and a more restrictive model of application interaction, where the model attempts to impose the constraint that applications are initiated from within the site, and that unknown or unidentifiable external traffic is considered hostile and should be subject to firewall-based inspection and filtering. From this perspective there is little desire to make more permissive NATs as an isolated exercise, and there is instead a codependence between NAT behaviors and popularly used applications. Applications that work across today's NATs appear to enjoy popular uptake, and applications that enjoy popular uptake appear to determine what forms of traffic pass across NATs.

Popular or not, there are a class of applications that simply cannot work in a "native mode" across NATs, nor can ALGs assist here. These are applications that attempt to impose some level of end-to-end protection on the IP header fields, or use the IP address of the endpoint in a context of some form of persistant identity token. When the NAT alters the IP address, an application that uses strong forms of header validation rejects such packets as corrupted. Within this class of applications and tools, one of the more commonly referenced tools is that of IPSec with Authentication Header. There is a certain sense of irony in the observation that NATs are often seen as part of an overall approach to site security, yet cannot support a "native mode" operation of some of the basic tools that applications could use to support secure end-to-end data transfer.

Views on NATs

It is certainly the case that NATs are very common in todays Internet, and it is worth understanding why NATs have enjoyed such widespread deployment while other technologies appear to be meeting some considerable resistance to widespread deployment. As the original NAT document points out:

"The huge advantage of this approach is that it can be installed incrementally, without changes to either hosts or routers. (A few unusual applications may require changes.) As such, this solution can be implemented and experimented with quickly. If nothing else, this solution can serve to provide temporarily relief while other, more complex and far-reaching solutions are worked out."

—Egevang and Fancis, "Network Address Translator," RFC 1631

More generally, the positive attributes of NATs include the following considerations:

- End hosts and local routers do not change. Whether there is a NAT in place between the local network and the Internet or not, local devices can use the same software and support the same applications. NATs do not require customized versions of operating systems or router images.
- As long as you accept the limitation that sessions must be initiated from the "inside," NATs can work in an entirely transparent fashion for a set of client-server classes of applications.
- If you accept the perspective that services and usage scenarios that are not supported by NATs are "unwelcome" or "unsafe," then NATs can be placed into a role as a component of a site's security architecture, providing protection from attacks launched from the outside toward the inside network.
- NAT conserves its use of public address space.
- NAT allows previously disconnected privately addressed networks to connect to the global Internet without any form of renumbering or host changes—and renumbering networks can be a very time-consuming, disruptive, and expensive operation, or, in other words, renumbering is difficult.
- NAT address space is an effective, provider-independent addressing solution with multihoming capabilities. NAT allows for rapid switching to a different upstream provider, by renumbering the NAT address pool to the new provider's address space. In essence, NATs provide the local network manager with the flexibility of using provider-independent space without having to meet certain size and use requirements that would normally be required for an allocation of public, provider-independent address space.
- NAT allows the network administrator to exercise some control over the form of network transactions that can occur between local hosts and the public network.
- NATs require no local device or application changes. This is perhaps one of the major "features" of NATs, in that the local network requires no changes in configuration to operate behind a NAT.

- NATs do not require a coordinated deployment. There is no transition, and no "flag day" across the Internet. Each local network manager can make an independent decision whether or not to use a NAT. This allows for incremental deployment without mutual dependencies.
- These days the common theme of the public address assignment policy stresses conservative use of address space with minimum waste. The standard benchmark is to be able to show that a target of 80 percent of assigned address space is assigned to a number of connected devices. Achieving such a very high usage rate is a challenging task in many network scenarios, and NATs represent an alternative approach where the local network can be configured using private addresses without reference to the use of public addresses.
- NATs are very widely available and bundled into a large variety of gateway and firewall units. In many units NATs are not an optional extra—they are configured in as a basic item of product functionality.

The market has taken NATs and embraced them wholeheartedly. And in a market-oriented business environment, what is wrong with that?

Unfortunately NATs represent a set of design compromises, and no delving into the world of NATs would be complete without exploring some of their shortcomings. So, after enumerating what are commonly seen as their benefits, it is now necessary to enumerate some of the broken aspects of the world of NATs.

"This solution has the disadvantage of taking away the end-to-end significance of an IP address, and making up for it with increased state in the network."

—Egevang and Francis,
"Network Address Translator," RFC 1631

"An opposing view of NAT is that of a malicious technology, a weed which is destined to choke out continued Internet development. While recognizing there are perceived address shortages, the opponents of NAT view it as operationally inadequate at best, bordering on a sham as an Internet access solution. Reality lies somewhere in between these extreme viewpoints."

—Tony Hain,
"Architectural Implications of NAT," RFC 2993

- First, NATs cannot support applications where the initiator lies on the "outside." The external device has no idea of the address of the local internal device, and, therefore, cannot direct any packets to that device in order to initiate a session. This implies that peer-to-peer services, such as voice, cannot work unaltered in a NAT environment.

- The workaround to this form of shortcoming is to force an altered deployment architecture, where service platforms used by external entities are placed "beside" the NAT, allowing command and control from the interior of the local network, and having a permanent (non-NAT) interface to the external network. Obviously this implies some further centralization of IT services within the NATed site.
- Even this approach does not work well for applications such as voice-over-IP, where the "server" now needs to operate as some form of proxy agent. The generic approach here for applications to traverse NATs in the "wrong" direction is for the inside device to forge a UDP connection to the outside agent, and for the inside device to then establish what NAT translated address has been used, and the nature of the NAT in the path, and then republish this address as the local entity's published service rendezvous point. Sounds fragile? Unfortunately, it is. The other approach is to shift the application to use a set of endpoint identifiers that are distinct from IP addresses, and use a distributed set of "agents" and "helpers" to dynamically translate the application level identifiers into transport IP addresses as required. This tends to create added complexity in application deployment, and also embarks on a path of interdependency that is less than desirable. In summary, workarounds to reestablish a peer-to-peer networking model with NATs tend to be limited, complex, and often fragile.
- The behavior of NATs varies dramatically from one implementation to another. Consequently, it is very difficult for applications to predict or expose the precise behavior of one or more NATs that may exist on the application data path.
- Robust security in IP environments typically operates on an end-to-end model, where both ends include additional information in the packet that can detect attempts to alter the packet in various ways. In IPSec the header part of the packet is protected by the Authentication Header, where an encrypted signature of certain packet header fields is included in the IPSec packet. If the packet header is changed in transit in unexpected ways, the signature check will fail. Obviously IPSec attempts to protect the packet address fields—the very same fields that NATs alter! This leads to the observation that robust security measures and NATs do not mix very well. NATs inhibit implementation of security at the IP level.
- NATs have no inherent failover. NATs are an active in-band mechanism that cannot fail into a safe operating fallback mode. When a NAT goes offline, all traffic through the NAT stops. NATs create a single point where fates are shared in the NAT device maintaining connection state and dynamic mapping information.

- NATs sit on the data path and attempt to process every packet. Obviously bandwidth scaling requires NAT scaling.
- NATs are not backed up by industry-standardized behavior. Although certain NAT-traversal applications make assumptions about the way NATs behave, it is not the case that all NATs necessarily behave in precisely the same way. Applications that work in one context may not necessarily operate in others.
- Multiple NATs can get very confusing with "inside" and "outside" concepts when NATs are configured in arbitrary ways. NATs are best deployed in a strict deployment model of an "inside" being a stub private network and an "outside" of the public Internet. Forms of multiple interconnects, potential loops, and other forms of network transit with intervening NATs lead to very strange failure modes that are at best highly frustrating.
- With NATs there is no clear, coherent, and stable concept of network identity. From the outside these NAT-filtered interior devices are visible only as transient entities.
- Policy-based mechanisms that are based on network identity (for example, *Policy Quality of Service* [QoS]) cannot work through NATs.
- Normal forms of IP mobility are broken when any element behind the NAT attempts to roam beyond its local private domain. Solutions are possible, generally involving specific NAT-related alterations to the behavior of the Home Agent and the mobile device.
- Applications that work with identified devices, or that actually identify devices (such as the *Simple Network Management Protocol* [SNMP] and DNS) require very careful configuration when operating an a NAT environment.
- NATs may drop IP packet fragments in either direction: without complete TCP/UDP headers, the NAT may not have sufficient stored state to undertake the correct header translation.
- NATs often contain ALGs that attempt to be context-sensitive, depending on the source or destination port number. The behavior of the ALGs can be difficult to anticipate, and these behaviors have not always been documented.
- Most NAT implementations with ALGs that attempt to translate TCP application protocols do not perform their functions correctly when the substrings they must translate span across multiple TCP segments; some of them are also known to fail on flows that use TCP option headers, for example timestamps.

From this perspective, NATs are a short-term expediency that is currently turning into a longer-term set of overriding constraints placed on the further evolution of the Internet. Not only do new applications need to include considerations of NAT traversal, but we appear to be entering into a situation where if an application cannot work across NATs, then the application itself fails to gain acceptance. We seem to be locking into a world that is almost the antithesis of the Internet concept. In this NAT-based world, servers reside within the network and are operated as part of the service provider's role, whereas end devices are seen as "dumb" clients, who can establish connections to servers but cannot establish connections between each other. The widespread use of NATs appears to be reinforcing a reemergence of the model of "smart network, dumb clients," whereas others would argue that the network is getting no smarter, it is just that the number of obstacles and amount of network debris is increasing while clients are getting worse at maintaining coherent end-to-end state in the face of such changes.

However, despite their shortcomings, despite the problems NATs create for numerous applications and their users, and despite the continued grappling over a common language to understand how NATs behave, numerous NATs are deployed, and, at least in the IPv4 realm, NATs appear to be a firmly fixed part of the future of the Internet. NATs continue to proliferate in today's Internet.

Moving on with NATs

One commonly held belief is that deployment of IPv6 will eliminate the problem of NATs within the Internet. Certainly it is reasonable to observe that if achieving high address utilization densities is no longer the objective, then there will be plentiful public IPv6 address space and that particular reason to deploy NATs is significantly discounted in an IPv6 realm.

That does not say that IPv6 NATs will not be implemented, nor used. Indeed IPv6 NATs are already available, and they are being used, albeit to some small extent. NATs are, rightly or wrongly, considered to be part of a security solution for a site because of their filtering properties that prevent incoming packets from entering the site unless the NAT already has a permitting binding initiated from the inside. In addition, NATs allow a site to use an internally persistent naming and addressing scheme based on some form of deployment of IPv6 unique *site local* address, and deploy NATs at the edge to create an external view of the site that fits within a provider-based address aggregated view of the IPv6 Internet.

So it would perhaps be too enthusiastic a level of conjecture to suppose that IPv6 will drive away all forms of NAT use in IPv6. It is reasonable to predict that some use of NAT will be seen in IPv6, although many would be highly disappointed if the level of IPv6 NAT use rose to anywhere approaching that of NAT in IPv4.

However, the Internet is still largely a network that uses IPv4 and NATs, and efforts continue along the lines of reducing the amount of friction and frustration in a world in which NATs are prolific. One of the ways to progress here is to treat NAT boxes as yet another instance of Internet middleware, and attempt to apply the same sets of processes to NATs that appear in other instances of middleware. The work of the IETF in the *Middlebox Communication Working Group* uses a model that attempts to expose NATs, as well as firewalls, performance-enhancing proxies, application proxies, and relay agents, to the application, and allows the application to specify the policy that the middlebox should apply. In the case of NATs, this could allow an application to communicate to a NAT that it does not require any form of third-party access, and that a fully symmetric behavior could be applied to the binding without any loss in application functionality. Equally, an application could indicate to the NAT that it expects third parties to be able to use the NAT binding, and that the binding that the NAT will set up for the application should be managed as a port-restricted cone. There is much that could be achieved here that would allow applications to function with some level of determinism, rather than attempting to equip an application with a large and complex toolset of all the relevant techniques of NAT traversal that may be required by the application when confronted by various NAT behaviors.

In the meantime the NAT-behavior guessing game continues. The generic class of techniques that support this function is termed *Unilateral Self-Address Fixing* (UNSAF). This is a process whereby the local entity attempts to determine the address and port by which the entity is known externally, and to determine the characteristics of this association to understand in what contexts the external address may be used as a service rendezvous point for externally initiated communication. Work in this area[10] has exposed many relevant considerations, including a set of deficiencies noted in the previous section.

So, what would a NAT implementation look like if there were standards relating to NAT behaviors and the implementation were to comply with these standards? Numerous efforts have been made to document various forms of network- and application-friendly ways in which NATs could behave, but it would appear that such an effort will require the imprimatur of a standard in order to attain a level of general acceptance from NAT implementations. However, it is possible to predict that any such effort at a "standardized" form of NAT behavior will include the following considerations. The following set of behaviors is based on that enumerated in[13]:

- NATs must show endpoint-independent behavior for UDP-based bindings. This is to ensure that the NAT can support application rendezvous without the need for various multiparty relays and agents.
- NAT should not use port preservation nor port overloading, and should operate in a deterministic manner. Port preservation exposes the NATs to nonstandard behaviors when port preservation cannot be enforced. In addition, NATs must have deterministic behavior.

- A dynamic NAT UDP binding timer should be 5 minutes, and should avoid expiration timers of 2 minutes or less. This is to ensure that the timeout is long enough to avoid excessively frequent timer refresh packets.
- The NAT UDP timeout binding must use a timer refresh based on outbound traffic, and all sessions that use a particular binding should use a common refresh timer. This requirement is a security consideration, in that letting inbound traffic refresh the timer allows an external party to keep a port open on the NAT.
- The NAT filtering function should be address dependant. This represents a balance between security and utility.
- The timeout behavior of the NAT UDP filter must be the same as that of the NAT UDP binding timeout. This is intended to reduce the complexity of applications that are reliant on long-held NAT state.
- The NAT should support hairpin connections, using the external address and port.
- If the NAT includes ALG support, the ALGs should be configurable in terms of being able to turn off the ALG function on a per-application basis.
- NATs should support fragmentation and forwarding of packet fragments.
- NATs must support ICMP *Destination Unreachable* messages, and the ICMP timeout should be greater than 2 seconds.

Learning from NATs

At this stage we can observe a few relevant lessons about NATs:

The first is that we need standards and we rely on standards. For many years the IETF has viewed standardization of NATs and their behavior as being an action that would encourage further deployment of a technology that was apparently considered undesirable. The result has been that NATs have been deployed for reasons entirely unconnected with the IETF and standardization, but because the original specification of NAT behavior was at such a general level each NAT implementor has been forced into making local decisions as to how the NAT should behave under specific circumstances. We now enjoy a network with widespread deployment of an active device that does not have consistent implementations and, in the worst cases, exhibits nondeterministic behaviors. This has made the task of deployment of certain applications on the Internet, including voice-based applications, incredibly difficult.

Whether NATs are good or bad, they would be less of a collective headache today if they shared a common standard core behavior. NATs for IPv6 may be considered to be unnecessary today, and it can be argued they represent no real value to an IPv6 site. But a collection of IPv6 NAT implantations with no common core behavior would constitute a far worse problem to application users. Standardization of technology at least eliminates some of the worst aspects of application level guesswork out of technology deployment.

Secondly, a little bit of security is often far worse than no security. NATs are very poor security devices, and in terms of their behavior with UDP, NATs afford only minor levels of protection. The task of securing a site from various forms of attack and disruption remains one of a careful exercise of assessment of acceptable risk coupled with detailed consideration of site-management functions. NATs are not a quick way out of this effort.

In considering NATs it seems that we are back to the very basics of networking. The basic requirements of any network are "who," "where," and "how," or "identity," "location," and "forwarding." In the case of IP, all these elements were included in the semantics of an IP address, and when addresses get translated dynamically we lose track of IP-level identity across the network. Maybe, just maybe, as we look at the longer-term developments of IP technology, one potential refinement may be the separation of endpoint identity to that of location, and as a potential outcome, NATs could readily manipulate location-based addresses while applications could look to a different token set as a means of establishing exactly who is the other party to the communications.

Of course, if we ever venture down such a path, I trust that such a move toward the use of explicit identities does not generate a complementary deployment of *Network Identity Translators,* or NITs, as an adjunct to the current set of NATs. Too many NITs and NATs will definitely send us all NUTs!

Further Reading

There is no shortage of material on NATs from a wide variety of sources. The following is a list of IETF-related documents, encompassing both published *Request for Comments* (RFCs) and works in progress, that have been circulated as *Internet Drafts.*

RFCs:

[1] Egevang, K., and P. Francis, "The IP Network Address Translator (NAT)," RFC 1631, May 1994.

[2] Srisuresh, P., and D. Gan, "Load Sharing Using IP Network Address Translation (LSNAT)," RFC 2391, August 1998.

[3] Srisuresh, P., and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," RFC 2663, August 1999.

[4] Tsirtsis, G., and P. Srisuresh, "Network Address Translation—Protocol Translation (NAT-PT)," RFC 2776, February 2000.

[5] Hain, T., "Architectural Implications of NAT," RFC 2993, November 2000.

[6] Srisuresh, P., and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," RFC 3022, January 2001.

[7] Holdrege, M., and P. Srisuresh, "Protocol Complications with the IP Network Address Translator," RFC 3027, January 2001.

[8] D. Senie, "Network Address Translator (NAT)-Friendly Application Design Guidelines," RFC 3235, January 2002.

[9] Srisuresh, P., J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan, "Middlebox Communication Architecture and Framework," RFC 3303, August 2002.

[10] Daigle, L., and IAB, "IAB Considerations for Unilateral Self-Address Fixing (UNSAF) Across Network Address Translation," RFC 3424, November 2002.

[11] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN—Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)," RFC 3489, March 2003.

[12] Aboba, B., and W. Dixon, "IPsec—Network Address Translation (NAT) Compatibility Requirements," RFC 3715, March 2004.

Internet Drafts:

Internet Drafts enjoy a fleeting existence, and the following documents may not be available when you read this article. In such cases it is often the case that a decent Internet search will locate the document, or its successor.

[13] Audet, F., and C. Jennings, "NAT/Firewall Behavioral Requirements," work in progress, **`draft-audet-nat-behave`**, July 2004.

[14] Ford, B., P. Srisuresh, and D. Kegel, "Peer-to-Peer(P2P) Communication across Network Address Translators (NATs)," work in progress, **`draft-ford-midcom-p2p`**, June 2004.

[15] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for the Session Initiation Protocol (SIP)," work in progress, **draft-ietf-mmusic-ice,** July 2004.

[16] Jennings, C., "NAT Classification Results Using STUN," work in progress, **draft-jennings-midcom-stun-results**, July 2004.

[17] J. Rosenberg, J. Weinberger, R. Mahy, and C. Huitema, "Traversal Using Relay NAT (TURN)," work in progress, **draft-rosenberg-midcom-turn-01**, July 2004.

Other Resources:

NAT Check: Ford, B. and D. Andersen, Nat Check Website:
http://midcom-p2p.sourceforge.net

STUN Client and Server:
http://sourceforge.net/projects/stun

Phifer, Lisa, "The Trouble with NAT," *The Internet Protocol Journal,* Volume 3, No. 4, December 2000.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector, and has served his time with Telstra, where he was the Chief Scientist in their Internet area. Geoff is currently the Internet Research Scientist at the *Asia Pacific Network Information Centre* (APNIC). He is also the Executive Director of the Internet Architecture Board, and is a member of the Board of the Public Interest Registry. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, and co-author of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@apnic.net**

Letters to the Editor

Content Networks

Dear Editor,

Christophe Deleuze's article on Content Networks (*The Internet Protocol Journal,* Volume 7, Number 2, June 2004) made me realize that there are very different ways to look at this issue. I would like to use the term *Content Addressable Network* for a network that is used to retrieve information not by specifying its location but the identity of the content itself. The term points to similar concepts in electronics (*Content Addressable Memory*) and storage (*Content Addressable Storage*). One could argue that a Content Addressable Network is in fact a distributed Content Addressable Storage.

In a very real sense the Internet already is content addressable. Several of my non-IT friends use the "Search" field in the *Google* toolbar even for regular URLs, foregoing the Address field in their browsers. In doing so, they simply ignore the distinction between *content* and *location*. It usually gets them where they want to go.

Let's define content as a static binary object, for example, a document, picture, song, or movie. How can we identify content if not by location? We can create a hash of the object as a handle or placeholder. (A hash is the result of a calculation that takes the whole object as input. A good hashing algorithm ensures that if you change a bit in the object, at least one bit in its hash changes too.) If we know the placeholder, we can retrieve a copy of the original object, even if we don't know the location of any of the copies out there on the net. I could mail you the hash of a paper, song, or movie and you would be able to retrieve a copy, although not necessarily from the same place as where I got it. (You might have to pay to get it though!)

Suppose that the Google bot, while traversing the Internet to build its index, calculates the hash for each object it encounters. It can then build an index of all hash codes, relating them to the URLs where they were found. (This requires no change in Google: the hash is just one more word it found in the document). We can then google a hash code to find all occurrences of the object. (You can simulate this today by selecting a line of text from a document and launching a search for that sequence of words. Google will often find multiple copies. Just one line of text is an extremely poor hash, so you may get a few false hits, but in my experience not many.)

Simply by adding these hashes, we have turned the Internet into a Content Addressable Network. If our purpose is to make ourselves independent of any single copy on any particular server, this is all we need. For other applications, the objective is to optimize the network paths to the servers that hold a copy of our object (for example, a movie). We need a metric that tells us which of the listed locations is "closest" to our point of entry. This is complicated by the fact that the Internet is a weird space. The shortest route between Amsterdam and Brussels might well go via London or Paris.

Fortunately, there is a database that keeps track of all the available routes and their cost. It is the *Border Gateway Protocol* (BGP) routing table. BGP divides the Internet in chunks called *Autonomous Systems* or ASs and tracks the cost of the routes to each AS. If the Google bot would record the AS along with each URL, our client system could query our local BGP router (or a proxy holding a copy of its database) to find the AS and thus the copy that is closest in terms of network costs. Note that these costs also reflect policy rules such as peering arrangements between ISPs.

If our objective is to dynamically optimize the load on the servers, we cannot avoid querying (a local subset of) these servers for a bid. Distributing the load over servers in different time zones may sometimes be more important than keeping the transports local. Our client should select a server that is not too busy but no further away than necessary.

The Content Networks as discussed by Christophe Deleuze were created as a commercial offering that would require no cooperation from the clients—in every sense an operator's approach. It is restricted to the case where all copies of the object are published by a single entity. The way ahead is to create protocols for requesting network cost for a list of sites, and service costs from a list of servers, independent of the nature of the object and the servers that hold copies of it.

It may seem more efficient to let the publisher add the hash code to the objects. HTML files would be labeled with a **`<MD5=`** tag, obviating the need for bots and users (for "content bookmarks") to do the calculation. This would allow publishers to change content without changing the hash, to correct typos or remove scenes deemed unsuitable for local viewers. But it would no doubt result in fake objects, purporting to be copies of popular objects but peddling dubious commercial proposals. Creating fake objects is more difficult if the hash code is calculated by an independent and unrecognizable bot, although I'm sure the problem is not completely solved with that.

—Ernst Lopes Cardozo, Aranea Consult BV, The Netherlands
`e.lopes.cardozo@aranea.nl`

IPJ Article Identification

Hi,

I noticed that the IPJ page footer only says "The Internet Protocol Journal" but neither the Volume/Issue number, nor the issue date. That makes it a bit hard to correctly reference a given article when you only have a copy of that article and not the whole issue. I propose that you add something like (from the August issue of CACM):

Communications of the ACM August 2004/Vol. 47, No. 8

(I only checked the archived PDF files but I suppose the hardcopy has the same problem.)

—Örjan Petersson
orjan.petersson@logcode.com

We could certainly add the Volume/Issue identifier to the footer, but since this would have to be done retroactively for all 26 issues to date it is probably better to use our soon-to-be-deployed ASCII index. This will allow you to find any article with a simple search. A short sample of the index is shown below.

```
The Internet Protocol Journal Volume 1, 1998

Article                                          Author(s)              Page
-------------------------------------------------------------------------------

* Volume 1, No. 1, June 1998:

What Is a VPN? - Part I                          Ferguson/Huston          2
SSL: Foundation for Web Security                 William Stallings       20
Book Review: Groupware                           Dave Crocker            31
Book Review: High-Speed Networks                 Neophytos Iacovou       33


* Volume 1, No. 2, September 1998:

What Is a VPN? - Part II                         Ferguson/Huston          2
Reliable Multicast Protocols and Applications    C. Kenneth Miller       19
Layer 2 and Layer 3 Switch Evolution             Thayumanavan Sridhar    38
Book Review: Gigabit Ethernet                    Ed Tittel               44


* Volume 1, No. 3, December 1998:

Security Comes to SNMP: SNMPv3                   William Stallings        2
CATV Internet Technology                         Mark Laubach            13
Digital TV                                       George Abe              27
I Remember IANA                                  Vint Cerf               38
Book Review: Internet Messaging                  Dave Crocker            40
Book Review: Web Security                        Richard Perlman         42
Book Review: Internet Cryptography               Frederick M. Avolio     44
```

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Fragments

IPv6 Address "Glue" added to the Root DNS Zone

The *Internet Corporation For Assigned Names and Numbers* (ICANN) recently announced that for the first time, an IPv6 nameserver address has been added to the Internet's root DNS zone. This next generation version of the Internet Protocol provides trillions more addresses than the IPv4 system that is in use by most networks today. By taking this significant step forward in the transition to IPv6, ICANN is supporting the innovations through which the Internet evolves to meet the growing needs of a global economy.

On 20 July 2004 at 18:33 UTC the IPv6 AAAA records for the Japan (**.jp**) and Korea (**.kr**) *country code Top Level Domain* (ccTLD) nameservers became visible in the root zone file with serial number 2004072000. It is expected that the IPv6 records for France (**.fr**) will be added shortly. Other requests are pending and will be added in accordance with documented procedure, which was developed through ICANN's unique multi-stakeholder consensus-based approach. See: **http://www.iana.org/procedures/delegation-data.html**

Recognizing the importance of IPv6 to the Internet community, ICANN has coordinated with its *Root Server System Advisory Committee, Top Level Domain* managers, *Security and Stability Advisory Committee,* and other interested parties in careful analysis of this issue. After a period of thorough examination, the decision was made to move forward with deployment of the IPv6 address records in the manner prescribed by the community.

ICANN is the global public-benefit non-profit organization responsible for coordinating the Internet's naming and numbering systems. For more information please visit: **http://www.icann.org**

Formation of Asia Pacific ENUM Engineering Team

China Network Information Center (CNNIC), *Japan Registry Service* (JPRS), *Korea Network Information Center* (KRNIC), *Singapore Network Information Center* (SGNIC) and *Taiwan Network Information Center* (TWNIC) recently announced the formation of the *Asia Pacific ENUM Engineering Team* (APEET), an informal technical project team formed to coordinate and synergize ENUM activities in the Asia Pacific region.

The proposal to form APEET was discussed during an ENUM BoF (Birds-of-a-Feather) session at the *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) in February 2004. Founding member organizations of APEET shared a common vision that as a collective group, they will be able to achieve greater community awareness and better interoperability of ENUM-based trials.

"ENUM allows IP devices to be assigned a telephone number which is globally interoperable," said James Seng, Chairman of APEET. "It is a key enabling technology for seamless IP Telephony that will greatly benefit the end-users."

Before the formation of APEET, each member organization has been conducting its own ENUM trials, most of which are isolated trials conducted within each member organization's country/region. With the formation of APEET, member organizations will be able to implement technical solutions that facilitate ENUM trials across Asia Pacific.

"We are extremely excited about the formation of this much needed organization," said Hiro Hotta, Director JPRS. "We are ready to bring ENUM trials to the next level."

One of APEET's key project is to implement a live ENUM trial at APRICOT 2005, Kyoto, Japan. The live trial will allow hundreds of APRICOT participants to experience IP Telephony using wireless SIP Phones and calling each another with standard 10-key telephone interface via ENUM. The live trial, believed to be the first of its kind, will serve to demonstrate and educate the technical community on the power, capabilities and feasibility of ENUM together with SIP.

"This looks like one of the most exciting events of 2005 with a demonstration of technologies to rock Asia Pacific," said Richard Shockey, co-Chair of the ENUM Working Group of the IETF.

The formation of APEET has been well received by the Industry. The *Asia Pacific Network Information Centre* (APNIC) has extended its goodwill to host DNS records of **apenum.org**, the selected "golden root" of APEET technical trials. APEET is also fortunate to have individual experts member such as Richard Shockey.

APEET welcomes all Asia Pacific ccTLD administrators (or its designated representatives) to join and contribute towards the success of ENUM adoption in Asia Pacific. For more information, please visit **http://www.apenum.org**

Phill Gross Receives Postel Award

Phill Gross is this year's recipient of the prestigious *Jonathan B. Postel Service Award*. A co-founder of the *Internet Engineering Task Force* (IETF), Gross has been instrumental in defining and shaping the way in which the IETF standards process functions. He was awarded the Postel Service Award in recognition of his early leadership of the IETF and for firmly establishing the principles that are essential for its success. The Postel Award was presented on August 5th, during the 60th meeting of the IETF in San Diego, California.

"The Internet Society is pleased to recognize Phill's significant contribution to the area of Internet standardization by awarding him this year's Postel Award," said Internet Society President and CEO Lynn St.Amour. "The continued success of the IETF's consensus-based processes shows the importance of Phill's pioneering work in developing the IETF's foundations."

According to Steve Crocker, noted Internet authority and chair of this year's Postel award committee, "Many of the IETF's current structures, including Working Groups, Technical Areas, Proceedings and Internet Drafts came about thanks to Phill's dedication and passion for the Internet standards area. And we're delighted to be presenting the award to Phill in San Diego, the location of the first ever IETF meeting back in 1986."

Gross, who is currently Director of Academics and Technology for the Northern Virginia ECPI College of Technology, has worked with the Internet community for over 20 years. His career has taken him from working with government-funded research projects through to networking engineering responsibilities for large corporations and startups, including leading the development of MCI Corporation's first national network.

In 1986 Gross helped found the IETF. He became the first official chair in 1987—a position he held for seven years. During his chairmanship, the IETF evolved from a government-sponsored research group to an industry-wide Internet standards body. As well as contributing to developing the IETF standards process itself, Gross played an active role as co-chair of the IETF Routing and Addressing Working Group. This group led to solutions for growth-related Internet problems and was instrumental in specifying the initial direction for the next generation *Internet Protocol* (IPv6) in RFC 1719. He also served as a member of the *Internet Architecture Board* (IAB) from 1987 to 1996.

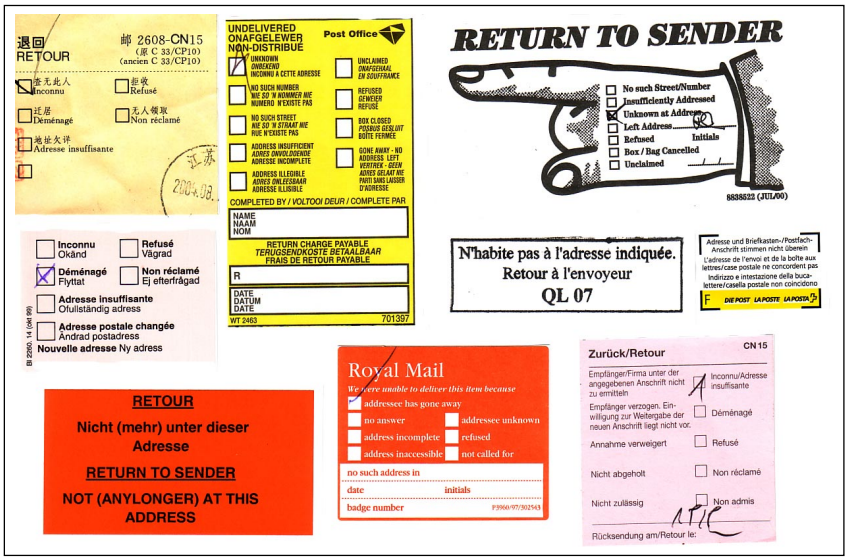
Expressing his appreciation for the award, Gross said "It was very gratifying to be there at the beginning and to work with such an incredible group of people. And, working with Jon over the years gives me a special appreciation for the honor that comes with this award."

The Jonathan B. Postel Service Award was established by the Internet Society to honor those who have made outstanding contributions in service to the data communications community. The award is focused on sustained and substantial technical contributions, service to the community, and leadership. With respect to leadership, the nominating committee places particular emphasis on candidates who have supported and enabled others in addition to their own specific actions. The award is named after Dr. Jonathan B. Postel, who embodied all of these qualities during his extraordinary stewardship over the course of a thirty-year career in networking. He served as the editor of the RFC series of notes from its inception in 1969, until 1998. He also served as the ARPANET "Numbers Czar" and the *Internet Assigned Numbers Authority* (IANA) over the same period of time. He was a founding member of the Internet Architecture Board and the first individual member of the Internet Society, where he also served as a trustee. Previous recipients of the Postel Award include Jon himself (posthumously and accepted by his mother), Scott Bradner, Daniel Karrenberg, Stephen Wolff and Peter Kirstein. For more information, please visit:
http://www.isoc.org

Where did my copy of IPJ go?

Each time we mail out a new issue of IPJ, a certain number of copies are returned to us as undeliverable by the postal authorities around the globe. These so-called "Nixies" can take as much as a year to arrive back in San Jose, California, and almost all of them are returned without any updated delivery information. Obviously we cannot do much other than delete these records from our database. However, if you tell us when you move, we can make sure your address is up-to-date so that you will receive the next issue of IPJ. You can update your own record using the subscription tool at **`http://www.cisco.com/ipj`** or just send your updates via e-mail to: **`ipj@cisco.com`**

Where did you go?
Do let us know!



The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.

The Internet Protocol Journal

December 2004 Volume 7, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

The electronics industry is full of examples of devices which contain one or two "special-purpose" chips. Your computer probably has a modem that is implemented with a single chip and a few analog components. It probably also contains a dedicated graphics processor responsible for driving your display. In networking, vendors have long since realized that in order to design highly efficient routers or switches, a custom-designed *network processor* is a good solution. We asked Doug Comer to give us an overview of network processors.

Attacks against individual computers on a network have become all too common. Usually these attacks take the form of a virus or worm which arrives via e-mail to the victim's machine. The industry has been relatively quick in responding to such attacks by means of antivirus software, as well as sophisticated filtering of content "on the way in." A more serious form of attack is the *Distributed Denial-of-Service* (DDoS) attack which may render an entire network unusable. Charalampos Patrikakis, Michalis Masikos, and Olga Zouraraki give an overview of the many variants of denial-of-service attacks and what can be done to prevent them.

Although we make every effort to provide you with an error-free journal, mistakes do happen occasionally. Sometimes it takes careful analysis by a reader to spot the mistake, and we are grateful for the correction provided in the "Letter to the Editor" on page 36. Other times, technology just gets in our way, such as when all the non-printing end-of-line and TAB characters became very much "printing"—see page 35 of the printed version of Volume 7, No. 3. At least it didn't show up in the PDF or HTML versions.

Take a moment to visit our Website: **http://www.cisco.com/ipj** and update your mailing address if necessary. You will also find all back issues and index files at the same address.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Network Processors: Programmable Technology for Building Network Systems

by Douglas Comer, Cisco Systems (on leave from Purdue University)

Chip vendors have defined a new technology that can be used to implement packet-processing systems such as routers, switches, and firewalls. The technology offers the advantages of being software-programmable and sufficiently high-speed to accommodate interfaces running at 10 Gbps.

This article provides an overview of the technology, describes the motivations, and presents a brief survey of hardware architectures. It also discusses the relationship between programming and the underlying hardware.

A wide variety of packet-processing systems are used in the Internet, including DSL modems, Ethernet switches, IP routers, *Network Address Translation* (NAT) boxes, *Intrusion Detection Systems* (IDS), Softswitches used for *Voice over IP* (VoIP), and security firewalls. Such systems are engineered to provide maximal functionality and performance (for example, operate at wire speed) while meeting constraints on size, cost, and time to market.

Engineers who design network systems face the additional challenges of keeping designs scalable, general, and flexible. In particular, because industry trends change rapidly, typical engineering efforts must accommodate changes in requirements during product construction and changes in the specification for a next-generation product.

Generations of Network Systems

During the past 20 years, engineering of network systems has changed dramatically. Architectures can be divided broadly into three generations:

- *First generation* (circa 1980s): Software running on a standard processor (for example, an IP router built by adding software to a standard minicomputer),
- *Second generation* (mid 1990s): Classification and a few other functions offloaded from the CPU with special-purpose hardware, and a higher-speed switching fabric replacing a shared bus.
- *Third generation* (late 1990s): Completely decentralized design with *Application-Specific Integrated Circuit* (ASIC) hardware plus a dedicated processor on each network interface offloading the CPU and handling the fast data path.

The change from a centralized to a completely distributed architecture has been fundamental because it introduces additional complexity. For example, in a third-generation IP router, where each network interface has a copy of the routing table, changing routes is difficult because all copies must be coordinated to ensure correctness and the router should not stop processing packets while changes are propagated.

Motivation for Network Processors

Although the demand for speed pushed engineers to use ASIC hardware in third-generation designs, the results were disappointing. First, building an ASIC costs approximately US$1 million. Second, it takes 18 to 22 months to generate a working ASIC chip. Third, although engineers can use software simulators to test ASIC designs before chips are manufactured, networking tasks are so complex that simulators cannot handle the thousands of packet sequences needed to verify the functionality. Fourth, and most important, ASICs are inflexible.

The inflexibility of ASICs impacts network systems design in two ways. First, changes during construction can cause substantial delay because a small change in requirements can require massive changes in the chip layout. Second, adapting an ASIC for use in another product or the next version of the current project can introduce high cost and long delays. Typically, a silicon respin takes an additional 18 to 20 months.

Network-Processor Technology

In the late 1990s as demand for rapid changes in network systems increased, chip manufacturers began to explore a new approach: programmable processors designed specifically for packet-processing tasks. The goal was clear: combine the advantage of software programmability, the hallmark of the first-generation network systems, with high speed, the hallmark of third-generation network systems.

Chip vendors named the new technology *network processors,* and predicted that in the future, most network systems would be constructed using network processors. Of course, before the prediction could come true, vendors faced a tough challenge: programming introduces an extra level of indirection, meaning that functionality implemented directly in hardware always performs faster than the same functionality implemented with software. Thus, to make a network processor fast enough, packet-processing tasks need to be identified and special-purpose hardware units constructed to handle the most intensive tasks.

Interestingly, vendors also face an economic challenge: although an ASIC costs a million dollars to produce, subsequent copies of the chip can be manufactured at very low cost. Thus, the initial development cost can be amortized over many copies. In contrast, purchasing conventional processors does not entail any initial development cost, but vendors typically charge at least an order of magnitude more per unit than for copies of an ASIC. So, vendors must consider a pricing strategy that entices systems builders to use network processors in systems that have many network interfaces with multiple processors per interface.

A Plethora of Architectures

As vendors began to create network processors, fundamental questions arose. What are the most important protocol-processing tasks to optimize? What hardware units should a network processor provide to increase performance? What I/O interfaces are needed? What sizes of instruction store and data store are needed? What memory technologies should be used (for example, *Static Random-Access Memory* [SRAM], *Dynamic Random-Access Memory* [DRAM], or others)? How should functional units on the network-processor chip be organized and interconnected (for example, what on-chip bus infrastructure should be used)?

Interestingly, although they realized that it was essential to identify the basic protocol-processing tasks before hardware could be built to handle those tasks efficiently, chip vendors had little help from the research community. Much effort had been expended considering how to implement specific protocols such as IP or TCP on conventional processors. However, researchers had not considered building blocks that worked across all types of network systems and all layers of the protocol stack. Consequently, in addition to designing network-processor chips, vendors needed to decide which protocol functions to embed in hardware, which to make programmable, and which (if any) to leave for special-purpose interface chips or coprocessors. Finally, chip vendors needed to choose software support including programming language(s), compilers, assemblers, linkers, loaders, libraries, and reference implementations.

Faced with a myriad of questions and possibilities about how to design network processors and the recognition that potential revenue was high if a design became successful, chip vendors reacted in the expected way: each vendor generated a design and presented it to the engineering community. By January 2003, more than 30 chip vendors sold products under the label "network processor."

Unfortunately, the euphoria did not last, and many designs did not receive wide acceptance. Thus, companies began to withdraw from the network-processor market, and by January 2004, fewer than 30 companies sold network processors.

Basic Architectural Approaches

Hardware engineers use three basic techniques to achieve high-speed processing: a single processor with a fast clock rate, parallel processors, and hardware pipelining. Figure 1 illustrates packet flow through a single processor, which is known as an *embedded processor architecture* or a *run-to-completion model*. In the figure, three functions must be performed on each packet.

Figure 1: Embedded Processor Architecture in Which a Single Processor Handles All Packets

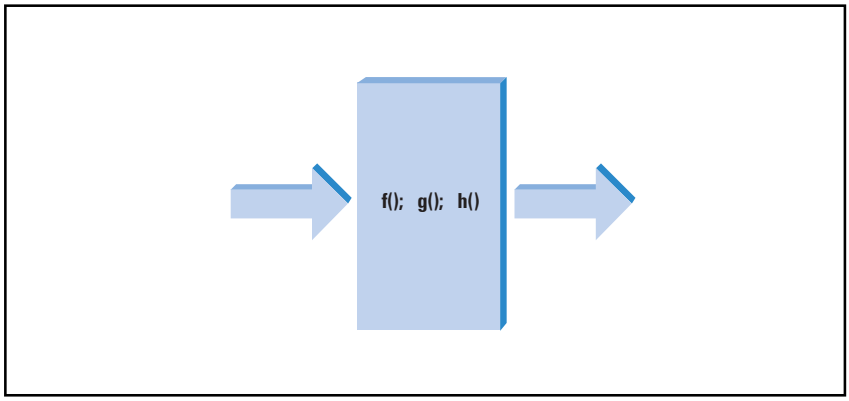


Figure 2 illustrates packet flow through an architecture that uses a parallel approach. A coordination mechanism on the ingress side chooses which packets are sent to which processor. Coordination hardware can use a simplistic round-robin approach in which a processor receives every *N*th packet, or a sophisticated approach in which a processor receives a packet whenever the processor becomes idle.

Figure 2: Parallel Architecture in Which the Incoming Packet Flow Is Divided Among Multiple Processors

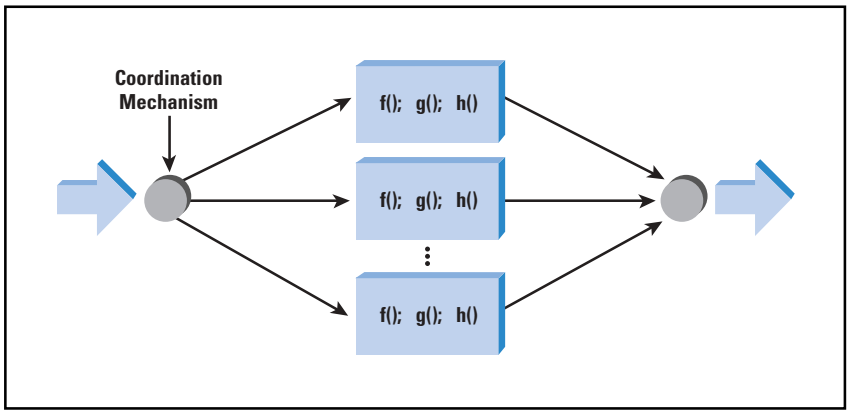
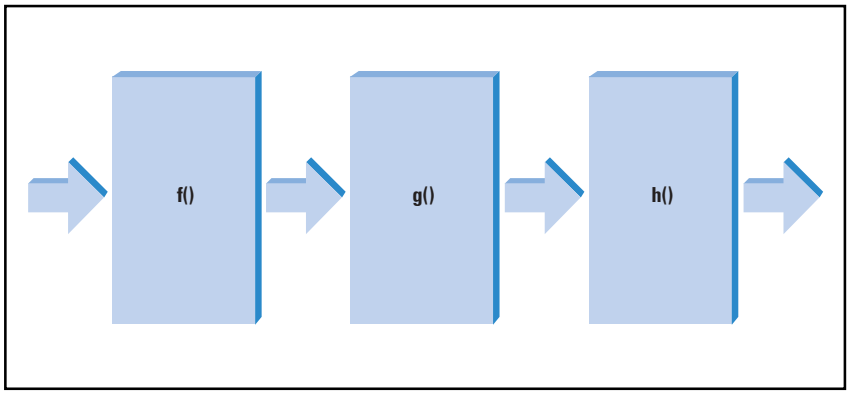


Figure 3 illustrates packet flow through a pipeline architecture. Each packet flows through the entire pipeline, and a given stage of the pipeline performs part of the required processing.

Figure 3: Pipeline Architecture in Which Each Incoming Packet Flows Through Multiple Stages of a Pipeline



As we will see, pipelining and parallelism can be combined to produce hybrid designs. For example, it is possible to have a pipeline in which each individual stage is implemented by parallel processors or a parallel architecture in which each parallel unit is implemented with a pipeline.

Examples of Commercial Architectures

To appreciate the broad range of network-processor architectures, we will examine a few commercial examples. Commercial network processors first emerged in the late 1990s, and were used in products as early as 2000. The examples contained in this article are chosen to illustrate concepts and show broad categories, not to endorse particular vendors or products. Thus, the examples are not necessarily the best, nor the most current.

Augmented RISC (Alchemy)

The first example, from Alchemy Semiconductor (now owned by Advanced Micro Devices), illustrates an embedded processor augmented with special instructions and I/O interfaces.

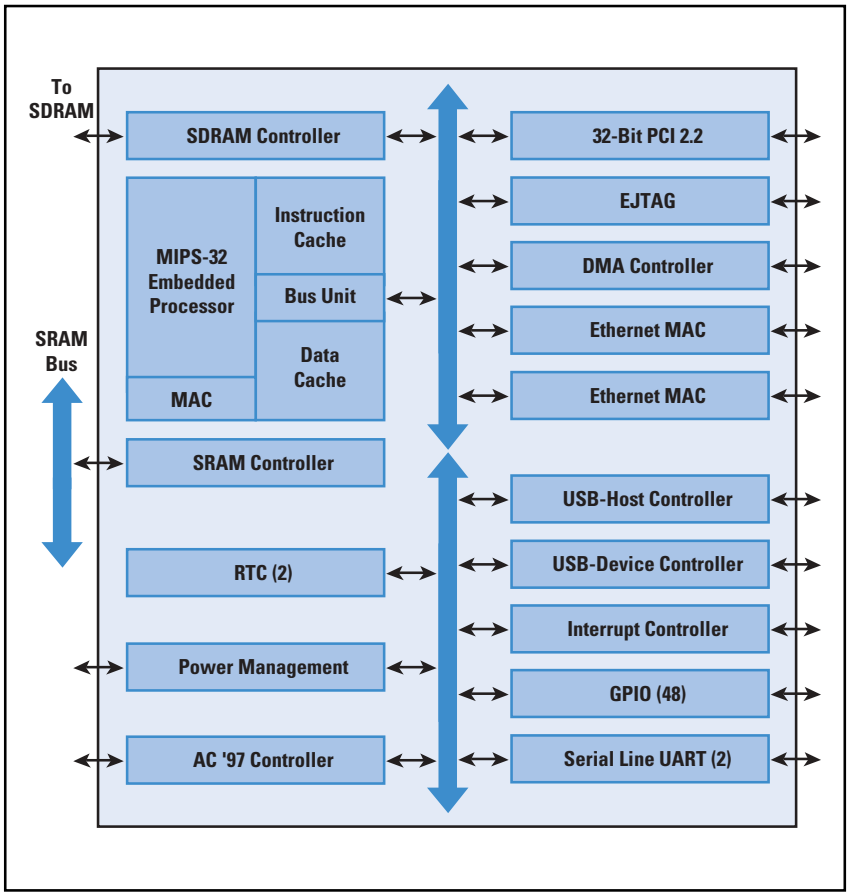


Figure 4: An Example Embedded Processor Architecture: The Processor Has Extra Instructions to Speed Packet Processing

Parallel Processors Plus Coprocessors (AMCC)

A network processor from AMCC uses an architecture with parallel processors plus coprocessors that handle packet-processing tasks. When a packet arrives, one of the parallel processors, called *cores*, handles the packet. The coprocessors are shared—any of the parallel processors can invoke a coprocessor, when needed.

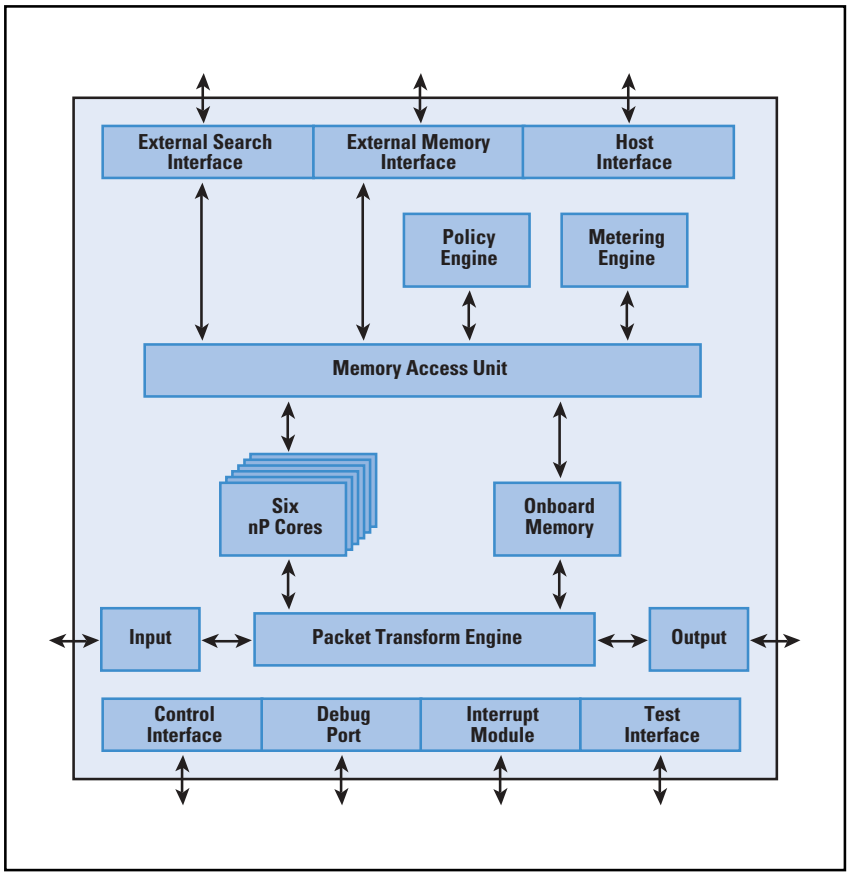
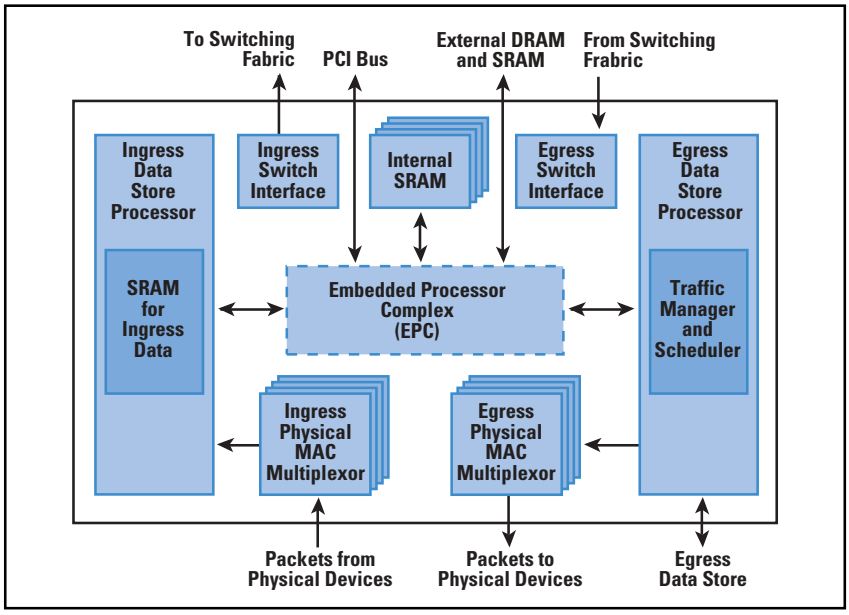


Figure 5: An Example Parallel Architecture that Uses Special-Purpose Coprocessors to Speed Execution

Extensive and Diverse Processors (Hifn)

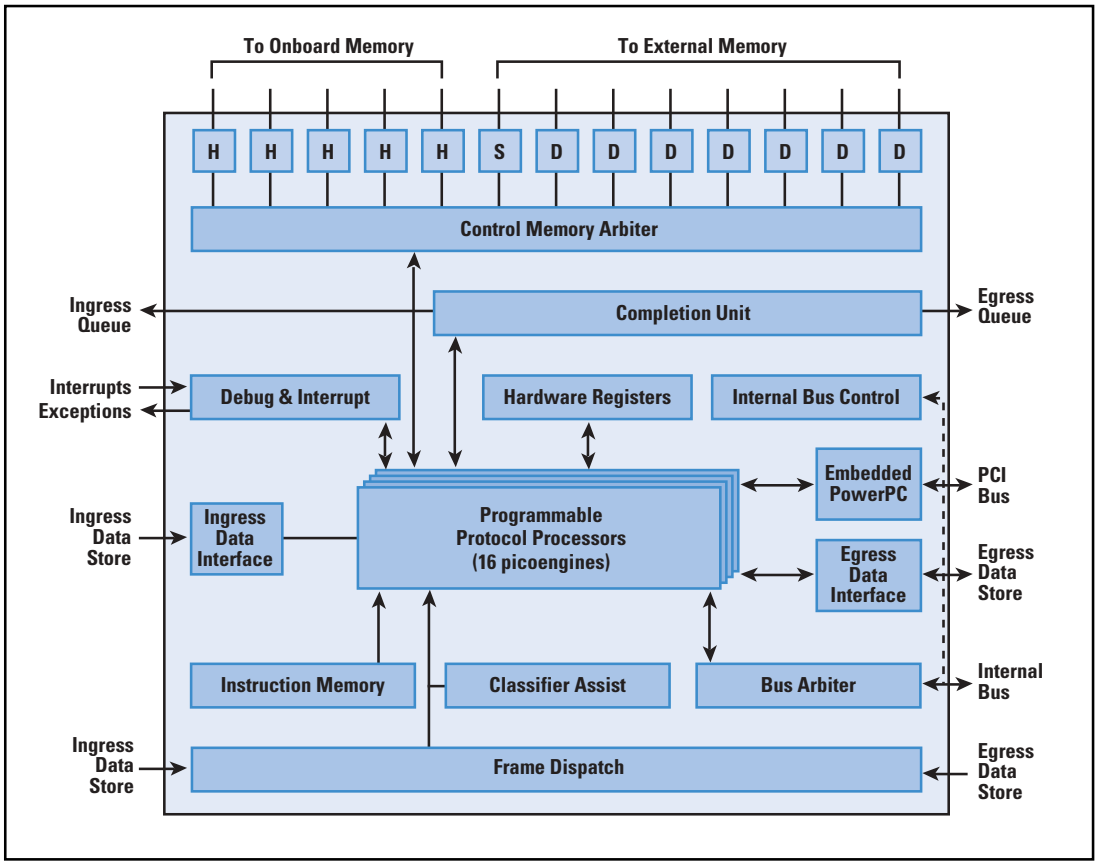
A network processor (named *Rainier*) originally developed by IBM and now owned by Hifn Corporation uses a parallel architecture, and includes a variety of special-purpose and general-purpose processors. For example, the chip provides parallel ingress and egress hardware to handle multiple high-speed network interfaces. It also has intelligent queue-management hardware that enqueues incoming packets in an ingress data store, a switching fabric interface built onto the chip, and an intelligent egress data store. Figure 6 illustrates the overall architecture of the Hifn chip.

Figure 6: An Example Parallel Architecture that Includes Hardware Support for Ingress and Egress Processing as well as Intelligent Queuing



The *Embedded Processor Complex* (EPC) on the Hifn chip contains 16 programmable packet processors, called *picoengines*, as well as various other coprocessors. In addition, the EPC contains an embedded PowerPC to handle control and management tasks. Figure 7 shows a few of the many processors in the EPC.

Figure 7: Structure of the Embedded Processor Complex on the Example Network Processor in Figure 6



Parallel Pipelines of Homogeneous Processors (Cisco)

Although it is not a chip vendor, Cisco Systems uses network processors in its products, and has developed network processors for internal use. One of the more interesting designs employs parallel pipelines of homogeneous processors. Figure 8 illustrates the architecture of the Cisco chip. When a packet enters, the hardware selects one of the pipelines, and the packet travels through the entire pipeline.

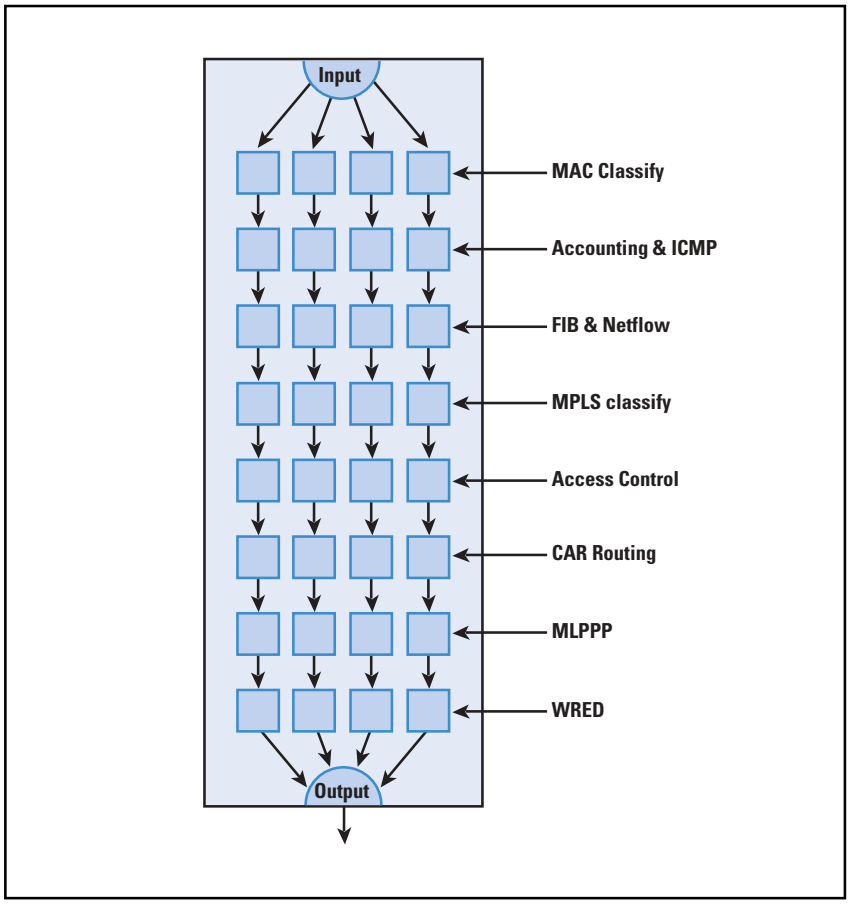
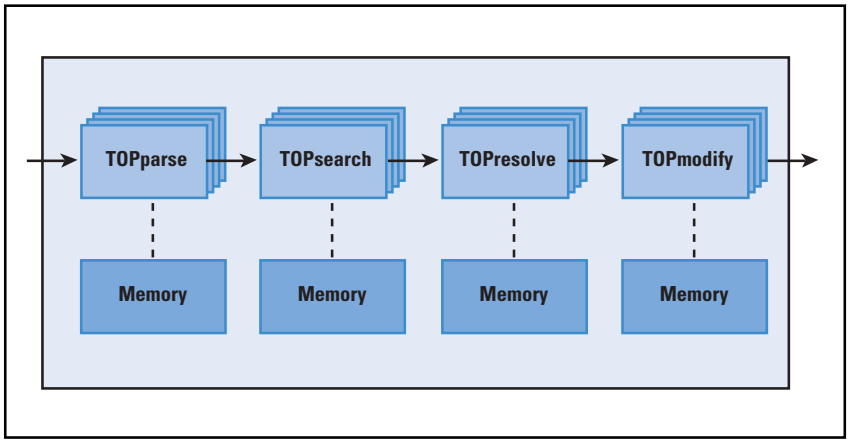


Figure 8: An Example Architecture that Uses Parallel Pipelines of Homogeneous Processors

Pipeline of Parallel Heterogeneous Processors (EZchip)

EZchip Corporation sells a network processor that combines pipelining and parallelism by using a four-stage pipeline in which each stage is implemented by parallel processors. However, instead of using the same processor type at each stage, the EZchip architecture employs heterogeneous processors, with the processor type at each stage optimized for a certain task (for example, the processor that runs forwarding code is optimized for table lookup). Figure 9 illustrates the architecture.

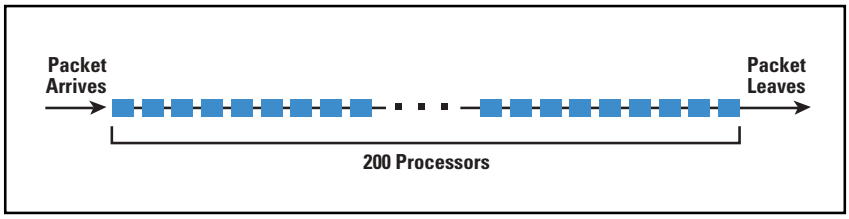
Figure 9: An Example Architecture that Uses a Pipeline of Parallel Stages with Heterogeneous Processors



Extremely Long Pipeline (Xelerated)

Xelerated Corporation sells an interesting network processor that uses a pipelining approach. Unlike other network processors, the Xelerated chip uses an extremely long pipeline of 200 stages. Figure 10 illustrates the overall architecture. To achieve high speed, each stage is limited to executing four instructions per packet.

Figure 10: An Example of an Extremely Long Pipeline with 200 Stages



In fact, the Xelerated architecture is more complex than the figure shows because the pipeline contains special hardware units after every 10 stages that allow external communication (for example, access to external memory or a call to a coprocessor).

More Details and Example Network-Processor Source Code

The previous survey is not meant to be complete. Two notable network processors have been omitted. Agere Systems and Intel each manufacture a network processor. Agere's design consists of a short pipeline that has two basic stages. Agere's architecture is both interesting and unusual because the two stages are composed of unconventional processors. For example, the processor used for classification performs high-speed pattern matching, but does not have conventional instructions for iteration or conditional testing. For details about the Agere network processor see[1], which includes the source code for an example *Differentiated Services* (DiffServ) network system.

Intel's chip uses a parallel approach in which a set of *microengines* are programmed to handle packets. The Intel hardware allows a programmer to pass packets between microengines, meaning a programmer can decide to arrange microengines in a software pipeline. For details about the Intel network processor see[2], which includes the source code for an example NAT implementation.

Programming Network Processors

Although the general idea of building programmable devices seems appealing, most network-processor designs make programming difficult. In particular, to achieve high speed, many designs use low-level hardware constructs and require a programmer to accommodate the hardware by writing low-level code. Many network processors are much closer to a microcontroller than a conventional processor, and are programmed in *microassembly* language. Programmers must be conscious of details such as register banks.

Programming is especially difficult in cases where the network-processor hardware uses explicit parallelism and requires a programmer to plan program execution in such a way that processors do not contend for resources simultaneously or otherwise stall. For example, on one vendor's chip, a packet processor can execute several hundred instructions while waiting for a single memory access to complete. Thus, to achieve high performance, a programmer must start a memory operation, go on with other calculations while the memory operation proceeds, and then check that the operation has completed.

In addition to considering processing, some network processors provide a set of memory technologies, and require a programmer to allocate each data item to a specific memory. A programmer must understand memory latency, the expected lifetime of a data object, and the expected frequency of access as well as properties of the hardware such as memory banks and interleaving.

A few pleasant exceptions exist. For example, Agere Systems provides special-purpose, high-level programming languages to program its network processors. Thus, it is easy to write classification code or traffic-management scripts for an Agere processor. More important, an Agere chip offers implicit parallelism: a programmer writes code as if a single processor is executing the program; the hardware automatically runs multiple copies on parallel hardware units and handles all details of coordination and synchronization.

Another pleasant exception comes from IP Fabrics, which has focused on building tools to simplify programming. Like Agere, IP Fabrics has developed a high-level language that allows a programmer to specify packet classification and the subsequent actions to be taken. The language from IP Fabrics is even more compact than the language from Agere.

Summary

To provide maximal flexibility, ease of change, and rapid development for network systems, chip vendors have defined a new technology known as network processors. The goal is to create chips for packet processing that combine the flexibility of programmable processors with the high speed of ASICs.

Because there is no consensus on which packet-processing functions are needed or which hardware architecture(s) are best, vendors have created many architectural experiments. The basic approaches comprise an embedded processor, parallelism, and hardware pipelining. Commercial chips often combine more than one approach (for example, a pipeline of parallel stages or parallel pipelines).

Programming network processors can be difficult because many network processors provide low-level hardware that requires a programmer to use a microassembly language and handle processor, memory, and parallelism details. A few exceptions exist where a vendor provides a high-level language.

References

[1] Comer, D., *Network Systems Design Using Network Processors, Agere Version*, Prentice Hall, 2005.

[2] Comer, D., *Network Systems Design Using Network Processors, Intel 2xxx Version,* Prentice Hall, 2005.

This article is based on material in *Network Systems Design Using Network Processors, Agere Version,* and *Network Systems Design Using Network Processors, Intel 2xxx Version* by Doug Comer. Both books are published by Prentice Hall in 2005. Used with permission.

DOUGLAS E. COMER is a Visiting Faculty at Cisco Systems, a Distinguished Professor of Computer Science at Purdue University, a Fellow of the ACM, and editor-in-chief of the journal *Software—Practice and Experience.* As a member of the IAB, he participated in the formation of the Internet, and is considered a leading authority on TCP/IP and Internetworking. He is the author of 16 technical books that that have been translated into 14 languages, and are used around the world in industry and academia. Comer has been working with network processors for several years, and has reference platforms from three leading vendors in his lab at Purdue. E-mail: **comer@cs.purdue.edu**

Distributed Denial of Service Attacks

By Charalampos Patrikakis, Michalis Masikos, and Olga Zouraraki
National Technical University of Athens

The Internet consists of hundreds of millions of computers distributed all around the world. Millions of people use the Internet daily, taking full advantage of the available services at both personal and professional levels. The interconnectivity among computers on which the World Wide Web relies, however, renders its nodes an easy target for malicious users who attempt to exhaust their resources and launch *Denial-of-Service* (DoS) attacks against them.

A DoS attack is a malicious attempt by a single person or a group of people to cause the victim, site, or node to deny service to its customers. When this attempt derives from a single host of the network, it constitutes a DoS attack. On the other hand, it is also possible that a lot of malicious hosts coordinate to flood the victim with an abundance of attack packets, so that the attack takes place simultaneously from multiple points. This type of attack is called a *Distributed DoS,* or DDoS attack.

DDoS Attack Description

DoS attacks attempt to exhaust the victim's resources. These resources can be network bandwidth, computing power, or operating system data structures. To launch a DDoS attack, malicious users first build a network of computers that they will use to produce the volume of traffic needed to deny services to computer users. To create this attack network, attackers discover vulnerable sites or hosts on the network. Vulnerable hosts are usually those that are either running no antivirus software or out-of-date antivirus software, or those that have not been properly patched. Vulnerable hosts are then exploited by attackers who use their vulnerability to gain access to these hosts. The next step for the intruder is to install new programs (known as *attack tools*) on the compromised hosts of the attack network. The hosts that are running these attack tools are known as *zombies,* and they can carry out any attack under the control of the attacker. Many zombies together form what we call an *army.*

But how can attackers discover the hosts that will make up the attack network, and how can they install the attack tools on these hosts? Though this preparation stage of the attack is very crucial, discovering vulnerable hosts and installing attack tools on them has become a very easy process. There is no need for the intruder to spend time in creating the attack tools because there are already prepared programs that automatically find vulnerable systems, break into these systems, and then install the necessary programs for the attack. After that, the systems that have been infected by the malicious code look for other vulnerable computers and install on them the same malicious code. Because of that widespread scanning to identify victim systems, it is possible that large attack networks can be built very quickly.

The result of this automated process is the creation of a DDoS attack network that consists of handler (master) and agent (slave, daemon) machines. It can be inferred from this process that another DDos attack takes place while the attack network is being built, because the process itself creates a significant amount of traffic.

Recruiting the Vulnerable Machines

Attackers can use different kinds of techniques (referred to as *scanning techniques*) in order to find vulnerable machines[1][2][3]. The most important follow:

- *Random scanning:* In this technique, the machine that is infected by the malicious code (such a machine can be either the attacker's machine or the machine of a member of their army, such as a zombie) probes IP addresses randomly from the IP address space and checks their vulnerability. When it finds a vulnerable machine, it breaks into it and tries to infect it, installing on it the same malicious code that is installed on itself. This technique creates significant traffic, because the random scanning causes a large number of compromised hosts to probe and check the same addresses. An advantage (to attackers) of this scanning method is that the malicious code can be spread very quickly because the scans seem to come from everywhere. However, the fast rate at which the malicious code is dispersed cannot last forever. After a small period of time, the spreading rate reduces because the number of the new IP addresses that can be discovered is smaller as time passes. This becomes obvious if we consider the analysis of David Moore and Colleen Shannon[4] on the spread of the Code-Red (CRv2) Worm, which uses random scanning to spread itself.
- *Hit-list scanning:* Long before attackers start scanning, they collect a list of a large number of potentially vulnerable machines. In their effort to create their army, they begin scanning down the list in order to find vulnerable machines. When they find one, they install on it the malicious code and divide the list in half. Then they give one half to the newly compromised machine, keep the other half, and continue scanning the remaining list. The newly infected host begins scanning down its list, trying to find a vulnerable machine. When it finds one, it implements the same procedure as described previously, and in this way the hit-list scanning takes place simultaneously from an enduringly increasing number of compromised machines. This mechanism ensures that the malicious code is installed on all vulnerable machines contained in the hit list in a short period of time. In addition, the hit list possessed by a new compromised host is constantly reducing because of the partitioning of the list discussed previously.

  As has been mentioned, the construction of the list is carried out long before the attackers start scanning. For that reason, the attackers can create the list at a very slow rate and for a long period of time. If the attackers conduct a slow scan, it is possible that this activity would not be noticed because a scanning process in a network usually occurs at extremely high frequencies, so a slow scan could occur without anyone realizing that it is a malicious scan.

It should also be mentioned that there are public servers such as the Netcraft Survey[2] that can create such hit lists without scanning.

- *Topological scanning:* Topological scanning uses information contained on the victim machine in order to find new targets. In this technique, an already-compromised host looks for URLs in the disk of a machine that it wants to infect. Then it renders these URLs targets and checks their vulnerability. The fact that these URLs are valid Web servers means that the compromised host scans possible targets directly from the beginning of the scanning phase. For that reason, the accuracy of this technique is extremely good, and its performance seems to be similar to that of hit-list scanning. Hence, topological scanning can create a large army of attackers extremely quickly and in that way can accelerate the propagation of the malicious code.
- *Local subnet scanning:* This type of scanning acts behind a firewall in an area that is considered to be infected by the malicious scanning program. The compromised host looks for targets in its own local network, using the information that is hidden in "local" addresses. More specifically, a single copy of the scanning program is running behind a firewall and tries to break into all vulnerable machines that would otherwise be protected by the firewall. This mechanism can be used in conjunction with other scanning mechanisms: for example, a compromised host can start its scans with local subnet scanning, looking for vulnerable machines in its local network. As soon as it has probed all local machines, it can continue the probing process by switching to another scanning mechanism in order to scan off-local network machines. In that way, an army with numerous zombies can be constructed at an extremely high speed.
- *Permutation scanning:* In this type of scanning, all machines share a common pseudorandom permutation list of IP addresses. Such a permutation list can be constructed using any block cipher of 32 bits with a preselected key[3]. If a compromised host has been infected during either the hit-list scanning or local subnet scanning, it starts scanning just after its point in the permutation list and scans through this list in order to find new targets. Otherwise, if it has been infected during permutation scanning, it starts scanning at a random point. Whenever it encounters an already-infected machine, it chooses a new random start point in the permutation list and proceeds from there. A compromised host can recognize an already-infected machine among noninfected ones, because such machines respond differently than other machines. The process of scanning stops when the compromised host encounters sequentially a predefined number of already-infected machines without finding new targets during that period of time. Then, a new permutation key is produced and a new scanning phase begins. This mechanism serves two major purposes: first, it prevents unnecessary reinfections of the same target because when a compromised host recognizes an already-compromised machine, it changes the way it scans according to the process described previously.

Second, this mechanism maintains the advantages (to attackers) of random scanning, because the scanning of new targets takes place in a random way. Hence, permutation scanning can be characterized as a coordinated scanning with an extremely good performance, because the randomization mechanism allows high scanning speeds.

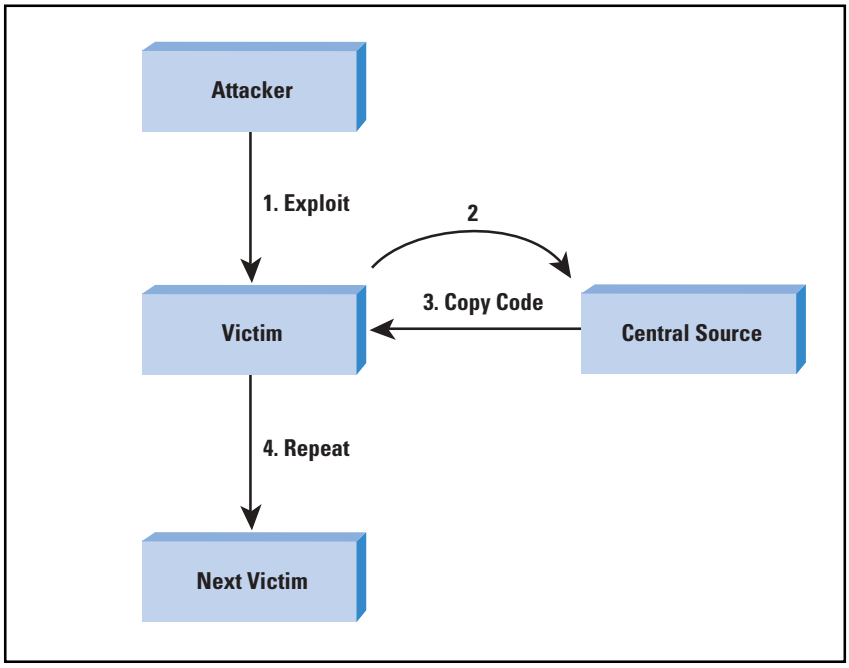
An improved version of permutation scanning is *partitioned permutation scanning*. This type of scanning is a combination of permutation and hit-list scanning. In this scenario, the compromised machine has a permutation list, which is cut in half when it finds a new target. Then it keeps one section of the list and gives the other section to the newly compromised machine. When the permutation list that an infected machine possesses reduces below a predefined level, the scanning scheme turns from partitioned permutation scanning into simple permutation scanning.

Propagating the Malicious Code

We can identify three groups of mechanisms for propagating malicious code and building attack networks[4]:

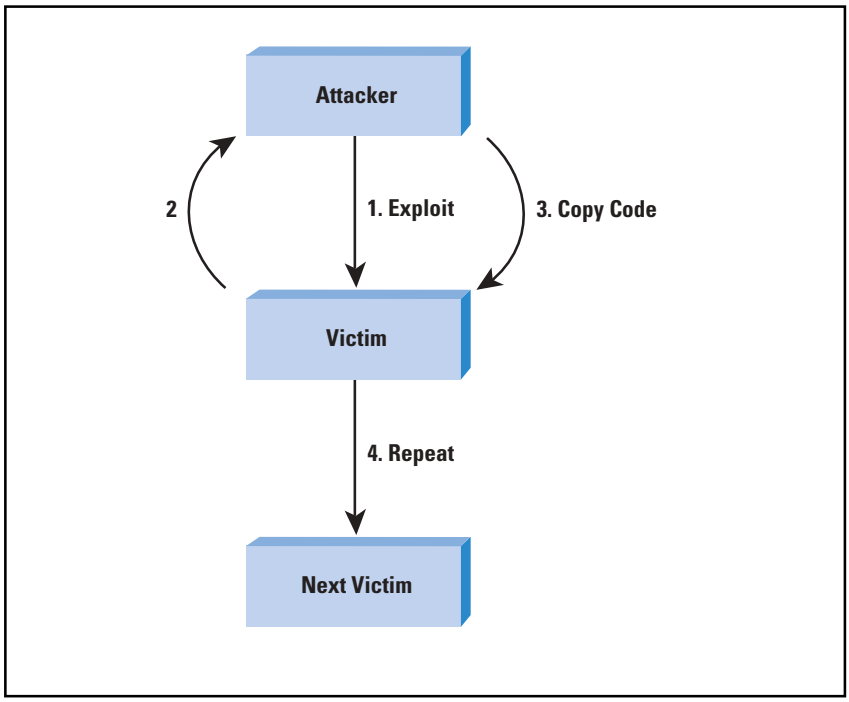
- *Central source propagation:* In this mechanism, after the discovery of the vulnerable system that will become one of the zombies, instructions are given to a central source so that a copy of the attack toolkit is transferred from a central location to the newly compromised system. After the toolkit is transferred, an automatic installation of the attack tools takes place on this system, controlled by a scripting mechanism. That initiates a new attack cycle, where the newly infected system looks for other vulnerable computers on which it can install the attack toolkit using the same process as the attacker. Like other file-transfer mechanisms, this mechanism commonly uses HTTP, FTP, and *remote-procedure call* (RPC) protocols. A graphical representation of this mechanism is shown in Figure 1.

Figure 1: Central Source Propagation



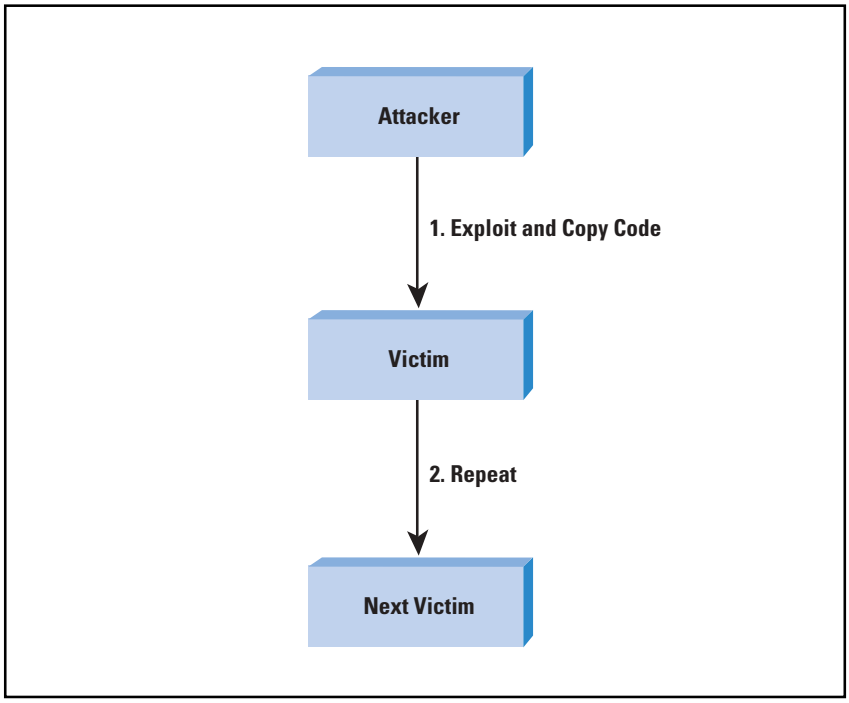
- *Back-chaining propagation:* In this mechanism, the attack toolkit is transferred to the newly compromised system from the attacker. More specifically, the attack tools that are installed on the attacker include special methods for accepting a connection from the compromised system and sending a file to it that contains the attack tools. This back-channel file copy can be supported by simple port listeners that copy file contents or by full intruder-installed Web servers, both of which use the *Trivial File Transfer Protocol* (TFTP). Figure 2 presents the this mechanism:

Figure 2: Back-Chaining Propagation



- *Autonomous propagation:* In this mechanism, the attacking host transfers the attack toolkit to the newly compromised system at the exact moment that it breaks into that system. This mechanism differs from the previously mentioned mechanisms in that the attack tools are planted into the compromised host by the attackers themselves and not by an external file source. Figure 3 shows the autonomous propagation.

Figure 3: Autonomous Propagation



After the construction of the attack network, the intruders use handler machines to specify the attack type and the victim's address and wait for the appropriate moment in order to mount the attack. Then, either they remotely command the launch of the chosen attack to agents or the daemons "wake up" simultaneously, as they had been programmed to do. The agent machines in turn begin to send a stream of packets to the victim, thereby flooding the victim's system with useless load and exhausting its resources. In this way, the attackers render the victim machine unavailable to legitimate clients and obtain unlimited access to it, so that they can inflict arbitrary damage. The volume of traffic may be so high that the networks that connect the attacking machines to the victim may also suffer from lower performance. Hence the provision of services over these networks is no longer possible, and in this way their clients are denied those services. Thus, the network that has been burdened by the attack load can be considered as one more victim of the DDos attack.

The whole procedure for carrying out a DDoS attack is mostly automated thanks to various attack tools. According to[5], the existence of the first controllable DDOS tool was reported by the *CERT Coordination Center* (CERT/CC) in early 1998 and it was called "Fapi." It is a tool that does not provide easy controls for setting up the DDoS network and does not handle networks with more than 10 hosts very well. In mid-1999 Trinoo arrived. Later that year the existence of *Tribe Flood Network* (TFN) and its upgraded version TFN2K (or TFN2000) was reported. Stacheldraht (German for "barbed wire") evolved out of the latter two tools (Trinoo and TFN). This tool is remarkable because it has full-control features and a Blowfish-encrypted control channel for the attacker. Moreover, in early 2000 it mutated into StacheldrahtV4, and later into Stacheldraht v1.666.

However, the development of attack tools did not stop, and many tools were later introduced, such as Mstream, Omega, Trinity, Derivatives, myServer, and Plague[6]. Dave Dittrich and his partners have provided the most comprehensive analyses of the Trinoo, Tribe Flood Network, Stacheldraht, shaft, and mstream DDoS attack tools[7]. Through this work, a lot of malicious code was captured, important observations were made about DDoS attack tools, and solutions were proposed toward detection and defense.

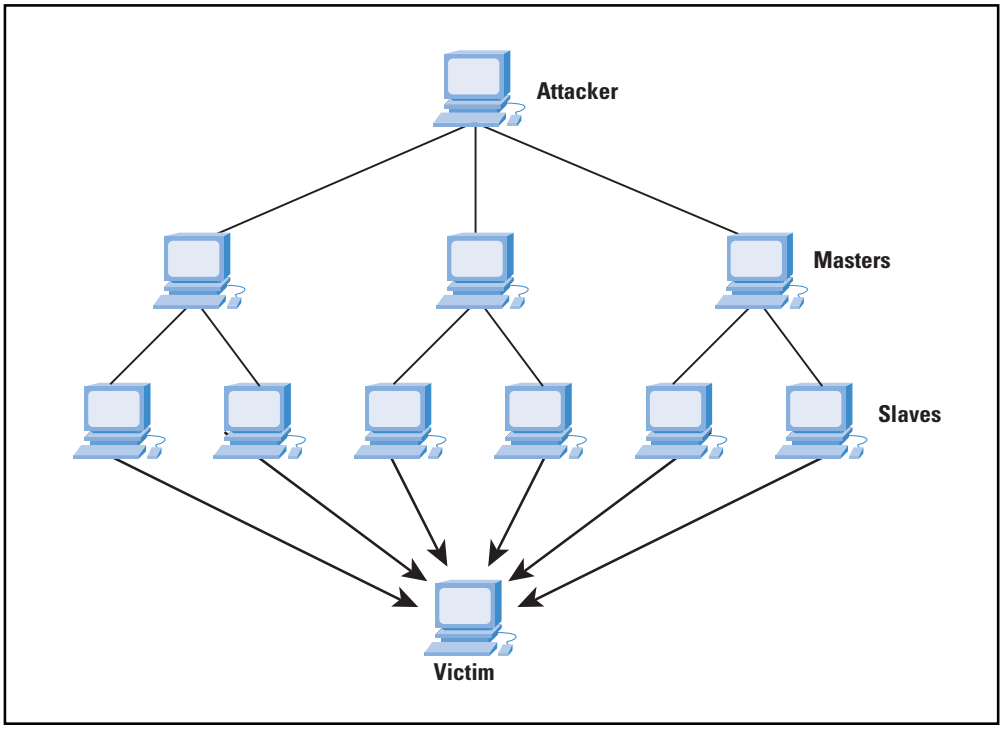
DDoS Attack Taxonomy

As has been already said, a DDoS attack takes place when many compromised machines infected by the malicious code act simultaneously and are coordinated under the control of a single attacker in order to break into the victim's system, exhaust its resources, and force it to deny service to its customers. There are mainly two kinds of DDoS attacks[10]: typical DDoS attacks and *distributed reflector DoS* (DRDoS) attacks. The following paragraphs describe these two kinds analytically.

Typical DDoS Attacks

In a typical DDoS attack, the army of the attacker consists of *master zombies* and *slave zombies*. The hosts of both categories are compromised machines that have arisen during the scanning process and are infected by malicious code. The attacker coordinates and orders master zombies and they, in turn, coordinate and trigger slave zombies. More specifically, the attacker sends an attack command to master zombies and activates all attack processes on those machines, which are in hibernation, waiting for the appropriate command to wake up and start attacking. Then, master zombies, through those processes, send attack commands to slave zombies, ordering them to mount a DDoS attack against the victim. In that way, the agent machines (slave zombies) begin to send a large volume of packets to the victim, flooding its system with useless load and exhausting its resources. Figure 4 shows this kind of DDoS attack.

Figure 4: A DDoS Attack



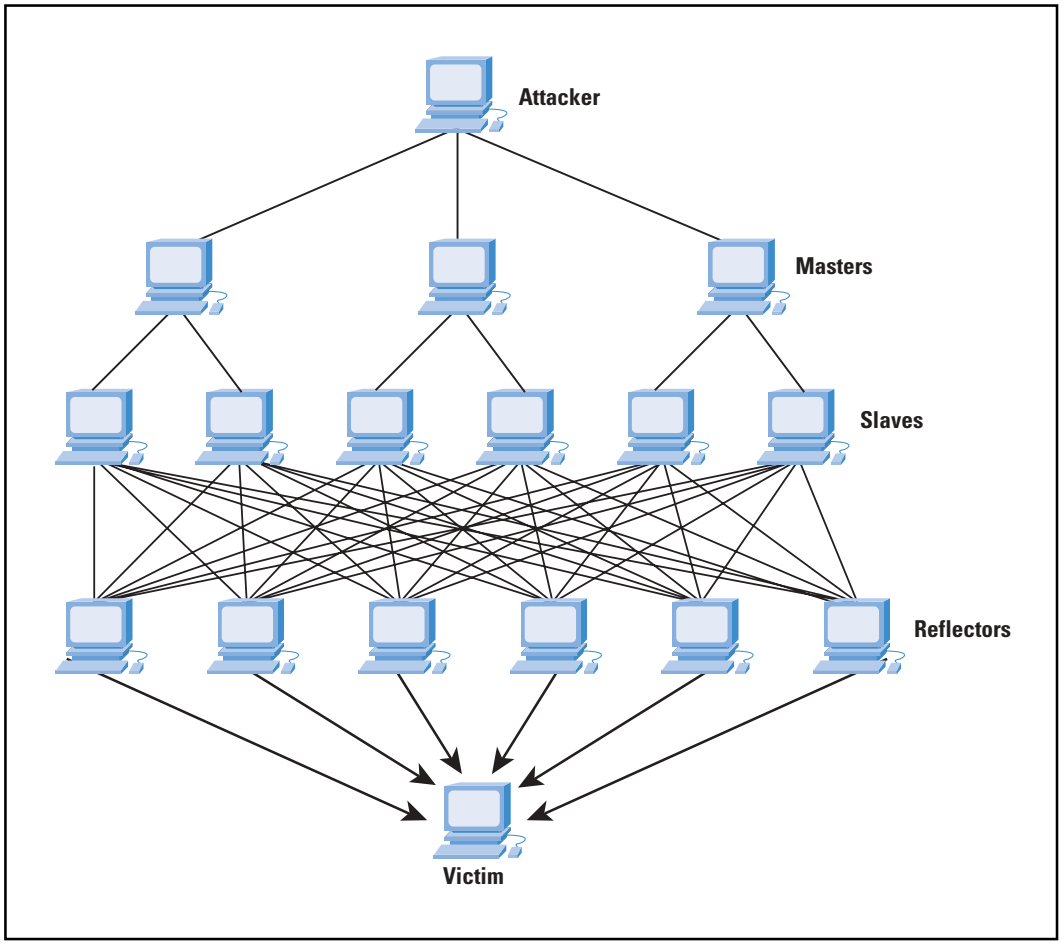
In cases of DDoS attacks, spoofed source IP addresses are used in the packets of the attack traffic. An attacker prefers to use such counterfeit source IP addresses for two major reasons: first, the attackers want to hide the identity of the zombies so that the victim cannot trace the attack back to them. The second reason concerns the performance of the attack. The attackers want to discourage any attempt of the victim to filter out the malicious traffic.

DRDoS Attacks

Unlike typical DDoS attacks, in DRDoS attacks the army of the attacker consists of master zombies, slave zombies, and reflectors[11]. The scenario of this type of attack is the same as that of typical DDoS attacks up to a specific stage. The attackers have control over master zombies, which, in turn, have control over slave zombies. The difference in this type of attack is that slave zombies are led by master zombies to send a stream of packets with the victim's IP address as the source IP address to other uninfected machines (known as *reflectors*), exhorting these machines to connect with the victim. Then the reflectors send the victim a greater volume of traffic, as a reply to its exhortation for the opening of a new connection, because they believe that the victim was the host that asked for it. Therefore, in DRDoS attacks, the attack is mounted by noncompromised machines, which mount the attack without being aware of the action.

Comparing the two scenarios of DDoS attacks, we should note that a DRDoS attack is more detrimental than a typical DDoS attack. This is because a DRDoS attack has more machines to share the attack, and hence the attack is more distributed. A second reason is that a DRDoS attack creates a greater volume of traffic because of its more distributed nature. Figure 5 graphically depicts a DRDoS attack.

Figure 5: A DRDoS Attack



Well-Known DDoS Attacks

This article would be incomplete without reference to some of the most well-known DDoS attacks. Some of the most famous documented DDoS attacks[12][13] are summarized in the following:

- *Apache2:* This attack is mounted against an Apache Web server where the client asks for a service by sending a request with many HTTP headers. However, when an Apache Web server receives many such requests, it cannot confront the load and it crashes.

- *ARP Poison: Address Resolution Protocol* (ARP) Poison attacks require the attacker to have access to the victim's LAN. The attacker deludes the hosts of a specific LAN by providing them with wrong MAC addresses for hosts with already-known IP addresses. This can be achieved by the attacker through the following process: The network is monitored for "arp who-has" requests. As soon as such a request is received, the malevolent attacker tries to respond as quickly as possible to the questioning host in order to mislead it for the requested address.
- *Back:* This attack is launched against an apache Web server, which is flooded with requests containing a large number of front-slash (/) characters in the URL description. As the server tries to process all these requests, it becomes unable to process other legitimate requests and hence it denies service to its customers.
- *CrashIIS:* The victim of a CrashIIS attack is commonly a Microsoft Windows NT IIS Web server. The attacker sends the victim a malformed GET request, which can crash the Web server.
- *DoSNuke:* In this kind of attack, the Microsoft Windows NT victim is inundated with "out-of-band" data (MSG_OOB). The packets being sent by the attacking machines are flagged "urg" because of the MSG_OOB flag. As a result, the target is weighed down, and the victim's machine could display a "blue screen of death."
- *Land:* In Land attacks, the attacker sends the victim a TCP SYN packet that contains the same IP address as the source and destination addresses. Such a packet completely locks the victim's system.
- *Mailbomb:* In a Mailbomb attack, the victim's mail queue is flooded by an abundance of messages, causing system failure.
- *SYN Flood:* A SYN flood attack occurs during the three-way handshake that marks the onset of a TCP connection. In the three-way handshake, a client requests a new connection by sending a TCP SYN packet to a server. After that, the server sends a SYN/ACK packet back to the client and places the connection request in a queue. Finally, the client acknowledges the SYN/ACK packet. If an attack occurs, however, the attacker sends an abundance of TCP SYN packets to the victim, obliging it both to open a lot of TCP connections and to respond to them. Then the attacker does not execute the third step of the three-way handshake that follows, rendering the victim unable to accept any new incoming connections, because its queue is full of half-open TCP connections.
- *Ping of Death:* In Ping of Death attacks, the attacker creates a packet that contains more than 65,536 bytes, which is the limit that the IP protocol defines. This packet can cause different kinds of damage to the machine that receives it, such as crashing and rebooting.

- *Process Table:* This attack exploits the feature of some network services to generate a new process each time a new TCP/IP connection is set up. The attacker tries to make as many uncompleted connections to the victim as possible in order to force the victim's system to generate an abundance of processes. Hence, because the number of processes that are running on the system cannot be boundlessly large, the attack renders the victim unable to serve any other request.
- *Smurf Attack:* In a "smurf" attack, the victim is flooded with *Internet Control Message Protocol* (ICMP) "echo-reply" packets. The attacker sends numerous ICMP "echo-request" packets to the broadcast address of many subnets. These packets contain the victim's address as the source IP address. Every machine that belongs to any of these subnets responds by sending ICMP "echo-reply" packets to the victim. Smurf attacks are very dangerous, because they are strongly distributed attacks.
- *SSH Process Table:* Like the Process Table attack, this attack makes hundreds of connections to the victim with the *Secure Shell* (SSH) Protocol without completing the login process. In this way, the daemon contacted by the SSH on the victim's system is obliged to start so many SSH processes that it is exhausted.
- *Syslogd:* The Syslogd attack crashes the *syslogd* program on a Solaris 2.5 server by sending it a message with an invalid source IP address.
- *TCP Reset:* In TCP Reset attacks, the network is monitored for "tcp-connection" requests to the victim. As soon as such a request is found, the malevolent attacker sends a spoofed TCP RESET packet to the victim and obliges it to terminate the TCP connection.
- *Teardrop:* While a packet is traveling from the source machine to the destination machine, it may be broken up into smaller fragments, through the process of fragmentation. A Teardrop attack creates a stream of IP fragments with their offset field overloaded. The destination host that tries to reassemble these malformed fragments eventually crashes or reboots.
- *UDP Storm:* In a *User Datagram Protocol* (UDP) connection, a character generation ("chargen") service generates a series of characters each time it receives a UDP packet, while an echo service echoes any character it receives. Exploiting these two services, the attacker sends a packet with the source spoofed to be that of the victim to another machine. Then, the echo service of the former machine echoes the data of that packet back to the victim's machine and the victim's machine, in turn, responds in the same way. Hence, a constant stream of useless load is created that burdens the network.

The first DoS attack occurred against Panix, the New York City area's oldest and largest *Internet Service Provider* (ISP), on September 6, 1996, at about 5:30 p.m.[14]. The attack was against different computers on the provider's network, including mail, news, and Web servers, user "login" machines, and name servers. The Panix attack was a SYN Flood attack deriving from random IP addresses and directed toward server *Simple Mail Transfer Protocol* (SMTP) ports. More specifically, Panix's computers were flooded by, on average, 150 SYN packets per second (50 per host), so Panix could not respond to legitimate requests[15]. Because the attackers used spoofed source IP addresses in their packets, the addresses could not be traced and malicious traffic could not be filtered. For that reason the attack was not immediately confronted. The solution was to use a special structure, instead of full *Transmission Control Block* (TCB), to hold half-open connections until the last ACK packet was received. In that way, the listen queue was large enough to keep all the SYN requests before the half-open connection timed out. The timeout, on the other hand, was adjusted to 94 seconds[16]. However, although Panix overcame this attack, the new threat (DoS attacks) made administrators worry.

Problems Caused and Countermeasures

The results of these attacks are disastrous. DDoS attacks have two characteristics: they are both distributed attacks and denial-of-service attacks. Distributed means that they are large-scale attacks having a great impact on the victims. Denial of service means that their goal is to deny the victim's access to a particular resource (service). This is not too difficult because the Internet was not designed with security in mind.

First, available *bandwidth* is one of the "goods" that attackers try to consume. Flooding the network with useless packets, for example, prevents legitimate ICMP echo packets from traveling over the network. Secondly, attackers try to consume *CPU power*. By generating several thousands of useless processes on the victim's system, attackers manage to fully occupy memory and process tables. In this way the victim's computer cannot execute any process and the system breaks down. Using this method, the attacker manages to prevent clients from accessing the victim's services and disrupts the current connections. Finally, attackers try to occupy victims' *services* so that no one else can access them. For example, by leaving TCP connections half open, attackers manage to consume the victim's data structures, and when they do so, no one else can establish a TCP connection with that victim.

The impact of these attacks is catastrophic, especially when victims are not individuals but companies. DDoS attacks prevent victims either from using the Internet, or from being reached by other people. Consequently, when the victim is an ISP, the results of such an attack are far more severe. ISPs' clients will not be served. E-business is also top on the "hit list." Being off line for a few hours could result in the loss of large sums of money for an ISP. Finally, the fact that companies use the Internet more and more for advertising or for providing goods and services increases the severity of such incidents.

Defense Mechanisms

From the beginning, all legitimate users have tried to respond against these threats. University communities and software corporations have proposed several methods against the DDoS threat. Despite the efforts, the solution remains a dream. The attackers manage to discover other weaknesses of the protocols and—what is worse—they exploit the defense mechanisms in order to develop attacks. They discover methods to overcome these mechanisms or they exploit them to generate false alarms and to cause catastrophic consequences.

Many experts have tried to classify the DDoS defense mechanisms in order to clarify them. This classification gives users an overall view of the situation and helps defense-mechanism developers cooperate against the threat. The basic discrimination is between *preventive* and *reactive* defense mechanisms.

Preventive Mechanisms

The preventive mechanisms try to eliminate the possibility of DDoS attacks altogether or to enable potential victims to endure the attack without denying services to legitimate clients. With regard to attack prevention, countermeasures can be taken on victims or on zombies. This means modification of the system configuration to eliminate the possibility of accepting a DDoS attack or participating unwillingly in a DDoS attack. Hosts should guard against illegitimate traffic from or toward the machine. By keeping protocols and software up-to-date, we can reduce the weaknesses of a computer. A regular scanning of the machine is also necessary in order to detect any "anomalous" behavior. Examples of system security mechanisms include monitoring access to the computer and applications, and installing security patches, firewall systems, virus scanners, and intrusion detection systems automatically. The modern trend is toward security companies that guard a client's network and inform the client in case of attack detection to take defending measures. Several sensors monitor the network traffic and send information to a server in order to determine the "health" of the network. Securing the computer reduces the possibility of being not only a victim, but also a zombie. Not being a zombie is very important because it wipes out the attacker's army. All these measures can never be 100-percent effective, but they certainly decrease the frequency and strength of DDoS attacks.

Many other measures can be taken in order to reduce the attacker's army or restrict its "power." Studying the attack methods can lead to recognizing loopholes in protocols. For example, administrators could adjust their network gateways in order to filter input and output traffic. The source IP address of output traffic should belong to the subnetwork, whereas the source IP address of input traffic should not. In this way, we can reduce traffic with spoofed IP addresses on the network[28].

Furthermore, over the last few years, several techniques have been proposed to test systems for possible drawbacks, before their shipment to the market. More precisely, by replacing the components of a system with malicious ones we can discover whether the system can survive an attack situation[38]. If the system breaks down, a drawback has been detected and developers must correct it.

On the other hand, DoS prevention mechanisms enable the victim to endure attack attempts without denying service to legitimate clients. Until now, two methods have been proposed for this scenario. The first one refers to policies that increase the privileges of users according to their behavior. When users' identities are verified, then no threat exists. Any illegitimate action from those users can lead to their legal prosecution. The second method is usually too expensive; it involves increasing the effective resources to such a degree that DDoS effects are limited. Most of the time application of such a measure is impossible.

Reactive Mechanisms

The reactive mechanisms (also referred to as *Early Warning Systems*) try to detect the attack and respond to it immediately. Hence, they restrict the impact of the attack on the victim. Again, there is the danger of characterizing a legitimate connection as an attack. For that reason it is necessary for researchers to be very careful.

The main detection strategies are *signature detection, anomaly detection,* and *hybrid systems*. Signature-based methods search for patterns (signatures) in observed network traffic that match known attack signatures from a database. The advantage of these methods is that they can easily and reliably detect known attacks, but they cannot recognize new attacks. Moreover, the signature database must always be kept up-to-date in order to retain the reliability of the system.

Anomaly-based methods compare the parameters of the observed network traffic with normal traffic. Hence it is possible for new attacks to be detected. However, in order to prevent a false alarm, the model of "normal traffic" must always be kept updated and the threshold of categorizing an anomaly must be properly adjusted.

Finally, hybrid systems combine both these methods. These systems update their signature database with attacks detected by anomaly detection. Again the danger is great because an attacker can fool the system by characterizing normal traffic as an attack. In that case an *Intrusion Detection System* (IDS) becomes an attack tool. Thus IDS designers must be very careful because their research can boomerang.

After detecting the attack, the reactive mechanisms respond to it. The relief of the impact of the attack is the primary concern. Some mechanisms react by limiting the accepted traffic rate. This means that legitimate traffic is also blocked. In this case the solution comes from traceback techniques that try to identify the attacker. If attackers are identified, despite their efforts to spoof their address, then it is easy to filter their traffic. Filtering is efficient only if attackers' detection is correct. In any other case filtering can become an attacker's tool.

The University of Washington provides an example of attack detection. Dave Dittrich and his team of 40 people discovered that more than 30 of their systems were zombies exploited by a single attacker[39]. By monitoring network traffic, Dittrich's team located directory and file names uncommon to the Windows operating systems the attacker ran on the network, as well as the port through which all these files were running communications.

Difficulties in Defending

Development of detection and defending tools is very complicated. Designers must think in advance of every possible situation because every weakness can be exploited. Difficulties involve:

- DDoS attacks flood victims with packets. This means that victims cannot contact anyone else in order to ask for help. So it is possible for a network neighbor to be attacked, but nobody would know it and nobody can help. Consequently, any action to react can be taken only if the attack is detected early. But can an attack be detected early? Usually traffic flow increases suddenly and without any warning[34][35][36]. For this reason defense mechanisms must react quickly.
- Any attempt of filtering the incoming flow means that legitimate traffic will also be rejected. And if legitimate traffic is rejected, how will applications that wait for information react? On the other hand, if zombies number in the thousands or millions, their traffic will flood the network and consume all the bandwidth. In that case filtering is useless because nothing can travel over the network.
- Attack packets usually have spoofed IP addresses. Hence it is more difficult to trace back to their source. Furthermore, it is possible that intermediate routers and ISPs may not cooperate in this attempt. Sometimes attackers, by spoofing source IP addresses, create counterfeit armies. Packets might derive from thousands of IP addresses, but zombies number only a few tens, for example.
- Defense mechanisms are applied in systems with differences in software and architecture. Also systems are managed by users with different levels of knowledge. Developers must design a platform independent of all these parameters.[37]

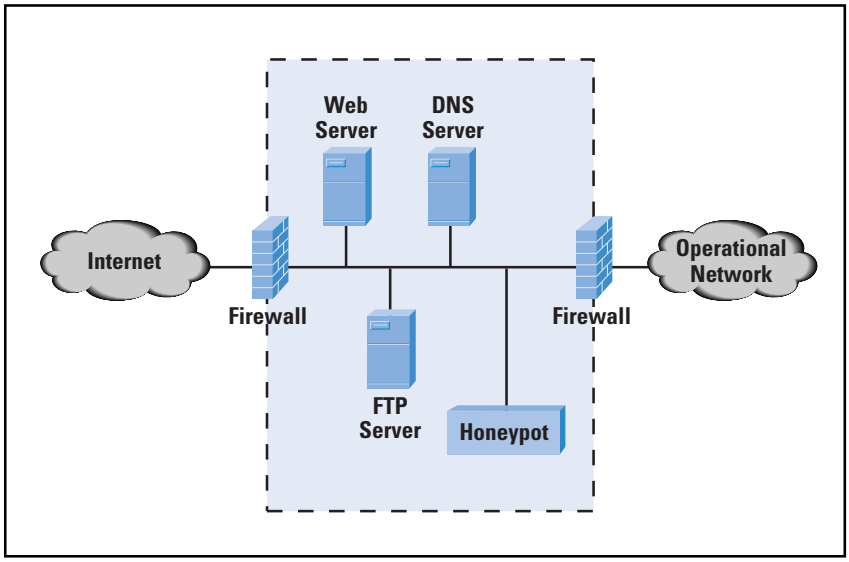
Modern Tendencies in Defending Against DDoS Attacks

Until now, developers have not managed to develop a 100-percent-effective defense mechanism. All mechanisms that have been presented either can confront only specific DDoS attacks or are being finally compromised by the attackers. Therefore, developers are currently working on DDoS diversion systems. *Honeypots* are the best representative of this category (See Figure 6).

Honeypots

There are two basic types of honeypots: *low-interaction honeypots* and *high-interaction honeypots*. The first ones refer to emulating services and operating systems. It is easy and safe to implement them. Attackers are not allowed to interact with the basic operating system, but only with specific services. For that reason, this type of honeypot cannot provide detailed informations for attackers' actions and they can easily be detected. However, they can detect communication attempts toward unused IP addresses. In that case an alarm is triggered, warning that someone is trying to compromise or attack the network. But what happens if the attack is not directed against the emulated service?

Figure 6: Honeypot



The answer comes from high-interaction honeypots. In [41], Honeynet is proposed. Honeynet is not a software solution that can be installed on a computer but a whole architecture, a network that is created to be attacked. Within this network, every activity is recorded and attackers are being trapped. Encrypted SSH sessions, e-mails, file uploads, and every possible attacker's action is captured. Moreover, a Honeywall gateway allows incoming traffic, but controls outgoing traffic using intrusion prevention technologies. This allows the attacker to interact with Honeynet systems, but prevents the attacker from harming other non-Honeynet systems. By studying the captured traffic, researchers can discover new methods and tools and they can fully understand attackers' tactics. However, Honeynet systems are more complex to install and deploy and the risk is increased as attackers interact with real operating systems and not with emulations. But what would happen if someone did compromise such a system? The consequences could be disastrous.

Route Filter Techniques

Different suggestions for defending against DDoS attacks derive from the *Border Gateway Protocol* (BGP) community. When routing protocols were designed, developers did not focus on security, but effective routing mechanisms and routing loop avoidance. Early on, attackers started directing their attention towards routers. By gaining access to a router, they could direct the traffic over bottlenecks, view critical data, and modify them. Cryptographic authentication mitigates these threats. Because of neighbor authentication, the routing update comes from a trusted source and there is no possibility that someone can give routers invalid routing information in order to compromise a network. On the other hand, routing filters are necessary for preventing critical routes and subnetworks from being advertised and suspicious routes from being incorporated in routing tables. In that way, attackers do not know the route toward critical servers and suspicious routes are not used.

Two other route filter techniques, *blackhole routing* and *sinkhole routing,* can be used when the network is under attack. These techniques try to temporarily mitigate the impact of the attack. The first one directs routing traffic to a null interface, where it is finally dropped. At first glance, it would be perfect to "blackhole" malicious traffic. But is it always possible to isolate malicious from legitimate traffic? If victims know the exact IP address being attacked, then they can ignore traffic originating from these sources. This way, the attack impact is restricted because the victims do not consume CPU time or memory as a consequence of the attack. Only network bandwidth is consumed. However, if the attackers' IP addresses cannot be distinguished and all traffic is blackholed, then legitimate traffic is dropped as well. In that case, this filter technique fails.

Sinkhole routing involves routing suspicious traffic to a valid IP address where it can be analyzed. There, traffic that is found to be malicious is rejected (routed to a null interface); otherwise it is routed to the next hop. A sniffer on the sinkhole router can capture traffic and analyze it. This technique is not as severe as the previous one. The effectiveness of each mechanism depends on the strength of the attack. Specifically, sinkholing cannot react to a severe attack as effectively as blackholing. However, it is a more sophisticated technique, because it is more selective in rejecting traffic.

Filtering malicious traffic seems to be an effective countermeasure against DDoS. The closer to the attacker the filtering is applied, the more effective it is. This is natural, because when traffic is filtered by victims, they "survive," but the ISP's network is already flooded. Consequently, the best solution would be to filter traffic on the source; in other words, filter zombies' traffic.

Until now, three filtering possibilities have been reported concerning criteria for filters. The first one is filtering on the *source address*. This one would be the best filtering method, if we knew each time who the attacker is. However, this is not always possible because attackers usually use spoofed IP addresses. Moreover, DDoS attacks usually derive from thousands of zombies and this makes it too difficult to discover all the IP addresses that carry out the attack. And even if all these IP addresses are discovered, the implementation of a filter that rejects thousands of IP addresses is practically impossible to deploy.

The second filtering possibility is filtering on the *service*. This tactic presupposes that we know the attack mechanism. In this case, we can filter traffic toward a specific UDP port or a TCP connection or ICMP messages. But what if the attack is directed toward a very common port or service? Then we must either reject every packet (even if it is legitimate) or suffer the attack.

Finally, there is the possibility of filtering on the *destination address*. DDoS attacks are usually addressed to a restricted number of victims, so it seems to be easy to reject all traffic toward them. But this means that legitimate traffic is also rejected. In case of a large-scale attack, this should not be a problem because the victims will soon break down and the ISP will not be able to serve anyone. So filtering prevents victims from breaking down by simply keeping them isolated.

Fred Baker and Paul Ferguson developed an technique called *Ingress Filtering* for mitigating DoS attacks (and, later, DDoS attacks too). After the Panix attack and a few other attacks, Paul Ferguson wrote RFC 2267[42], which became *Best Current Practices* (BCP) 38 in RFC 2827[43]. This RFC presents a method for using ingress traffic filtering against DoS attacks that use forged IP addresses and try to be propagated from "behind" an ISP's aggregation point. This method prevents the attack from forged source addresses, but nothing can be done against an attack from a valid source address. However, in that case, if the attack is detected, it is easy to trace the attacker. Finally, although this solution allows the network to protect itself from other attacks too (for example, spoofed management access to networking equipment), it can also create some problems, for example, with multihoming.

For that reason, RFC 2827 was recently (March 2004) updated by Fred Baker in BCP 84/ RFC 3704[44]. This RFC describes and evaluates the current ingress filtering mechanisms, examines some implementation matters related to ingress filtering, and presents some solutions to ingress filtering with multihoming. According to this RFC, ingress filtering should be implemented at multiple levels in order to prohibit the use of spoofed addresses and to make attackers more traceable, even if asymmetric/multihomed networks are presented. However, although Ferguson's work was published a long time ago, service providers in some cases ignore his suggestions.

Hybrid Methods and Guidelines

Currently researchers try to combine the advantages from all the methods stated previously in order to minimize their disadvantages. As a result, several mechanisms that implement two or more of these techniques are proposed for mitigation of the impact of DDoS attacks. The best solution to the DDoS problem seems to be the following: victims must detect that they are under attack as early as possible. Then they must trace back the IP addresses that caused the attack and warn zombies administrators about their actions. In that way, the attack can be confronted effectively.

However, as we saw previously, this is currently impossible. The lack of a 100-percent-effective defending tool imposes the necessity of private alerts. Users must care for their own security. Some basic suggestions follow:

- Prevent installation of distributed attack tools on our systems. This will help to restrict the zombies army. Several tasks also need to be performed. First, keep protocols and operating systems up-to-date. We can prevent system exploitation by eliminating the number of weaknesses of our system.
- Use firewalls in gateways to filter incoming and outgoing traffic. Incoming packets with source IP addresses belonging to the subnetwork and outgoing packets with source IP addresses not belonging to the subnetwork are not logical.
- Deploy IDS systems to detect patterns of attacks.
- Deploy antivirus programs to scan malicious code in our system.

Further Thoughts

The Internet is not stable—it reforms itself rapidly. This means that DDoS countermeasures quickly become obsolete. New services are offered through the Internet, and new attacks are deployed to prevent clients from accessing these services. However, the basic issue is whether DDoS attacks represent a network problem or an individual problem—or both. If attacks are mainly a network problem, a solution could derive from alterations in Internet protocols. Specifically, routers could filter malicious traffic, attackers could not spoof IP addresses, and there would be no drawback in routing protocols. If attacks are mostly the result of individual system weaknesses, the solution could derive from an effective IDS system, from an antivirus, or from an invulnerable firewall. Attackers then could not compromise systems in order to create a "zombies" army. Obviously, it appears that both network and individual hosts constitute the problem. Consequently, countermeasures should be taken from both sides. Because attackers cooperate in order to build the perfect attack methods, legitimate users and security developers should also cooperate against the threat. The solution will arise from combining both network and individual countermeasures.

References

[1] Kevin Tsui, "Tutorial-Virus (Malicious Agents)," University of Calgary, October 2001.

[2] Nicholas Weaver, "Warhol Worms: The Potential for Very Fast Internet Plagues,"
http://www.iwar.org.uk/comsec/resources/worms/warhol-worm.htm

[3] Nicholas Weaver, U.C. Berkeley BRASS group, "Potential Strategies for High Speed Active Worms: A Worst Case Analysis," February 2002

[4] David Moore and Colleen Shannon, "The Spread of the Code Red Worm (crv2)," July 2001,
http://www.caida.org/analysis/security/codered/coderedv2_analysis.xml#animations

[5] "A Chronology of CERT Coordination Center Involvement with Distributed Denial-of-Service Tools,"
http://www.cdt.org/security/dos/000229senatehouse/chron.html

[6] "Analyzing Distributed Denial Of Service Tools: The Shaft Case," Sven Dietrich, NASA Goddard Space Flight Center; Neil Long, Oxford University; David Dittrich, University of Washington,
http://www.usenix.org/events/lisa2000/full_papers/dietrich/dietrich_html/

[7] **http://staff.washington.edu/dittrich**

[8] Kevin J. Houle, CERT/CC; George M. Weaver, CERT/CC, in collaboration with: Neil Long, Rob Thomas, "Trends in Denial of Service Attack Technology," V1.0, October 2001.

[9] **http://staff.washington.edu/dittrich/misc/stacheldraht.analysis**

[10] T. Peng, C. Leckie, and K. Ramamohanarao, "Detecting Distributed Denial of Service Attacks Using Source IP Address Monitoring," The University of Melbourne, Australia, 2003.

[11] Steve Gibson, "Distributed Reflection Denial of Service Description and Analysis of a Potent, Increasingly Prevalent, and Worrisome Internet Attack," February 2002.

[12] **http://www.ll.mit.edu/IST/ideval/docs/1999/attackDB.html**

[13] Yanet Manzano, "Tracing the Development of Denial of Service Attacks: A Corporate Analogy," 2003,
http://www.acm.org/crossroads/xrds10-1/tracingDOS.html

[14] **http://www.panix.com/press/synattack.html**

[15] **http://cypherpunks.venona.com/date/1996/09/msg01055.html**

[16] **http://cypherpunks.venona.com/date/1996/09/msg01061.html**

[17] Larry Rogers, "What Is a Distributed Denial of Service (DDoS) Attack and What Can I Do About It?" February 2004, **http://www.cert.org/homeusers/ddos.html**

[18] Alefiya Hussain, John Heidemann, and Christos Papadopoulo, "A Framework for Classifying Denial of Service Attacks," 25 February 2003.

[19] **http://www.cs.berkeley.edu/~nweaver/warhol.old.html**

[20] CIS 659 "Introduction to Network Security – Fall 2003," **http://www.cis.udel.edu/~sunshine/F03/CIS659/class15.pdf**

[21] Miguel Vargas Martin, School of Computer Science, Carleton University, "Overview of Worms and Defence Strategies," October 2003.

[22] "Computer Security," Testimony of Richard D. Pethia, Director, CERT Centers Software Engineering Institute, Carnegie Mellon University, March 2000, **http://www.cert.org/congressional_testimony/Pethia_testimony_Mar9.html#Distributed**

[23] Jelena Mirkovic, Janice Martin, and Peter Reiher, UCLA, "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms."

[24] Distributed Denial of Service Tools, **http://www.cert.org/incident_notes/IN-99-07.html**

[25] Barbara Fraser, Lawrence Rogers, and Linda Pesante, "Was the Melissa Virus So Different?" *The Internet Protocol Journal*, Volume 2, No. 2, June 1999.

[26] **http://news.bbc.co.uk/1/hi/sci/tech/635444.stm**

[27] **http://www.nta-monitor.com/newrisks/feb2000/yahoo.htm**

[28] **http://www.cert.org/advisories/CA-1996-21.html**

[29] S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," Technical Report 99-15, Department of Computer Engineering, Chalmers University, March 2000.

[30] J. Shapiro and N. Hardy, "EROS: A principle-driven Operating System from the Ground Up," *IEEE Software*, pp. 26–33, January/February 2002.

[31] A. Garg and A. L. Narasimha Reddy, "Mitigating Denial of Service Attacks Using QoS Regulation," Texas A & M University Tech report, TAMU-ECE-2001-06.

[32] Y. L. Zheng and J. Leiwo, "A method to implement a Denial of Service Protection Base," *Information Security and Privacy*, Volume 1270 of *Lecture Notes in Computer Science* (LNCS), pp. 90–101, 1997.

[33] CERT on Home Network Security:
http://www.cert.org/tech_tips/home_networks.html

[34] CERT on SMURF Attacks:
http://www.cert.org/advisories/CA-1998-01.html

[35] CERT on TCP SYN Flooding Attacks:
http://www.cert.org/advisories/CA-1996-21.html

[36] CERT TRIN00 Report:
http://www.cert.org/incident_notes/IN-99-07.html#trinoo

[37] **http://falcon.jmu.edu/~flynngn/whatnext.htm**

[38] Charalampos Patrikakis, Thomas Kalamaris, Vaios Kakavas, "Performing Integrated System Tests Using Malicious Component Insertion," *Electronic Notes in Theoretical Computer Science*, Volume 82 No. 6 (2003).

[39] **http://www.paypal.com/html/computerworld-011402.html**

[40] Ho Chung, "An Evaluation on Defensive Measures against Denial-of-Service Attacks," Fall 2002.

[41] Nathalie Weiler, "Honeypots for Distributed Denial of Service Attacks,"
www.tik.ee.ethz.ch/~weiler/papers/wetice02.pdf

[42] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," RFC 2267, January 1998.

[43] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," RFC 2827, May 2000.

[44] F. Baker and P. Savola, "Ingress Filtering for Multihomed Networks," RFC 3704, March 2004.

[45] Taxonomies of Distributed Denial of Service Networks, Attacks, Tools, and Countermeasures:
www.ee.princeton.edu/~rblee/DDoS%20Survey%20Paper_v7final.doc

[46] Lance Spitzner, "Honeypots Definitions and Value of Honeypots," May 2003, **http://www.tracking-hackers.com**

[47] How to Get Rid of Denial of Service Attacks:
http://www.bgpexpert.com/antidos.php

[48] Proposed Solutions to DDoS Information, March 2001:
http://www.cs.virginia.edu/~survive/ddos/ddos_solutions.html

[49] Dennis Fisher, "Thwarting the Zombies," March 2003:
http://www.eweek.com/article2/0,3959,985389,00.asp

[50] Merike Kaeo, "Route to Security," March 2004,
http://infosecuritymag.techtarget.com/ss/0,295796,sid6_iss346_art668,00.html

[51] "Report to the President's Commission on Critical Infrastructure Protection," James Ellis, David Fisher, Thomas Longstaff, Linda Pesante, and Richard Pethia, January 1997,
http://www.cert.org/pres_comm/cert.rpcci.body.html

[52] "Cisco Quality of Service and DDOS, Engineering Issues for Adaptive Defense Network," MITRE, 7/25/2001.

[53] "Denial of Service Attacks," CERT Coordination Center, June 4, 2001,
http://www.cert.org/tech_tips/denial_of_service.html

[54] Tom Chen, "Trends in Viruses and Worms," *The Internet Protocol Journal,* Volume 6, No. 3, September 2003.

CHARALAMPOS Z. PATRIKAKIS holds a Dipl.-Ing. and a Ph.D. degree from the Electrical Engineering and Computer Science Department of the National Technical University of Athens (NTUA). He is currently a senior research associate of the Telecommunications Labor-atory of NTUA. He has participated in several European Union projects (ESPRIT, RACE, ACTS, IST). His main interests are in the area of IP service design and implementation, multicasting in IP networks, IP transport protocols, and media streaming over IP networks. He is a member of IEEE, a member of the Greek Computer Society, a certified trainer by the National Accreditation Centre of Vocational Training Structures and Accompanying Support Services, and a member of the Technical Chamber of Greece. He can be reached at: **bpatr@telecom.ntua.gr**

MICHALIS MASIKOS holds a Dipl.-Ing. degree from the Electrical Engineering and Computer Science Department of the National Technical University of Athens (NTUA). He is currently a research associate of the Telecommunications Laboratory of NTUA. His interests are in the fields of network security, network simulation, and analysis. He can be reached at: **mmasik@telecom.ntua.gr**

OLGA ZOURARAKI holds a Dipl.-Ing. degree from the Electrical Engineering and Computer Science Department of the National Technical University of Athens (NTUA). She is currently a research associate of the Telecommunications Laboratory of NTUA. Her interests are in the fields of network security, Internet application design, and implementation. She can be reached at: **ozour@telecom.ntua.gr**

Letter to the Editor

Ole,

I was reading your latest issue of IPJ (Volume 7, No. 3, September 2004) and I could be wrong but I think you mis-typed an explanation about the STUN protocol. On page 12, 3rd paragraph, last sentence, it reads: "A received response indicates the presence of a port-restricted cone, and the lack of a response indicates the presence of a restricted cone."

According to the definitions you gave about "restricted cone" and "port-restricted cone" on pages 10 and 11. Shouldn't this sentence instead read: "A received response indicates the presence of a restricted cone, and the lack of a response indicates the presence of a port-restricted cone."

—Ryan Liles
ryanliles@hotmail.com

The author responds:

Ryan is correct, there is an error here in the text.

The flow control of the sequence of STUN tests is detailed in Figure 9 of the article. The test referred to here is to determine if the NAT is a restricted cone NAT, or a port-restricted cone NAT.

The restricted cone NAT, in Figure 7, is one where the NAT binding is accessible using any source port number on the external host when responding to a UDP packet from the internal sending host.

The port-restricted cone NAT, in Figure 8, is one where the NAT binding is accessible using the same port number as originally used by the internal lost host, and this binding is accessible from any external IP address.

The test referenced in this section, as per Figure 9, is one where the local host requests the external agent to respond using the same port number, but an altered source address. The text should read "This fourth request includes a control flag to direct the STUN server to respond using the alternate IP address, but with the same port value," in which case the interpretation of the response—that a response indicates the presence of a port-restricted cone NAT and the lack of response indicates the presence of a restricted cone NAT—would be correct.

Ryan is also correct in that if the test is performed the other way, requesting the agent to use the same IP address, but with the alternate port value, then the opposite interpretation would hold, namely that a response indicates the presence of a restricted cone NAT, and the lack of a response would indicate the presence of a port-restricted cone NAT, as Ryan points out.

Thanks to Ryan for following through this rather complex explanation of the STUN algorithm and spotting this error.

Regards,

—Geoff Huston, APNIC
gih@apnic.net

Book Review

The IP Multimedia Subsystem

The IP Multimedia Subsystem—Merging the Internet and the Cellular Worlds, by Gonzalo Camarillo and Miguel A. Garcia-Martin, John Wiley & Sons, 2004. ISBN 0470 87156 3.

The Internet and the cellular telephony system are the two most influential communication systems of the last half century. That the telecommunications industry would attempt to merge them into a single system was inevitable. The potential benefits are compelling—a single packet-based communication system with the capability to carry voice, video and data while providing ubiquitous wireless access and global mobility. The resulting system architecture is called the *Internet Multimedia Subsystem* (IMS) and is described comprehensively in this volume by Gonzalo Camarillo and Miguel A. Garcia Martin.

A "merging" of the two systems is only superficially what has happened. In practice, the IMS is an "embrace and extend" exercise which adapts the IP protocol suite to the existing architecture of the cellular telephony system. The cellular industry has taken a broad collection of IP protocols and mapped them onto their existing architecture, effecting a "protocol transplant" into an environment somewhat different from the Internet. Among the protocols imported are IPv6, SIP, DHCP, DNS, SDP, RTP, IPSec, and DIAMETER. Many are adopted unaltered; some are profiled by introducing new configuration data and rules; others are extended in various ways. The authors navigate their way through the various parts of the system with clarity and confidence. They can speak with authority on the subject—both were major contributors to the design through their key roles in the IETF and 3GPP (*Third Generation Partnership Project*—the standardization body for third generation cellular systems).

The book is clearly written and logically organized. The first part explains the reasoning behind adopting Internet-style packet networking for cellular mobile systems and describes the evolution of the standardization efforts. Although interesting, much of this material can be skimmed by those only interested in the meaty technical material which follows. The authors then explain the general principles behind the IMS architecture, including how various requirements of the cellular telephony industry drove the choices, and particularly the perceived need to extend and adapt the protocols rather than use them as deployed on the Internet. The majority of the book is devoted to ex-plaining in considerable technical depth how the protocols have been modified and how they are intended to work when IMS is successfully deployed. While not for the faint of heart, the writing is extremely clear and logical and hence should be understandable by anyone with a moderate background in the principles of protocol and system design. One aspect of the organization is particularly helpful to readers unfamiliar with some of the protocols in their native Internet instantiation. The authors divide the material into blocks where they first describe the native Internet flavor of the protocol, and then introduce the IMS-specific extensions and modifications.

Much of the volume is devoted to the *Session Initiation Protocol* (SIP) as the core signaling plane for IMS. All aspects of session establishment and management are covered. In addition, the ancillary parts of the control system are covered, including *Authentication, Authorization, and Accounting* (AAA), Security, Session Policies, and Quality of Service. For completeness, the data plane is also covered briefly through a discussion of the 3GPP audio, video, and text encoders, plus material on the media transport protocols.

The book concludes with a substantial section on how services are build on top of the core IMS protocols. Two of the most important, *Presence* and *Instant Messaging*, get comprehensive treatment, with a briefer discussion of the push-to-talk application.

As an old time "IP-head," it is hard to come away from this deep exploration of IMS without a bit of trepidation. The hallmark of IP and the Internet are simplicity and generality. IMS arguably succeeds at the latter, but at the expense of almost numbing complexity. This was perhaps inevitable given that the goal was to adapt Internet packet technology to the cellular system, which is itself quite complex. IMS will be quite a challenge to deploy. It remains to be seen if transplanting IP into a cellular telephony architectural model will result in economically sustainable services for the service providers or if a more native peer-to-peer Internet approach will simply bypass all the fancy IMS elements and just use basic packet transport. Such a market experiment is currently playing out in the broadband access arena with the broadband pipe suppliers offering telephony-oriented services themselves via customized standards like PacketCable, while third parties like Vonage and Skype simply piggyback on basic IP packet transport.

The next few years will be interesting. Whatever the outcome, anyone needing to be technically conversant with the architecture and protocols of IMS will find *The IP Multimedia Subsystem* indispensable.

—David Oran
oran@cisco.com

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

March 2005 Volume 8, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Internet Protocol Version 6 (IPv6) continues to be the focus of much work within the IETF as well as throughout the world in numerous deployment projects. The success of IPv6 depends not only on the protocol itself but also on its interaction with existing services such as the *Domain Name System* (DNS). In our first article, David Malone looks at some issues with DNS servers and IPv6. If you are interested in following the progress of IPv6 deployment, you might want to visit The IPv6 Forum's Website at: **http://www.ipv6forum.org**

A couple of years ago I signed up for GSM cellphone service and later added GPRS data service to my account. With my Bluetooth-enabled phone and laptop, I can access the Internet from almost anywhere in the world. The service is neither particularly fast nor inexpensive, but for occasional use it works very well, and has "saved the day" for me numerous times. However, GPRS is not the only wide-area wireless data network technology. Kostas Pentikousis gives an overview of the many alternatives.

The term "Internet Governance" is not well-defined, but it is being used more frequently when speaking about such organizations as the *Internet Corporation for Assigned Names and Numbers* (ICANN). The formation of the *World Summit on the Information Society* (WSIS) and its *Working Group on Internet Governance* (WGIG) has certainly brought the term into sharper focus. Although governance is certainly not a technical protocol issue, we still believe that it is important for our readers to follow both the debate about and the actual evolution of Internet Governance issues. However, we fully appreciate that this is an area where opinions differ—and that is why the article by Geoff Huston on this topic is labeled "Opinion."

We remind you to visit our Website, **http://www.cisco.com/ipj**, where you can find back issues of this journal, search the index files, or make changes to your subscription information. Your feedback is also very much appreciated, so drop us a line at **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Misbehaving Name Servers and What They're Missing

by David Malone, Hamilton Institute, NUI Maynooth, Ireland

IPv6-capable hosts abound, and the number is growing. Evidence[1] shows that more than 2 million Windows XP machines are probing for 6to4[2] connectivity. When combined with deployments of Linux and BSD that have been shipping with IPv6 support enabled by default for some time, that is a sizable platform on which to build IPv6 applications. Most Web browsers (Internet Explorer, Mozilla, Opera) now support IPv6 if the underlying platform does, so that is a significant number of applications ready to start making IPv6 queries.

In fact, many of these applications are already looking for IPv6 addresses in the *Domain Name System* (DNS), even if IPv6 connectivity is not actually available. This usually does not result in a problem—the name server says there are no IPv6 records and the application falls back to IPv4. In a small number of cases, name servers running outdated or errant software are misbehaving when faced with a request for an IPv6 address.

The Problem

So, what problem are these name servers having with the request for IPv6 addresses? Well, the DNS stores different types of information, such as host names and addresses. Different types of data are stored using different record types. For example, IPv4 addresses are stored using a type "A" record and host names are stored using a type "PTR" record. Some new record types have been introduced for IPv6. The most important one is "AAAA," which is for storing IPv6 addresses. (Another type called "A6" was also introduced, but it is now consigned to experimental status because it proved too complicated in certain situations.)

When you issue a request to the DNS, you indicate the domain and type of record that you are interested in. If the server has records of that type for that domain, it replies, including those records. If the server has no records of that type, it should respond saying "there are no records of this type." If the domain does not exist, then the server should return a "no such domain" error.

However, the problems arise when the DNS server does something different, and some name servers behave badly when faced with a query for a type they do not explicitly know about. For the sake of simplicity, we will highlight three wrong reactions to an unknown query that have been observed. A more complete technical analysis of the problem can be found in[3].

The first reaction that people notice is that some name servers do not reply when faced with a query for an unknown type. In this case, the person who made the request waits a while before the request is reissued. Eventually the application falls back to IPv4. "Eventually" means anything from 10 seconds to 100 seconds, depending on the operating system and application—enough to irk the casual Web user.

The second reaction is more subtle. Here the name server returns a "no such domain" response. At first glance this may seem harmless enough—the query for an IPv4 address is issued quickly. However, DNS specifications say that the "no such domain" response may be cached. This means that the "A" query is never issued, and the system acts as if the domain does not exist.

The third reaction is that the server issues some other sort of incorrect response. Usually this is less serious than the two previous reactions, because other responses at worst result in a particular name server being considered "bad" and being avoided for future queries. This means that some better-behaved name server can answer the query.

The Extent of the Problem

Although sites with these problems are sometimes discussed on mailing lists, the extent of a problem is not always proportional to the coverage it receives. Historically, numerous online advertising companies have had load-balancing DNS servers that exhibit these symptoms. Because the content of an ad server is embedded in the Web pages of many organizations, this means a single errant DNS server can give the end user the impression that this problem is more widespread than it is.

To give some idea of the scale of the problem, Table 1 shows the results of querying the name servers for the names mentioned in a month's worth of Web proxy logs. The number of servers responding in each of the three ways mentioned (no reply, no such domain, or other error) is shown, along with a total. Also shown is the number of name servers that actually returned IPv6 addresses.

These results show that actually only a small number of name servers have this problem. Unfortunately, it also looks as if the number of name servers distributing IPv6 addresses is actually comparable. However, it does look like the proportion of problem name servers is decreasing over time.

Table 1: Responses to Name Queries

Nameservers that:	January 2004	April 2004	*August 2004*
Responded to type A	*16838*	*20631*	*17934*
Did not reply to type A	*64 (0.38%)*	*49 (0.24%)*	*36 (0.20%)*
Returned no such domain	*11 (0.07%)*	*19 (0.09%)*	*11 (0.06%)*
Returned other error	*22 (0.13%)*	*39 (0.19%)*	*11 (0.06%)*
Had any issue with AAAA	*97 (0.58%)*	*107 (0.52%)*	*58 (0.32%)*
Returned AAAA records	*105 (0.62%)*	*123 (0.60%)*	*18 (0.66%)*

Looking at Web logs to determine the size of the problem gives us a feeling for the number of name servers that need attention. Another interesting parameter to consider is the proportion of requests that might be subject to this problem. The answer would tell us how many queries might be mishandled if your name server cannot deal with new query types.

Looking at the queries for addresses at one authoritative name server shows that 65 percent of queries are for A records, 21 percent are for AAAA records, and 14 percent are for A6 records. Although this server is IPv6-capable and might attract more queries for AAAA records, even the root servers run by RIPE show that 10 percent of address queries are for IPv6 addresses.

The Solution

Some of the name servers that exhibit this problem are simply running old versions of DNS server software. If this is the case, then the fix is simple: *upgrade!*

A significant number of the remaining problem servers are running unusual name server software, and the only way to fix the problem is to have that software fixed. Where the name server software is maintained in house, there should be enough DNS expertise to resolve the issue when it is identified. Where DNS systems have been bought in, it can be difficult to get the relevant information to the developers who can make the necessary changes. Thus increasing awareness of the issue among DNS vendors and troubleshooters is important.

In some cases[5,6], discussions on Internet mailing lists has alerted those responsible for the server to the problem and the issue has been resolved. In other cases, feedback provided by users and customers has marked IPv6 conformance as an issue for future upgrades of a site's DNS infrastructure. Unfortunately, on some occasions, feedback has been ignored and the problem has persisted. This is maybe not so surprising because it is a subtle problem. The fact that it is IPv6-related means it is sometimes dismissed because the organization thinks "we have not begun IPv6 deployment yet, so it cannot affect us."

Where problems have persisted, people have resorted to various practical solutions (hacks?) to avoid the issue. Some people, who do not need IPv6 at this time, have just suppressed the AAAA queries. Others, when they discover a name server that times out, add it to a blacklist. This avoids any delays, but may make a site unavailable. Mozilla includes a more forgiving style of blacklisting, in the form of a "ipv4OnlyDomains" setting, that can be set to a list of domains known to have problems[7].

The long-term solution seems straightforward. As we have seen, the number of name servers exhibiting this problem is relatively small, though some do serve some often-queried domains. If we can ensure that no more servers with these problems get deployed, then as the existing servers are updated or retired the problem will be resolved.

To this end, it is worth testing new DNS deployments to make sure that they correctly respond to unusual query types[8]. This will smooth the path not just for IPv6, but also for other new technology such the *Domain Name System Security Extension* (DNSSEC)[9].

References

[1] "Observations of 6to4 Traffic on a 6to4 Router," Pekka Savola, preprint, October 2004.

[2] "Connecting IPv6 Routing Domains Over the IPv4 Internet," Carpenter, Moore, and Fink, *The Internet Protocol Journal,* Volume 3, No. 1, March 2000.

[3] "Common Misbehavior against DNS Queries for IPv6 Addresses," Y. Morishita and T. Jinmei, **draft-ietf-dnsop-misbehavior-against-aaaa-01.txt**, April 2004.

[4] K-Root information page, RIPE, **http://k.root-servers.org/**

[5] "**news.bbc.co.uk** NXDOMAIN problem fixed," itojun, Simon Lockhart, et al., 6bone mailing list, April 2002.

[6] "**ftp.perl.org** strangenes" thread, Mark Andrews, Ask Bjoern Hansen, et al., freebsd-stable mailing list, March 2004.

[7] "IPv6: Some IPv4 addresses won't resolve w/IPv6 OS," Mozilla bug 68796,
https://bugzilla.mozilla.org/show_bug.cgi?id=68796

[8] "AAAA lookup checker," David Malone,
http://www.cnri.dit.ie/cgi-bin/check_aaaa.pl

[9] "DNSSEC," Miek Gieben, *The Internet Protocol Journal,* Volume 7, No. 2, June 2004.

DAVID MALONE received B.A. (mod), M.Sc., and Ph.D. degrees from Trinity College Dublin. He has been involved with system administration since 1994 and has been slowly growing IPv6 networks since 1999, when he also became a FreeBSD committer. With Niall Murphy, he is the coatuthor of *IPv6 Network Administration,* ISBN 0-596-00934-8 published by O'Reilly and Associates, 2005. He is currently on secondment to the Hamilton Institute of NUI Maynooth. E-mail: **dwmalone@maths.tcd.ie**

Wireless Data Networks

by Kostas Pentikousis, VTT

Most IPJ readers are familiar with *Wireless Local-Area Networks* (WLANs; see, for example, IPJ Volume 5, No. 1). Some may even be familiar with recent developments in *Wireless Metropolitan-Area Networks* (WMANs), such as WiMAX. Although nonproprietary WMAN technologies are still in the standardization phase, the IEEE 802.11 family of protocols has reached maturity and rendered inexpensive (and often free) WLAN access increasingly popular. Both WLANs and WMANs provide high-speed connectivity (in the order of tens of Mbps), but user mobility is restricted. In fact, it is probably more appropriate to talk about "portability" rather than "mobility"[1] when referring to WLANs and WMANs.

Wireless wide-area networks (WWANs), on the other hand, allow full user mobility but at data rates typically in the order of tens of kbps. This will change to some extent when *third-generation* (3G) cellular networks are fully deployed. Still, 3G deployment is slower than originally anticipated, a development often attributed to the combination of high spectrum license costs, the recent economic downturn, and high equipment costs. As a result, both population and geographical coverage tend to be uneven. For example, in Finland, a forerunner in wireless communications, population coverage is well below the 35-percent level, and geographical coverage is even smaller

This article introduces several wireless network technologies, perhaps not so widely known, which deserve attention when considering how to provide mobile connectivity to field personnel, introduce *machine-to-machine* (M2M) communication, or deploy applications that require always-on connectivity. The approach taken in this article is a bit different from the one typically followed in the literature: We focus more on higher-level issues, the information that is essential for application developers, instead of modulation, channel coding, and other low-level details. Unlike WLANs and WMANs, none of the networks surveyed provide data rates in the order of tens of Mbps. Nevertheless, successful applications can be built even with stringent bandwidth limitations. For example, online gambling and several gaming applications can be served by really "thin" networks (and possibly "thick" clients).

Cellular Networks

The *Global System for Mobile Communications* (GSM) specifies a cellular, wide-area, circuit-switched, digital mobile phone network architecture[2]. Circuit-switched networks such as GSM and IS-95, commonly referred to as *Code Division Multiple Access* (CDMA) in the United States, can provide wireless data connectivity, cover a large area, and handle mobile host handovers efficiently[3]. Users can transfer data over, say, GSM, by establishing a "dialup" connection[4]. Mobile hosts can roam, even at high speeds, and remain connected throughout.

Communication is full-duplex at a radio data rate of 9.6 kbps or 14.4 kbps in GSM Phase 2+[5]. User throughput is always smaller than the nominal radio data rate.

While the user is connected using a wireless circuit-switched network, phone calls cannot be initiated or received whether data is being transferred or not. This is not much different from wire-line dialups over basic telephone service. The difference is that a dialup over a *Public Switched Telephone Network* (PSTN) takes up a resource, namely the wire-line local loop, which is dedicated to a single user, whereas a dialup over a cellular network such as GSM consumes a resource, the radio channel, which is shared among many users. Because of the *burstiness* that data traffic usually exhibits, circuit switching may lead to inefficient use of the network capacity. Establishing a GSM dialup connection usually takes several seconds, meaning that if the user has a small amount of data to send, a small e-mail message, for example, the overall experience is poor. Moreover, after the connection is established, the channel remains idle between traffic bursts and the allocated bandwidth is wasted. Packet switching is more efficient for bursty data transmission over a shared medium[6].

Another variable that favors packet-switching over circuit-switching, especially over slow wireless networks, is *billing*. Users of circuit-switched networks are usually charged based on the duration of a connection regardless of the amount of traffic transmitted or received. On the other hand, users of packet-switched networks can be charged based solely on the amount of data transferred—not how long they remain attached to the network. In short, introducing packet switching to wireless networks can lead to better use of network resources and attract more users as data transfers become more economical.

Two-way, packet-switched WWANs permit users to roam freely indoors and outdoors, even at relatively high speeds[7]. Most WWANs employ a cellular architecture to take advantage of frequency reuse and increase capacity while covering a larger area. Furthermore, because the coverage area of a single cell is generally large (cell diameters are typically in the order of dozens of kilometers), mobile hosts do not have to go through frequent and lengthy handovers. Hosts remain connected throughout after they attach to the network, permitting users to receive and transmit data on demand without having to dial up. The following sections survey some of the most widely deployed packet-switched wireless data networks.

Mobitex

Mobitex is the first digital data-only WWAN developed by Ericsson and Swedish Telecom. Not based on IP, Mobitex was introduced in Sweden in 1986 for emergency communications[8]. It uses a cellular architecture with cell diameters of up to 30 km. Each service area can operate 10–30 channels[9] and each base station is usually allocated 1 to 4 channels. Each channel is composed of a frequency pair: different frequencies are used for the uplink and the downlink.

Communication between the base station and a single mobile host is, nevertheless, effectively half-duplex. Although base stations can transmit and receive simultaneously, mobile nodes are unable to do so[10]. The Mobitex *Maximum Transmission Unit* (MTU) is 545 bytes, with up to 512 bytes of user data. Although the system has undergone several revisions, the raw transfer rate remains only 8 kbps. Effective user throughputs range from 4 kbps (for 125-byte packets) to 4.6 kbps (for 512-byte packets)[11], and round-trip times can be up to 10 seconds.

Mobitex deals with network lapses using a store-and-forward procedure: Packets destined for a mobile node outside the network coverage area are stored while awaiting delivery. When the mobile node reconnects, the stored packets are delivered. Mobitex uses a hierarchical routing architecture that prevents local traffic from being injected into the backbone network. In other words, packets destined for a node in the range of the same base station are switched locally[8]. Besides supporting unicast addressing, Mobitex allows hosts to send one packet to several recipients[10]. According to the *Mobitex Association* (**`www.mobitex.org`**), the technology features "true push functionality," whereby data can be pushed to both a single mobile node and a predefined group of nodes, a feature that can be very useful when trying to send an urgent message to field personnel. And, because the mobile host does not have to keep querying for pending data, network traffic can be kept to a minimum. All these features can also significantly boost battery life.

According to the Yankee Group, despite the limited data rates, a variety of applications have been developed based on Mobitex, including: burglar and fire alarm systems; paging, interactive messaging, e-mail, form-based applications, and access to databases; telemetry; credit card authorizations; field service; and fleet management. Virtually all of them require small and bursty transfers. Mobitex does not lend itself to large file transfers, e-mail with large attachments, or video transmission. In fact, file transfers of more than 20 KB used to be discouraged[8]. On the other hand, by using a slotted ALOHA[12] variation for channel access, Mobitex can provide message delivery delay guarantees and support hundreds of users within the same cell. Parsa[13] calculated that Mobitex can accommodate 2,000 users per channel, assuming two uplink and two downlink messages per hour. Other networks simply cannot provide tight delay bounds for such a large number of users. For example, the *Mobile Data Magazine* (No. 1, 2002) reported that a Korean operator launched real-time stock trading and horse gambling mobile applications with great commercial success, by guaranteeing delay bounds notwithstanding the low data rates.

DataTAC

DataTAC (also known as ARDIS in the United States) was developed by Motorola in the mid-1980s. DataTAC is also a non-IP based, wide-area, data-only message-oriented network. A single base station can cover an area exceeding 20 km in diameter[14]. Like Mobitex, communication between the base station and a single DataTAC mobile node is half-duplex, and mobile hosts have to compete to get access to transmit and receive data.

Unlike Mobitex, DataTAC was designed to provide optimal in-building coverage, and it uses a cellular architecture that does not take advantage of frequency reuse. Instead, a single frequency is used, increasing the probability that a packet transmission is successful (because the same transmission can be picked up by more than one base station), but at the expense of network capacity[8]. Bodsky notes that the U.S. DataTAC operator formerly recommended refraining from transferring files larger than 10 KB.

Although neither Mobitex nor DataTAC provides native IP support, middleware can take care of protocol translation and allow unmodified, off-the-shelf applications to communicate. The maximum Data-TAC message size is 2048 bytes[15], but the maximum over-the-air packet size depends on the link layer. For rural areas the maximum radio data rate is 4.8 kbps, and the maximum over-the-air packet size is 256 bytes. In metropolitan areas, the radio data rate is 19.2 kbps and the maximum packet size is 512 bytes[16]; end-user throughput does not exceed 10 kbps on average. Traditionally, DataTAC was used for dispatching and law enforcement applications. The *Worldwide Wireless Data Network Operators Group* (**www.datatac.com**) reports that DataTAC networks are also used for two-way messaging, wireless e-mail, telemetry, access to corporate databases, and package tracking by courier carriers.

CDPD

Cellular Digital Packet Data (CDPD) was designed by IBM and McCaw Cellular Communications in the early 1990s to take advantage of channels that do not carry voice traffic in the *Advanced Mobile Phone Service* (AMPS), the first-generation analog cellular network[17]. Data channels are allocated dynamically, sharing the network capacity with AMPS voice traffic, which is quite different from Mobitex and DataTAC. This, for example, might mean that data can be transmitted and received only when phone calls do not consume all available capacity. One could argue that CDPD considers data traffic less important than voice. However, the standard allows network operators to specifically assign channels to data traffic only. In theory, deployment can be more economical than it is for other WWANs because CDPD takes advantage of existing AMPS infrastructure and does not require licensing new spectrum. Original projections anticipated that as CDPD gained popularity—and AMPS became obsolete—more CDPD dedicated channels would be allocated. With time, CDPD would have taken over the existing AMPS bandwidth, effectively becoming a data-only WWAN.

CDPD is based on a *Carrier Sense Multiple Access* (CSMA) variant called *Digital Sense Multiple Access*[14] and transparently provides IP services, constituting a great advantage. CDPD allows for an MTU of 2048 bytes. However, one has to account for the TCP/*User Datagram Protocol* (UDP) and IP headers that are used to encapsulate the application payload before sending it over the CDPD network and also for the fact that CDPD user data is transmitted in much smaller blocks. Although the CDPD raw data rate is 19.2 kbps, the effective throughput is in the order of 10 kbps and response times have been reported to be in the order of 4 seconds[18].

GPRS

The *General Packet Radio Service* (GPRS) is overlaid on a GSM network in a fashion similar to the way CDPD is embedded in AMPS: Voice and data traffic share the same bandwidth and network infrastructure[14]. In other words, GPRS is an add-on to GSM networks, and it requires certain hardware and software upgrades and introduces packet switching to a circuit-switched architecture. GSM voice traffic is oblivious to the presence of GPRS data traffic. Similar to CDPD, GPRS is designed to appear as a regular IP subnetwork both to hosts attached over the air interface and to hosts outside the GPRS network.

The GPRS standard was finalized by the *European Telecommunications Standards Institute* (ETSI) in late 1997 as part of GSM Phase 2+[5]. It is regarded as a transitional technology toward 3G networks[19], and is commonly referred to as 2.5G. One of its main advantages is that the same device can be used to transmit and receive data, and initiate and accept phone calls. GPRS defines three classes with respect to simultaneous usage of voice and data. Class A mobile hosts can transmit and receive voice and data at the same time. Class B hosts can transmit and receive either voice or data but not both simultaneously. Finally, class C hosts have the user manually select if the host should be attached to the GSM (voice) or GPRS (data) network. When compared to Mobitex, DataTAC, and CDPD, GRPS class A devices can have simultaneous access to a packet-switched and circuit-switched network. Of course, GSM-only devices do not have this capability either, as mentioned earlier.

GSM uses a combination of *Frequency Division Multiple Access* (FDMA) and *Time Division Multiple Access* (TDMA) for channel allocation, as explained in detail in[5]. In short, each frequency channel carries eight TDMA channels. Each of these channels is essentially a time slot in a TDMA frame. Thus, any GSM frequency channel can carry up to eight circuit-switched connections with each slot reserved for a single connection (read *voice call*). In GPRS, each slot is treated as a shared resource and any mobile host can use it to transmit or receive data. In addition, a mobile host can be allocated more than one of the eight available slots in the same TDMA frame. In other words, GPRS can multiplex different traffic sources in one channel and allocate several channels to the same traffic source.

GPRS defines four different channel coding schemes[20], namely CS1, CS2, CS3, and CS4, with radio data rates 8.8 kbps, 13.3 kbps, 15.6 kbps, and 21.4 kbps, respectively. CS1 is the most "conservative" (includes more error correction bits) and is used for signaling packets and when poor channel conditions prevail. CS4 is the most "optimistic" (includes minimal error correction bits), and, assuming excellent channel conditions, allows operators to advertise a maximum radio data rate of 171.2 kbps per 200-kHz frequency channel (or TDMA frame).

In practice, CS4 is rarely used because it can lead to frequent retransmissions of lost packets and overall network underperformance. CS3 is commonly used, providing 124.8 kbps per frequency channel. Because a mobile host can be allocated multiple slots, user throughputs can range between 40 and 60 kbps. Mobile hosts typically use an MTU of 1500 bytes.

Communication between the base station and any given mobile host is full-duplex but can be *asymmetric;* that is, the downlink and uplink capacities need not be the same. The *GSM Association* has defined 12 multislot classes for GPRS. Each class is associated with a maximum number of uplink and downlink slots that can be allocated to a single mobile host. The slot allocation is usually written as *M* + *N*, where *M* is the maximum number of downlink slots and *N* is the maximum number of uplink slots. For example, class 1 is "1 + 1" (one downlink slot plus one uplink slot); class 2 is "2 + 1"; . . . ; and class 12 is "4 + 4" (four downlink and four uplink slots). In addition, each multislot class has an active slot constraint: A mobile host cannot use more than *K* active slots simultaneously. Given the number of slots and the channel coding scheme, one can calculate the peak rate. For example, for a class 12 device the sum of the physical downlink and uplink rates cannot exceed 124.8 kbps, if CS3 is used. However, the active slot constraint limits this rate even further. In the case of a class 12 mobile node, *K* = 5, that is, only "4 + 1", "3 + 2", "2 + 3", or "1 + 4" slots can be used simultaneously. See **www.gsmworld.com**

EDGE and Beyond

Enhanced Data for GSM Evolution (EDGE), also known as Enhanced GPRS, builds on the changes introduced by GPRS to GSM. EDGE essentially increases the radio data rates by using a more efficient modulation scheme[21], namely *8-Phase Shift Keying* (8-PSK) instead of the *Gaussian Minimum Shift Keying* (GMSK) used by both GSM and GPRS. EDGE defines nine modulation coding schemes named MCS1 to MCS9. MCS1 to MCS4 use GMSK with radio data rates similar to the four GPRS coding schemes. The real throughput improvements come from MCS6 (29.6 kbps per slot) through MCS9 (59.2 kbps per slot). The data rate usually associated with EDGE is a (shared) 384 kbps. This corresponds to using MCS7 for all 8 TDMA slots. Higher data rates are theoretically possible (up to 473 kbps using MCS9) but are not commonly deployed.

EDGE improves not only on the high end of data rates but also on the low end[22]. First, the greater diversity of coding schemes permits an EDGE network to choose the most appropriate one depending on channel conditions. Changing coding schemes is dynamic. Second, EDGE supports *packet resegmentation:* Packets that failed to be transmitted successfully can be resegmented and retransmitted using a more "conservative" coding scheme.

Table 1 summarizes the main high-level features for the WWANs surveyed.

Table 1: WWAN Characteristics

	Transmit/ Receive	Radio Data Rate	User Throughput	MTU
Mobitex	*Half duplex*	*8.0 kbps*	*<4.6 kbps*	*512 B*
DataTAC	*Half duplex*	*19.2 kbps*	*<10 kbps*	*2048 B**
CDPD	*Full duplex*	*19.2 kbps*	*<10 kbps*	*2048 B*
GPRS	*Full duplex*	*<171 kbps*	*40–60 kbps*	*1500 B*
EDGE	*Full duplex*	*<473 kbps*	*50–60 kbps*	*1500 B*

* Typically 512 B

Discussion and Trends

Among the WWANs presented, Mobitex and GPRS can be singled out as the most widely deployed; they also have enjoyed significant gains in the number of users and traffic volume in recent years. The popularity of enterprise wireless e-mail (due in part to the success of the Research in Motion BlackBerry devices) allowed Mobitex and DataTAC operators to revive their business models briefly. Worldwide, however, GSM dwarfs all other technologies: There are more than 1 billion GSM subscribers compared to the 1 million Mobitex users. DataTAC enjoys an even smaller user base. Even if a small percentage of GSM subscribers use GPRS and EDGE, the potential market for wireless applications is tremendous. On the other hand, subscribers who do not take advantage of GPRS or EDGE do use the inexpensive, (two-way) *Short Message Service* (SMS), which is built in GSM. Two-way messaging was available for many years but was certainly popularized by less-affluent and younger GSM users in the late 1990s. SMS is now commonplace, and in many countries it is more popular than e-mail. Dedicated data-only networks such as Mobitex have to look elsewhere for their niche.

For some, Mobitex, let alone DataTAC and CDPD, is virtually moribund. In the United States, for example, Cingular sold its Mobitex network and is investing heavily on GRPS and EDGE. DataTAC and CDPD are phased out by service providers in the United States in favor of newer technologies. Low-speed packet radio is considered lackluster and is not popular with younger crowds. After all, narrowband WWANs had their chance and failed to attract large numbers of subscribers. Recent pricing trends, too, reveal a heavy operator push in favor of GRPS and EDGE. In Finland, for example, 100 MB over GPRS costs less than 18 euros (approximately $24). Compare that to the $30–50 that 1 MB of traffic costs over Mobitex. Service and product popularity create economies of scale that cannot be ignored.

Nonetheless, open standards, an explicit focus on business applications with *Quality-of-Service* (QoS) guarantees in service response times, and narrowband M2M communication may well keep Mobitex going for years to come. Besides, bundling Mobitex with a wireless network that features fast and inexpensive connectivity, for example, WLAN or Bluetooth, might be promising: Large downloads and software updates can be done over the high-speed wireless network and critical messages can always reach the user through the WWAN.

Bundling several functions in a single handheld device is, after all, a major trend in the industry. Vendors scramble to integrate *Personal Information Managers* (PIMs), voice and data communications, as well as entertainment features (digital camera, games, or digital music players) in a single product. This is quite different from earlier mobile devices, which tended to be either single-purpose or tied to a particular set of applications. Even the BlackBerry devices still work, to some extent, in a closed architecture. Enterprise e-mail systems need to be supported by and integrated with BlackBerry servers in order to be accessible over the WWAN. Yet, one of the main objectives in 2.5G and 3G is to allow mobile users to use standard Internet protocols on a mobile radio network at significantly higher bit rates than other systems. In particular, GPRS was designed with certain office applications in mind and can support consumer and enterprise mobile communications alike, without being tied to any given platform or application servers. I expect that functionality bundling and 2.5G and 3G WWANs will allow for more open systems and will expedite the transformation of WWAN operators from integrated application providers to wireless ISPs.

For Further Reading

[1] Charles Perkins, *Mobile IP Design Principles and Practices,* ISBN 0201634694, Addison-Wesley, 1998.

[2] Joachim Tisal, *GSM Cellular Radio Telephony,* ISBN 0471968269, John Wiley & Sons, 1998.

[3] Tero Ojanpera and Ramjee Prasad, *WCDMA: Towards IP Mobility and Mobile Internet,* ISBN B000066OB4, Artech House, 2001.

[4] R. Ludwig, B. Rathonyi, A. Konrad, K. Oden, and A. Joseph, "Multi-layer tracing of TCP over a Reliable Wireless Link," presented at ACM SIGMETRICS 1999.

[5] C. Bettstetter, H.-J. Vogel, and J. Eberspacher, "GSM phase 2+ General Packet Radio Service GPRS: Architecture, Protocols, and Air Interface," *IEEE Communications Surveys & Tutorials,* Vol. 2(3), pp. 2–14, 1999.

[6] Larry L. Peterson and Bruce S. Davie, *Computer Networks: A Systems Approach,* 3rd ed., ISBN 155860832X, Morgan-Kaufmann, 2003.

[7] Rudi Bekkers and Jan Smits, *Mobile Telecommunications: Standards, Regulation, and Applications,* ISBN 0890068062, Artech House, 1999.

[8] Ira Brodsky, *Wireless: The Revolution in Personal Telecommunications,* ISBN 089006717, Artech House, 1995.

[9] Nathan J. Muller, *Wireless Data Networking,* ISBN 0890067538, Artech House, 1995.

[10] A. K. Salkintzis and C. Chamzas, "Mobile Packet Data Technology: An insight into Mobitex Architecture," *IEEE Personal Communications,* Vol. 4(1), pp. 10–18, 1997.

[11] M. S. Taylor, M. Banan, W. Waung, and M. Taylor, *Internetwork Mobility: The CDPD Approach,* ISBN 0132096935, Prentice-Hall, 1996.

[12] Andrew S. Tanenbaum, *Computer Networks, 4th ed.,* ISBN 0130661023, Pearson Education, 2003.

[13] K. Parsa, "The Mobitex packet-switched Radio Data System," presented at IEEE PIMRC '92, 1992.

[14] Sami Tabbane, *Handbook of Mobile Radio Networks,* ISBN 1580530095, Artech House, 2000.

[15] J. Rodriguez, W. Schollenberger, M. Anzib, and B. Widyarso, *Mobile Computing: The eNetwork Wireless Solution,* ISBN 0738412856, IBM Redbooks, 1999.

[16] Research in Motion, "Developer's guide for BlackBerry and RIM Wireless Handhelds—Radio API (DataTAC) Version 2.1," 2001.

[17] John Agosta and Travis Russel, *CDPD: Cellular Digital Packet Data Standards and Technology,* ISBN 0070006008, McGraw-Hill, 1996.

[18] P. Sinha, N. Venkitaraman, T. Nandagopal, R. Sivakumar, and V. Bharghavan, "A Wireless Transmission Control Protocol for CDPD," presented at IEEE WCNC '99, 1999.

[19] A. K. Salkintzis, "A survey of mobile data networks," *IEEE Communications Surveys & Tutorials,* Vol. 2(3), pp. 2–18, 1999.

[20] L. F. Chang, "Wireless Internet—Networking Aspect," in *Wireless Communication Technologies,* New Multimedia Systems, N. Morinaga, R. Kuhno, and S. Sampei, Eds., Kluwer Academic Publishers, 2000, pp. 215–244.

[21] Behrouz A. Forouzan, *Data Communications and Networking, 2nd ed.* Update, ISBN 0072822945, McGraw-Hill, 2002.

[22] Alexander J. Huber and Josef F. Huber, *UMTS and Mobile Computing,* ISBN B000089CJ3, Artech House, 2002.

KOSTAS PENTIKOUSIS, PhD, studied computer science at Aristotle University of Thessaloniki and Stony Brook University. He is an ERCIM Fellow at VTT, The Technical Research Center of Finland, and currently resides in Oulu, Finland. For more about his research and publications visit: **www.cs.stonybrook.edu/~kostas**. The best way to reach him is via skype. E-mail: **kostas@cs.sunysb.edu**

Opinion: ICANN, the ITU, WSIS, and Internet Governance

by Geoff Huston, APNIC

This is an opinion piece, intended primarily to provoke thought and comment. The author does not claim to personally hold any of the opinions expressed in this article.

It may have taken some three decades to get here, but there is no doubt that the Internet is now a major public communications utility. That is hardly the most important piece of news you are likely to read today, but the implication of this public role is that there are legitimate issues of public policy to consider when looking at the broad topic of coordination of various aspects of Internet infrastructure. In other words, "Internet Governance" is a matter of significant concern to many.

This opinion piece looks at the various range of views about the *Internet Corporation for Assigned Names and Numbers* (ICANN)[1] and its rationale and role over its brief history. Of course, no look at Internet Governance would be complete without also looking at the role of the *International Telecommunications Union* (ITU), as well as the broader background to this topic. It is a large topic and it has already been the catalyst for numerous articles.

Data Networking and Public Networks

Whether it was because of its antecedents in the research community, or simply because it was not originally envisaged that the Internet would become a global communications platform in its own right, or for whatever reasons, the administration of the Internet infrastructure was not originally crafted with conventional public network coordination in mind. The retrofitting of a model that incorporates considerations of a public utility role is proving to be a rather complicated process.

For example, the original hierarchical name space for the Internet used a set of generic top-level root zone names of "**edu**," "**net**," "**com**," "**gov**," and "**mil**." Adding country codes to the root of the name space was a later modification. Even then the original country code delegations were undertaken to individuals or entities who appeared to have some form of link to the national Internet community, rather than specifically seeking out an appropriate office of the national administration of communications services as the point of delegation. Similarly, IP addresses were structured without any form of national prefix, nor were IP addresses distributed along any national lines. In these respects the Internet was really no different from any other computing networking protocols of the 1980s, such as *DECnet,* the *Xerox Network System* (XNS), *AppleTalk,* or IBM's *Systems Network Architecture* (SNA), where names and addresses were defined in a limited context of the scope of the network, rather than within some broader public name framework.

There were two notable exceptions to this characterization of computer network protocols, and both were designed with a public communications utility as their primary objective, namely *X.25* and the *Open Systems Interconnection* (OSI) model. They can be regarded as offerings from the data services sector of the established telephone industry. X.25, the earlier of these two protocols, had a very obvious relationship to telephony, complete with the notion of a "call" as the means of establishing a data connection and as the unit of a transaction. The addressing scheme used a structured space that drew heavily on the telephone number structure. Like telephony, there was no associated name scheme and endpoints were identified by their numeric X.25 protocol address. OSI represented a later effort to design a packet-switched network architecture that was intended to reflect an increasing level of experience with this technology, but nevertheless continued to draw heavily on telephony design. Much was written about OSI at the time, and it would be a diversion to explore it in depth here. However, the salient observation here is that despite the extensive effort invested into its promotion, OSI was a market failure, and whatever its technical merits it was simply not accepted by the communications industry.

OSI was heavily supported by the ITU, and by virtue of this very active sponsorship of this technology, the implication of the aftermath of OSI was that the ITU was seen as being simply out of touch with data networking. It was often portrayed that the ITU was coming from a mindset that was incapable of engaging with either the data communications industry or the broader consumer market for data services. From the perspective of data networking, the failure of OSI was seen as a failure of the ITU itself.

The ITU and the Internet

The ITU is certainly one of the more venerable institutions in the communications sector. It can trace its origins to May 1865, when the first *International Telegraph Convention* was signed by 20 founding national members, and the *International Telegraph Union* was established to facilitate subsequent amendments to this initial agreement. Two decades later, in 1885, the ITU drafted international legislation governing telephony. With the invention in 1896 of wireless telegraphy, similar coordinating measures were adopted by the *International Radiotelegraph Convention*. In 1932 the Union combined the International Telegraph Convention of 1865 and the International Radiotelegraph Convention of 1906 to form the *International Telecommunication Convention*. The name of the body was changed to *International Telecommunication Union* to properly reflect the full scope of the Union's responsibilities, which by this time covered all forms of wireline and wireless communication.

In 1947 the ITU, under an agreement with the newly created United Nations, became an agency of the United Nations, with responsibilities in international telephony, telegraphy, and radio communications. Over the next four decades the ITU oversaw a system of international interconnection of telephony and data systems that became an industry in and of itself.

The ITU assumed a role of facilitating what was asserted to be a balanced international environment where the costs of running the international system were fairly apportioned between national service providers. In practice these lofty goals were not achieved very efficiently, and international facilities were priced at levels that were considerably higher than the associated costs of actual service provision. When attempts were made to redress the imbalances between large and small national carriers, the outcomes included collective action on the part of the national carriers that operated in ways not dissimilar to a cartel.

In 1992 the ITU was restructured into three sectors, corresponding to its three main areas of activity, namely the standardization of telecommunications technologies in the ITU-T, the coordination of radiocommunications in the ITU-R, and telecommunication development in the ITU-D. In 1994 the ITU established the *World Telecommunication Policy Forum* (WTPF), a group that encouraged the exchange of ideas and information about emerging policy issues arising from the changing telecommunication environment. The first WTPF was held in 1996 on the theme of global mobile personal communications by satellite, and the second in 1998, on trade in telecommunication services.

The ITU was heavily criticized over the ponderous amount of time taken to generate telecommunications standards, the nature of the process used in developing these standards in a closed set of forums, the marginal relevance of these standards, and the final indignity, that the ITU charged for paper and electronic copies of these standards. As some critics pointed out, perhaps harshly, this was not just a case of paperware about vapourware, it was a case of very expensive paperware about vapourware!

More recently, the ITU has focused on attempting to strengthen the participation of the private sector in the work of the Union, as well as streamlining the ITU's processes to reduce the level of delay and amount of process overhead in standardization of technology and operational practices. The ITU has sponsored the establishment of the *World Summit on the Information Society* (WSIS)[2], and has been attempting to position itself more centrally in the process of further evolution of the Internet as part of its overall charter.

The Internet has posed a severe challenge to the ITU. Not only was the ITU often perceived as being out of touch with the data communications sector, more critically it had been perceived as being incapable of making the necessary reforms to its mode of operation and policy setting to bring it back into relevance for the rapidly changing communications industry. The inference was being drawn that the ITU was apparently in a state of denial over progressive deregulation of national communications sectors. In many cases the national position had already moved to a position of lightweight regulation, relying on strong competitive pressures in the private sector to enforce regimes of efficiency and effectiveness in the supply of communications services to consumers. The ITU, as an intergovernmental organization, was being seen in some quarters as an anachronistic recalcitrant relic of an earlier era of communications service provision.

It was also evident that this critical view of the ITU was most strongly held within the United States, and in particular those parts of the U.S. administration and industry that were involved with the growth of the Internet. It was perhaps no coincidence that in these growth industries of personal computer technologies and the related Internet industry it was U.S. enterprises that were the "poster children" of this new model of industry-led deregulated communications services. Their consequent rapid expansion into a massive global undertaking of the global Internet was perhaps the most eloquent form of statement about the effectiveness of deregulation, and the degree to which the previous regulatory model had simply not managed to encompass the burgeoning demand for data services in a timely fashion.

From this perspective it should be no surprise to observe that when the transition of the *Internet Assigned Numbers Authority* (IANA) function from a fully federally funded research activity to some form of new foundational base was being considered by the U.S. administration, it appears that the ITU was never seriously contemplated as a viable home for this function. If the Internet was a child of deregulation and industry initiative taking on the outcomes of research activity, then the appropriate progression of the IANA function was also from a research context into an enterprise context. IANA should be responsive to industry needs, and to best achieve this the IANA function itself should be undertaken as a task housed within the deregulated private enterprise sector, rather than establishing yet another public bureaucracy, or using existing bureaucracies for the role. ICANN was the embodiment of this aspiration on the part of the U.S. administration, and to pass the effective levers of control of the Internet to the ITU was seen as denying the Internet any form of a productive, innovative, and successful future.

The Formation of ICANN

Whatever the original motivation in creating ICANN to administer the IANA responsibilities, it is now apparent that ICANN was deliberately structured to provide the industry with an alternative structure of coordination and regulation within national and international communications sectors to that of the ITU. The critical difference is that ICANN had not placed governments at the forefront of visible activity, but instead placed industry needs and the operation of a competitive deregulated international communications sector as being the major thrust of coordination activities.

As with any novel model of public policy determination, ICANN's acceptance ranged from cautious approval to advanced skepticism. Even within the U.S. administration ICANN has yet to be "unleashed," and it currently operates under the terms of a Cooperative Agreement with the *National Telecommunications and Information Administration* of the U.S. Department of Commerce under a sole source cooperative agreement. In this light ICANN appears to be a cautious step in a bold direction.

ICANN undertakes activities of management of Internet Protocol infrastructure in the areas of the content of the root of the *Domain Name System* (DNS) and the identification of parties to whom are delegated administrative and operational control of the top-level domains and the associated specification of terms and conditions of this delegation. ICANN, through IANA, also manages the pool of unallocated IP addresses (IPv4 and IPv6 addresses and Autonomous System numbers), and also manages the protocol parameter registries as defined by IETF Standards Actions.

ICANN Mkl

The initial structure of ICANN had three "supporting organizations," focusing on:

- Coordination of the DNS with the *Names Supporting Organization* (NSO)
- Coordination of address policies with the *Address Supporting Organization* (ASO)
- Operation of Internet Protocol parameter registries with the assistance of the *Protocol Supporting Organization* (PSO)

The intended role of these supporting organizations was to provide a venue where interested parties could develop and consider policy proposals, leaving the task of ultimate identification of broad support for particular policy initiatives to the ICANN Board.

As has been evident to any observer of the ICANN process, things did not proceed within the parameters of that plan. The NSO met problems due to the diversity of interests that were encompassed with the DNS domain, including emerging national and regional interests in the country code top-level domains, the operators of the generic top-level domains, the trademark and intellectual property collection of interests, the emerging industry of registrars, and a continual interest of individuals who maintained that they had legitimacy of inclusion by virtue of their representation of interests of end users and consumers, or, to use an emerging ICANN lexicon, the "at large" constituency.

The ASO was formed within the parameters of a different model. The *Regional Internet Registries* (RIRs) had already developed a considerable history of working within their communities, and being widely accepted by these communities as an appropriate means of coordination of activity in the role of number resource administration and distribution. The ASO was formed with membership of the associated council based on processes determined by each RIR. Even then it was unclear as to the relationship between the RIRs' already well-established open policy development process and the ASO and ICANN. The RIRs were unwilling to pass all regionally developed policies to ICANN for a second round of consideration and potential alteration. They insisted that only those policies that were considered to be "global," in that they were common to all the RIRs, would be passed into this ICANN sphere.

The PSO was placed under strong pressure to include the ITU-T and the *European Telecommunications Standards Institute* (ETSI), and the *World Wide Web Consortium* (W3C) was also enlisted, in addition to the IETF. If the objective of the PSO was oversight and policy formulation concerning the role of protocol parameter registration of IETF protocols, then this enlarged membership of the PSO was unwarranted. Even within the terms of consideration of the PSO as a source of standards-based technical advice to the ICANN Board, the presence of these additional organizations was somewhat puzzling in terms of the match of resultant structure of the PSO to its intended role. The PSO, however, had a role in seating individuals onto the board of ICANN, and it was likely that this aspect of the PSO had been part of the reason for the interest in broader institutional membership. Uncertainty about the extent of the role of ICANN saw many groups attempting to gain access to board seats.

Missing from this mosaic of diverse interests was the inclusion of various national public communications sector entities who also felt that they had clear legitimacy to undertake an active role within the ICANN policy development process, and, in response, the *Government Advisory Committee* (GAC) was formed.

ICANN Evolution and Reform

If a camel is a horse designed by a committee, then it is unclear whether ICANN was a three-humped camel or a three- and three-quarter-humped camel as a result of all this, but camel it undoubtedly was.

The PSO was dysfunctional and missing any tangible agenda of activity. A fracture was apparent in the relationship between ICANN and the IETF. Attempts to create an agreement between ICANN and the IETF over the IANA function were not recognized by the U.S. administration, who continued to insist that, formally, the IANA function for the IETF was undertaken at the behest of the U.S. Department of Commerce rather than the IETF. This view was not shared by the IETF.

The ASO was criticized by ICANN itself of being insufficiently "representative" of the addressing community, and the ICANN Board established its own temporary advisory committee on addresses, and in so doing alienated the RIR community from the entire ICANN framework.

The NSO was hopelessly wedged into factional-based politics.

The GAC decided at the outset that it would operate behind closed doors, in contrast to ICANN's continuing efforts to operate in an open and transparent manner.

The "At Large" election process undertaken by ICANN appeared to be of dubious validity because of problems in establishing a reliable constituency of individuals who had an interest in ICANN, and a direct election process was attempted only once.

Not surprisingly, ICANN fell into some disarray under these pressures, and by early 2002 the CEO of ICANN at the time, Stuart Lynn[3], was warning all who cared to listen that ICANN was paralyzed, dysfunctional, and in danger of an imminent demise. Whether this was a message directed to the ICANN Board or to a fractious set of communities that had some intersection with ICANN, or to the U.S. administration who had been influential in determining the original ICANN structure was not entirely clear to any observer of the process.

However, given that ICANN had been set up as an example of a new form of international coordination of communication infrastructure support activities that was based on private-sector activity rather than governmental fiat, this message of imminent failure was widely interpreted both as a potential failure of ICANN and a sign of failure of this new model of coordination of international activity. ICANN was seen as a point of vulnerability with respect to the U.S. administration's diplomatic efforts to reform this international activity sector. The ITU-T's activities in this same area was reinvigorated, with considerable support from national sectors who saw their national interests being potentially advantaged in a ITU-led international environment.

ICANN MkII

Although still firmly positioned as a private-sector activity, and although still making no concessions in the direction of the ITU, ICANN has managed to reorganize its structure through a protracted evolution and reform process.

With respect to the ASO, The Regional Internet Registries formed its own coordination entity, the *Number Resource Organization* (NRO)[4], and has proposed this entity to ICANN as the means of interfacing between the addressing community and ICANN's policy-development activities.

The PSO was abolished, to be replaced by a *Technical Liaison Group* that, apart from its function of seating an individual on the ICANN Board, is a group without an obvious role or agenda.

The NSO was forced to recognize the fundamental difference between the generic top-level domains, which fall under a more direct relationship with ICANN and its processes, and the country code domains (ccTLDs), which have from the outset been quite wary of ICANN. From the ICANN reform process emerged the *Country Code Name Supporting Organization* (CCNSO) and the *Generic Names Supporting Organization* (GNSO), as a recognition that these two groupings are so dissimilar that they have almost nothing in common.

In addition, an *At Large Advisory Committee* was formed.

The reform process has had some more tangible outcomes, in that formal open meetings of the ICANN Board of Directors have managed to be progressively refined from efforts at direct dialogue and open debate into highly structured events with many formalisms and appropriate quantities of ceremony.

ICANN Today

Despite the effort to encompass coordination activities in the areas of names, addresses, and protocol parameters, ICANN has been largely captured by the names industry, and ICANN's agenda, activity focus, and outcomes are concentrated mostly in the name domain.

In this activity domain, the track record of ICANN is very mixed. To its credit, it has managed to dismantle the most objectionable parts of the monopoly hold over the *generic Top-Level Domains* (gTLDs), create an operational model that makes a clear distinction between registry operators and registrars, impose price and business controls on the registry operation as a means of controlling the natural tendency for the registry operation to reflect its unique position in the form of monopoly rentals, and assist in the creation of a global network of competitive enterprises, with the expectation that competition will instill operational and price efficiency in the registrar business.

In addition, ICANN has been successful in not only introducing new gTLDs to complete with the established brands of **.com**, **.net**, and **.org**, but also in moving **.org** and **.net** to new registry operations (**.net** is under way at the time of writing of this article). Despite these positive achievements, it is not clear that this new regime has been entirely successful.

True competition in the name space is still some way off, and the recently introduced gTLD brands have failed to gain any leverage within the market. The name market itself remains one where the role of name speculators continues to play a significant role in terms of proportion of registered names. The overarching dominance of **.com** as a brand has continued, and the advantaged position of the U.S.-based registrar of this zone continues.

The obscure nature of the relationships between the IETF, ICANN, and the U.S. administration over the protocol parameter registries remains unresolved. The IETF is clearly not in control of its own protocol parameters, and has abrogated this role to ICANN. Standards making entirely divorced from any effective engagement with deployment tends to result in a standards body of dubious long-term validity, and despite its impressive track record in the past, the IETF is clearly already well-distanced from current technology directions in the industry—and the gap continues to widen.

The DNS *Root Server Operators* continue to operate as an independent group. The recent moves to dramatically increase the number of DNS root servers and improve the overall robustness of DNS resolution through anycasting root servers and distributing anycast instances across the globe has been a well-received initiative. The fact this has occurred without any form of ICANN involvement is an interesting commentary on the ability of ICANN to engage with the operational parts of the infrastructure of the Internet. Comparable activities to improve the DNS in terms of resolution services within the ICANN sphere have become protracted exercises that impose a very heavy burden on the patience of the players.

The moves to introduce IPv6 AAAA records into the DNS root have been anticipated for many years, and the response to the recent ICANN announcement is, in general, of the tenor "why didn't this happen some years ago?" The continuing frustration to get the DNS root to include *Secure DNS* (DNSSEC)[5] important information continues to illustrate a perspective that the ICANN process appears to be unresponsive to technical needs and end-user imperatives.

The situation today is that ICANN appears to enjoy a mixed level of success. It has managed to establish itself as a means of administering the infrastructure elements of the Internet Protocol in a manner that is reflective of the deregulated nature of the Internet industry. It has managed to reform parts of the landscape and generate an industry structure that uses open competition as the major control mechanism. ICANN has managed to bring much of the discussion about the administration of Internet infrastructure out into the open. All these are major milestones, and it is to the credit of many dedicated individuals that ICANN has managed these impressive outcomes. However, it has been able to achieve all this with the continued sponsorship of the U.S. administration, and the question of whether it can firmly establish itself in its own right in the coming years remains today perhaps a matter of hope rather than absolute certainty.

There are still the lingering concerns that if ICANN, as a private-sector entity, were to once more explore positioning itself on the brink of imminent demise, the collective task of picking up the pieces and continuing to support the operation of the Internet is one that appears to have a very uncomfortable level of uncertainty. In addition, the perception of ICANN as an entity whose single purpose is to maintain an entrenched advantaged position of the United States and of U.S.-based enterprises in the global Internet has been widely promulgated. It is often portrayed that ICANN offers no viable mechanisms for other national or regional interests at a governmental level to alter this somewhat disturbing picture of international imbalance. Although other aspects of international activity fall under various political or trading frameworks, and national and regional interests and positions can be collectively considered and negotiated, critics of ICANN point out that the message ICANN sends to the rest of the world is that the United States is withholding the Internet from conventional international governance processes. Skeptical commentators interpret the U.S. administration's use of ICANN as at best a delaying technique to gain time to further strengthen the position of U.S.-based enterprises across a lucrative global Internet market, aided and abetted by a compliant industry body that masquerades as an international standards organization.

Such a critical perspective also points to ICANN's tenuous lines of authority, its lack of performance in many aspects of the domain name enterprise, its seeming obsession with the registrar sector to the apparent exclusion of any other activity, its burgeoning costs, and its lack of acceptance, particularly as it relates to the acceptance of ICANN by the various country code DNS administrators, to name but a few factors.

Accompanying this strident criticism is the line of argument that the Internet does not actually represent a viable challenge to existing mechanisms for coordination of international activity. At both a national and international level, the Internet should not require novel and untested regulatory mechanisms as a means of expressing public interest and public policies. The line of argument from this perspective is that there is neither the demonstrated need, nor any appropriate level of international support at a governmental level to sustain the argument that a private-sector, nonprofit corporation is the best, or even the only viable model of coordination of Internet activity. If "Internet Governance" is the question, then, the line of argument goes, the model upon which ICANN is based is definitely not the best answer we can devise. This very critical line of reasoning has become particularly prominent in the WSIS process, and lies behind much of the continual fascination of the topic of "Internet Governance" in WSIS meetings.

WSIS and Internet Governance

The WSIS has been a long time coming, and it represents a move on the part of the ITU to formulate a revised role for the ITU to engage with a world richly populated by all manner of information services layered upon a highly diverse and capable communications environment. This summit was planned in two phases. The first summit was held in Geneva December 10–12, 2003, where the foundations were laid by reaching agreement on a *Declaration of Principles* and a *Plan of Action*. The second phase will be held in Tunis, November 16–18, 2005, to implement the agenda leading up to achievable targets by 2015, and to agree on unfinished business, most importantly on the question of Internet governance and of financing mechanisms.

Irrespective of any particular political perspective here, the universal observation is that the Internet has heralded a revolutionary change to the global communications enterprise. Markets for communications services are changing, the technology base is changing, the economic models of communication are changing, and the models of interaction at the provider level are changing. The challenge from the public-policy perspective at a world level is to create a framework that ensures that the benefits of this change, in both social and economic terms, are accessible to all, rather than to a subset of the world's population. It is within this broad framework that WSIS has been positioned.

These are lofty and ambitious goals, and the task before WSIS is certainly as challenging as any in this environment. The hope is that the myriad of participants in this process includes sufficient resources to engage in the agenda in a meaningful way.

However, the underlying issue is that of the progressive change in the role of communications infrastructure from a predominately public-sector activity to a very diverse spectrum of public- and private-sector activity. We appear to have become increasingly reliant on private-sector investment and private enterprise to support the public communications enterprise. But is this necessarily the appropriate model for the entire world, or even any part of the world?

As many recently privatized industries could attest, private-sector activity has entirely different investment motivations and entirely different service objectives. If the nature of the activity is one that requires long-term investment in infrastructure with low returns, then private-sector activity tends to use the existing infrastructure base without necessarily making adequate longer-term replenishment investments. Private activity also tends to concentrate service delivery to the most lucrative sectors of the market, and, if possible, will deliberately avoid establishing services in areas that are less financially attractive. The task of structural cross-subsidization that makes ubiquitous equity of access possible is not seen as a private enterprise outcome, and aspects of communications such as universal service obligations and equity of access are seen as public regulatory functions rather than natural market outcomes of a deregulated industry.

The Internet today is anything but a level and balanced environment. There are concentrations of investment capability, concentrations of technical knowledge and logistical capability, concentrations of intellectual wealth, and concentrations of power and influence. How to create from this current diverse environment some form of structural cross-subsidization that extends the basic means of access to all is the appropriately lofty goal of the WSIS endeavor. There is also the more focused investigation of "Internet Governance" and the agenda of establishing to what extent the perception of the advantaged position of a small number of national entities in all this can be balanced by measures that allow other national economies to invest in this space on terms and conditions that do not involve a continuing flow of money and a ceding of power to these existing advantaged national interests.

As the WSIS documentation points out, "... building the foundations for an Information Society is a complex task. The digital revolution is already impacting the world in deeply intrinsic ways, perhaps more profoundly than even the industrial revolution itself. Yet, while the digital revolution has extended the frontiers of the global village, the vast majority of the world remains unhooked from this unfolding phenomenon."

The Secretary General of the UN chartered a smaller group to examine Internet Governance, in particular, the *Working Group on Internet Governance,* or WGIG. Its nine-month brief is to glean these issues of public policy in an environment that has very significant private-sector interest. Indeed from an international perspective, where regulatory powers, even of a reserve nature, are in a very real sense ephemeral, the work in WGIG to date with its discussion papers has done little. The discussion papers have illustrated the broad nature of the topics raised in the context of Internet Governance, but their poor depth, visibly poor levels of research, and lack of any real analysis of the selected topics only highlights the complexity of the underlying interplay of public- and private-sector interests within a domain that is also bounded by technical considerations.

At the same time the poor quality of these reports highlights the inability of WGIG to engage directly into the heart of this exercise, given their obvious constraints of time and resources. It is not surprising to observe that, following its February meeting WGIG has decided to abandon this set of discussion papers. If a fresh start is being contemplated for WGIG, then perhaps it is time to note that only half of the group's allocated time remains, and the topic is getting no easier with the passing of the days.

For those interests who wanted the ITU to become engaged in the Internet, hope has now been passed to the WSIS process and the related WGIG study into Internet Governance issues. This is seen as being a means of opening up the control of the Internet into a more conventional international process that dismantles what they see as the current position of global taxation that U.S. national interests have imposed on the rest of the world's population in the adoption of Internet-based services. For those who think the ITU remains an unreformed vehicle for the imposition of anachronistic, inappropriate regulatory measures that stultify any form of innovation and progress in telecommunications, the WSIS process is yet another venue to parade the stark contrast between the rather impressive track record of a deregulated market-driven approach to coordination of telecommunications services, as seen with the Internet, and the ineffectual outcomes from the international public regulatory sector.

Looking Forward

One view of this process is that this is a negotiation of national roles of influence and power over the coming century or more, and that this process requires some considerable care and attention at an international level.

This topic is one that places a model of deregulated private sector-led activity, with its market-based disciplines, into direct contrast with a more traditional model of the balancing of various national interests through common regulatory measures undertaken within each national regime as a regulated public-sector process. The proponents of a deregulated approach argue that the Internet is a child of the progressive position of deregulation of communications markets in many national environments, and it is the dynamic and creative impetus of highly competitive markets that has led to the rapid spread of the Internet and the consequent improvements in the efficiency and effectiveness of national and international communications systems. None of these outcomes would have been achievable, they argue, in a regulated regime where innovation and competition for the consumer were completely stifled by the deadening weight of regressive regulation.

Like many bold innovative experiments in international coordination and the establishment of new world orders, ICANN stands a strong risk of falling foul of an inherent conservatism in international politics, where the careful balancing of national interests is seen as being far more critical an objective than any actual outcomes that may be achieved from the process.

From this perspective, ICANN is critically reliant on its acceptance by all players of its legitimacy to operate in this space, and also critically reliant on acceptance of the proposition that these issues are best addressed in open forums of debate. This task is difficult, and the limited set of outcomes that ICANN can point to as being products of this process do not install a high degree of confidence that this process is stable, scalable, well-founded, and sustaining. Currently the proposition is not that ICANN represents the most appropriate enduring framework here, but that the track record of the alternative has failed in the past and nothing has changed to prevent the historical alternative framework making similar flawed decisions in the future.

The opposite end of the spectrum of views argues that nothing has really changed with the introduction of the Internet, and the international regime remains one where various national interests need to be resolved in a coordinated and equitable fashion. Without some form of common regulatory constraint, there are inevitable market distortions where the expression of vigorous national aspirations results in an advantaged position in the international domain. Public communications is a public-sector activity, they argue, and, ultimately, the only points of control rest within national regulatory regimes, and internationally it is a case where national interests must be balanced through a process that recognizes political realities of coordination and compromise. From this perspective it is asserted that the ITU is the intergovernmental venue for this activity as it relates to the communications sector, and it is to the ITU that national interests must look to redress distortions where one national entity or one region holds a contrived privileged position with respect to international communications.

In looking at these two extremes of perspective, an obvious question is what then is the role of international public policy setting? In this form of market-mediated service supply functions, are international issues being progressively transformed into aspects of international trade? Does such an environment provide adequate protection for developing economies? Are common social priorities being adequately considered in such a framework?

This leads to a more basic question of whether the existing international institutions, such as the ITU, are appropriately positioned to meet these public policy challenges, or should we be considering changes here in order to bring the international institutional framework into better alignment with the emerging information society?

These are certainly difficult positions to attempt to reconcile, and perhaps it is being impatient to expect clear outcomes in the near future, and certainly very difficult to expect that in a few short months WGIG and WSIS will be able to deliver a balanced, considered, and generally acceptable outcome in this space. It is also a natural concern in looking at these rather aggressive schedules for WSIS that short-term political expediency will obstruct genuine attempts to truly understand the fundamental nature of the changes that are happening with the differing model of communications that are heralded by the Internet model.

References

[1] Internet Corporation for Assigned Names and Numbers (ICANN): **http://www.icann.org**

[2] The World Summit on the Information Society (WSIS): **http://www.itu.int/wsis/**

[3] M. Stuart Lynn, "A Unique, Authoritative Root for the DNS," *The Internet Protocol Journal,* Volume 4, No. 3, September 2001.

[4] Number Resource Organization (NRO): **http://www.nro.net/**

[5] Miek Gieben, "DNSSEC: The Protocol, Deployment, and a Bit of Development," *The Internet Protocol Journal,* Volume 7, No. 2, June 2004.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector, and has served his time with Telstra, where he was the Chief Scientist in the company's Internet area. Geoff is currently the Internet Research Scientist at the Asia Pacific Network Information Centre (APNIC). He is also the Executive Director of the Internet Architecture Board, and is a member of the Board of the Public Interest Registry. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, and co-author of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@apnic.net**

Book Review

Unix Network Programming

Unix Network Programming, 3rd Edition, by W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, ISBN 0131411551, Addison-Wesley Professional, 2003.

It would be difficult to put value on a book that has been a classic text and a reference in academia and in the real world in the context of network programming for over a decade. Richard Stevens published the ever-popular *Unix Network Programming* [UNP] back in 1990, and the second edition followed in 1998. With a dedication to the memory of R. Stevens, the UNP book found itself two new authors, Bill Fenner and Andrew M. Rudoff, who would write the third edition of this book. The third edition has many updates, a new look and feel and many of new chapters that cover the topics more applicable these days. In my opinion, it is still the most valuable and profound text in the context of network programming.

Changes and Updates

For those of us who have the first two editions of this book, the third edition has the following changes:

- IPv6 updates. In the second version of the book, IPv6 was merely a draft, and the sections covering IPv6 have been updated to reflect these changes.
- POSIX updates. The functions/APIs and examples have been updated to reflect the changes to the latest version of the POSIX specification (1003.1-2001).
- SCTP coverage. Three new chapters that cover this new reliable, message-based transport protocol have been added.
- Key Management Sockets coverage. Network security and its applicability and use with IPsec are covered.
- The Operating Systems and machines that are used for the examples have been updated.
- Some topics such as Transaction TCP and X/Open Transport Interface have been dropped.

Many topics and sections have been updated with the authors' comments. These comments even though simple for someone new to the profession, are extremely useful because they are like hints and tips from one developer to the next to help you in your next programming assignment.

Unix Focus

If this is the only edition of the book that you will read, you are in for a treat. Topics in Network Programming are covered in detail, using concrete programming examples that all of us can relate to—all Unix, but what else is there?!

All kidding aside, the topics are covered well enough that they are useful information under any operating system. The concepts don't change; sockets are sockets under any operating system. The function call is different, but one needs to go through the same steps under any environment.

Being the most popular networking protocol, TCP/IP is covered in Part I of the book. You need to have prior understanding of the TCP/IP protocol and the OSI model, however. If this is the first time you are looking at the programming aspects of networking protocols, Part I of this book covers the basics. It begins with a couple of simple examples such as such as daytime client and a daytime server and it builds on that. TCP, UDP, and SCTP (*Stream Control Transmission Protocol*) are covered in brief in Part I, and basic concepts such as the three-way handshake of TCP and the four-way handshake of SCTP are depicted.

Part II of the book covers *sockets* and socket programming. Topics such as the socket Address Structure in IPv4 and IPv6 for TCP, UDP and SCTP are covered and examples (the same daytime client/server) are given to convey the point. It is important to mention here that all the topics and concepts are depicted for the three transport protocols: TCP, UDP and SCTP. Every socket API under the Unix programming environment is covered and examples are given for each function call to show the reader how the function can be utilized. Much attention is dedicated to Socket Options and how they are used or can be used for best results. Hints are given throughout the chapter about the pitfalls and best practices of each option.

After the basics are been covered, various I/O models are depicted in detail and examples are shown to convey the advantages and disadvantages of each I/O model. The five I/O models used through the book (and available under the Unix environment) follow:

- Blocking I/O
- Non-blocking I/O
- I/O Multiplexing (using select and poll)
- Signal driven I/O
- Asynchronous I/O

The *Stream Control Transmission Protocol* (SCTP), a new IETF standard is also covered in detail—from the basics to the advanced. The two interface models of SCTP (one-to-one and one-to-many) are covered in detail, and their differences with TCP are also explained in full. The client/server example used throughout the book is ported to use the new SCTP protocol. The authors then explain in detail the problems that SCTP solves over TCP and where and how it would be useful to use SCTP.

Advanced topics such as IPv4 and IPv6 portability, Unix Domain Protocols, Multicasting and advanced Socket programming for UDP, TCP and SCTP cover the rest of the chapters in this book.

Various options for interoperability between IPv4 and IPv6 are discussed in the last section of the book. Advanced I/O functions bring us a new perspective of how complicated Network Programming can become. Benefits and examples of nonblocking I/O are covered in detail—the authors give examples to show us how, with very few modifications, the performance of a socket application can improve dramatically. Various methods on how to control socket operations are discussed including the use of an alarm along with SIGALRM, the use of select and various timeout options that are available in the API.

The chapters that discuss Multicasting and adding reliability to UDP are my favorite chapters in this book. The Time Server used throughout the book is re-coded to become a multicast application. Some issues that arise when designing multicast applications such as multicast on a WAN are also discussed.

As Good as Ever

The third edition of *Unix Network Programming* is as good as ever. The updates truly reflect solutions to today's challenges in network programming. Bill Fenner and Andrew Rudoff did an amazing job continuing the work of a true legend in the field of Computer Science.

—Art Sedighi
asedighi@tibco.com

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

Internet Pioneers Cerf and Kahn to Receive ACM Turing Award

The *Association for Computing Machinery* (ACM), has named Vinton G. Cerf and Robert E. Kahn the winners of the 2004 *A.M. Turing Award,* considered the "Nobel Prize of Computing," for pioneering work on the design and implementation of the Internet's basic communications protocols. The Turing Award, first awarded in 1966, carries a $100,000 prize, with financial support provided by Intel Corporation. Cerf and Kahn developed TCP/IP, a format and procedure for transmitting data that enables computers in diverse environments to communicate with each other. This computer networking protocol, widely used in information technology for a variety of applications, allows networks to be joined into a network of networks now known as the Internet.

ACM President David Patterson said the collaboration of Cerf and Kahn in defining the Internet architecture and its associated protocols represents a cornerstone of the information technology field. "Their work has enabled the many rapid and accessible applications on the Internet that we rely on today, including e-mail, the World Wide Web, Instant Messaging, Peer-to-Peer transfers, and a wide range of collaboration and conferencing tools. These developments have helped make IT a critical component across the industrial world," he said.

"The Turing Award is widely acknowledged as our industry's highest recognition of the scientists and engineers whose innovations have fueled the digital revolution," said Intel's David Tennenhouse, Vice President in the Corporate Technology Group and Director of Research. "This award also serves to encourage the next generation of technology pioneers to deliver the ideas and inventions that will continue to drive our industry forward. As part of its long-standing support for innovation and incubation, Intel is proud to sponsor this year's Turing Award. As a fellow DARPA alumnus, I am especially pleased to congratulate this year's winners, who are outstanding role models, mentors and research collaborators to myself and many others within the network research community."

In 1973, Cerf joined Kahn in a *Defense Advanced Research Projects Agency* (ARPA, now called DARPA) project to link three independent networks into an integrated "network of networks." They sought to develop an open-architecture network model for heterogeneous networks to communicate with each other independent of individual hardware and software configuration, with sufficient flexibility and end-to-end reliability to overcome transmission failures and disparity among the participating networks. Their collaboration led to the realization that a "gateway" (now known as a *router*) was needed between each network to accommodate different interfaces and route packets of data. This meant designating host computers on a global Internet, for which they introduced the notion of an *Internet Protocol* (IP) address.

As a graduate student at the University of California at Los Angeles, Cerf had contributed to a host-to-host protocol for ARPA's fledgling packet-switching network known as ARPANET. Kahn, prior to his arrival at ARPA, led the architectural development of the ARPANET packet switches while at Bolt Beranek and Newman (BBN), and had showcased the ARPANET in 1972, at the first International Conference on Computer Communications. ARPANET had already connected some 40 different computers and demonstrated the world's first networked e-mail application.

In May 1974, they published a paper describing a new method of communication called *Transmission Control Protocol* (TCP) to route messages or packets of data. Like an envelope containing a letter, TCP broke serial streams of information into pieces, enclosed these pieces in envelopes called "datagrams" marked with standardized "to and from" addresses, and passed them through the underlying network to deliver them to host computers. Only the host computers would "open" the envelope and read the contents.

This networking arrangement allowed for a three-way "handshake" that introduced distant and different computers to each other and confirmed their readiness to communicate in a virtual space. In 1978, Cerf and several colleagues split the original protocol into two parts, with TCP responsible for controlling and tracking the flow of data packets ("letters"), and IP responsible for addressing and forwarding individual packets ("envelopes"). The new protocol, TCP/IP, has since become the standard for all Internet communications.

Vinton Cerf and Robert Kahn share a number of awards, including the 1991 ACM Software System Award, the 2001 Charles Stark Draper Prize from the National Academy of Engineering, the 2002 Prince of Asturias Award, and the 1997 National Medal of Technology from President Bill Clinton. They are both the recipients of numerous honorary degrees. ACM will present the Turing Award at the annual ACM Awards Banquet on June 11, 2005, in San Francisco, CA.

The A.M. Turing Award was named for Alan M. Turing, the British mathematician who articulated the mathematical foundation and limits of computing, and who was a key contributor to the Allied cryptanalysis of the German Enigma cipher during World War II. Since its inception, the Turing Award has honored the computer scientists and engineers who created the systems and underlying theoretical foundations that have propelled the information technology industry.

For additional information see:
http://www.acm.org/awards/taward.html

New Administrative Structure for the IETF

The *Internet Engineering Task Force* (IETF) is well advanced in the process of making a significant change to the administrative structure that supports the world's leading Internet standards development group. The creation of an *IETF Administrative Support Activity* (IASA) is an important move designed to help the IETF maintain and expand the unique open processes that have enabled the development of Internet standards since 1986.

The new structure will allow the IETF to take full responsibility for managing the resources required to accomplish its work—giving the IETF a solid foundation on which future operations will be based.

This is the first time that all the IETF's administrative and support functions will be managed directly by the IETF as one fully integrated entity. Until now, administration of the IETF has been carried out exclusively by helper organizations and volunteers. The new IASA will be formally structured as an activity within the *Internet Society* (ISOC)—the organizational home of the IETF—and an *IASA Administrative Director* (IAD) will be appointed to provide central management of IETF administration.

The decision to move forward with the new structure was taken after extensive consultations with the Internet community. A number of key prerequisites for efficient administrative operations were identified, including the need for the IETF to have budgetary autonomy. The IETF is currently supported by funding from multiple sources, including meeting fees, donations from interested corporate and non-corporate entities, and donations in kind of equipment or manpower. The IASA will allow the IETF to be able to consider all sources of income, and all expenses involved in running the IETF, as pieces of one budget.

The IASA will also be responsible for defining clear contractual relationships with other organizations that will continue to provide basic services, including meeting organization, secretarial services, IT services, etc. The new structure also gives the IETF flexibility in how it chooses to fund and develop any additional services that may be required.

The IETF is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is open to any interested individual. See: **`http://www.ietf.org`**

ISOC is a non-governmental international organization for global cooperation and coordination for the Internet and its internetworking technologies and applications. Members comprise commercial companies, governmental agencies, foundations, and individuals. ISOC has 82 Chapters in over 60 countries around the world. For more information see: **`http://www.isoc.org`**

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

June 2005 Volume 8, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

The Internet is a constantly evolving environment which puts pressures on existing and evolving protocols. Any protocol changes must be carefully designed and even more carefully deployed to avoid any disruption to the running system. It is no longer possible to orchestrate a simple overnight switch, so engineers are considering various transition and evolution strategies. In this issue we bring you two examples of this kind of evolutionary protocol development.

Our first example relates to *IP Version 6* (IPv6). A great deal of effort is going into the deployment of IPv6, and good transition strategies can help. Tejas Suthar explains how *Multiprotocol Label Switching* (MPLS) can be used for a transition from IPv4 to IPv6.

Our second example looks at a possible enhancement to the *Border Gateway Protocol* (BGP). BGP in its current form is already nearly ten years old, and calls for its replacement can be heard from network operators. Russ White discusses some possible changes that would not require a wholesale protocol replacement.

It is not every day that a book on punctuation becomes an international best seller, and it is certainly not common for IPJ to review such a non-computer related book. But I think it is appropriate for several reasons. First, accurate punctuation is important not just for computer parsers, it is important for all professionals whether we are sending quick e-mails or writing project reports. Second, this is a really *fun* as well as informative book. And last, but not least, it gives me an opportunity to introduce you to Bonnie Hupton, who provides copy-editing services for this journal. Without her help, IPJ would be far less readable.

Our Website at **www.cisco.com/ipj** has a new look, but still contains links to our back issues, index files and the IPJ subscription system. Please take a moment to renew or update your subscription. If you have questions or comments, please send them to **ipj@cisco.com**.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

IPv6—A Service Provider View in Advancing MPLS Networks

by Tejas Suthar, TELUS Communications Inc.

We are all aware of the evolution of the *Internet Protocol* (IP) and its dominance on all aspects of our lives, either directly or indirectly. Currently IP Version 4 delivers critical business application traffic in a so-called new world of the Internet. As the evolution goes on, *IP Version 6* (IPv6)[5] is becoming a necessary element of the network. IPv6 will enable businesses to expand their capabilities exponentially without having any limitations or restrictions. As technologies evolve and the adoption of IP-enabled devices accelerates, IP will enter a new era as the protocol of choice for communications. Using globally unique IPv6 addresses increases the opportunity for service providers to create new business models and add revenue, and it increases the portfolio of services. However, the major demand for support of IPv6 will be mobile applications; the IT world will also tie in all the systems for transparent operation. The days are not far when permanent IPv6 addresses will be assigned to individuals for their communication purposes—either *Voice over IP* (VoIP), video over IP, video on demand, wireless Internet access, unified messaging, etc. Also, IP smart appliances are becoming more and more popular, and the result will be explosive usage and adoption of IPv6 addresses. Articles outlining the importance of IPv6 and limitations of IPv4 abound. This article is mainly geared toward highlighting the service provider networks that are built or currently being built to support IPv6 in a VPN fashion.

Multiprotocol Label Switching (MPLS)[4] is widely accepted as a core technology for the Next-Generation Internet that provides speed and functions in packet forwarding. Service providers that offer MPLS/VPN services to their customers are looking forward to adding IPv6 VPN services to their portfolio. Service providers that want to support IPv6 in traditional ways have few options, such as tunneling methods (for example, manual, *Tunnel Broker, Generic Routing Encapsulation* [GRE], or *Intrasite Automatic Tunnel Addressing Protocol* [ISATAP], which has scalability problems); or Native IPv6 with dual-stacked MPLS core. However, consider the following:

- For MPLS VPN services, service providers made a significant investment in building the IPv4/MPLS backbone. The return on investment thresholds are probably yet to be achieved.
- Backbone stability is another critical factor; service providers must offer reliable services, especially with regard to voice over MPLS. Most service providers have recently managed to stabilize their IPv4 infrastructure, and they are hesitant to make another significant move when it comes to supporting IPv6 unless the integration is smooth.

Standards bodies with help from vendors and leading service providers are addressing these concerns. Currently service providers have two approaches that they can deploy to support IPv6 without making any changes to the current IP (v4) MPLS backbones, namely *6PE*[1] and *6VPE*[2], originally defined in RFC 2547.

The 6PE approach lets IPv6 domains communicate with each other over an IPv4 cloud without explicit tunnel setup, requiring only one IPv4 address per IPv6 domain. The 6PE technique allows service providers to provide global IPv6 reachability over IPv4 MPLS. It allows one shared routing table for all other devices. Typical applications are IP toll voice traffic and Internet transit services over a common MPLS infrastructure. The 6PE technique does not provide any logical separation because it is for MPLS VPN.

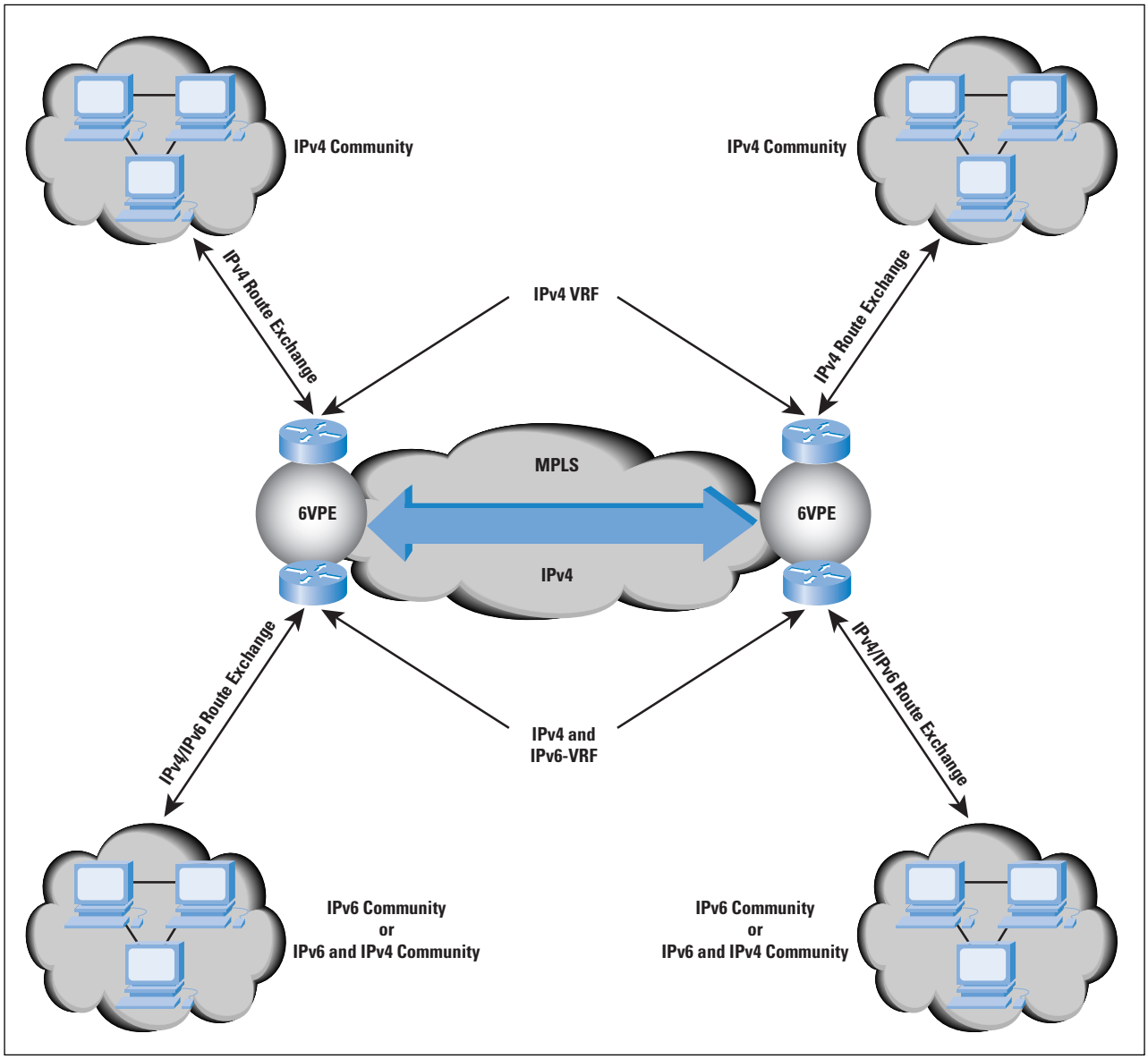
The newest feature to facilitate the RFC 2547bis-like VPN model for IPv6 networks is called 6VPE. It will save service providers from enabling a separate signaling plane, and it takes advantage of operational IPv4 MPLS backbones. Thus there is no need for dual-stacking within the MPLS core. This represents a huge cost savings from the operating expenses perspective and addresses the security limitations of the 6PE approach. 6VPE is more like a regular IPv4 MPLS-VPN provider edge, with an addition of IPv6 support within *Virtual Routing and Forwarding* (VRF). It provides logically separate routing table entries for VPN member devices. This article reviews this approach in more detail because it is the likely approach to succeed in the service provider network.

Under the Hood of 6VPE

Before we look into the 6VPE, it is important to clarify the definition of "dual stack," a technique that allows IPv4 and IPv6 to coexist on the same interfaces. Today, IPv4 has roots in most of the hosts that run applications. Moreover, stability as well as reliability of new applications over IPv6 is maturing. Therefore, coexistence of IPv4 and IPv6 is a requirement for initial deployment. With regard to supporting IPv6 on a MPLS network, two important aspects of the network should be examined:

- *Core:* The 6VPE technique allows carrying IPv6 in a VPN fashion over a non-IPv6-aware MPLS core. It also allows IPv4 or IPv6 communities to communicate with each other over an IPv4 MPLS backbone without modifying the core infrastructure. By avoiding dual-stacking on the core routers, the resources can be dedicated to their primary function to avoid any complexity on the operational side. The transition and integration with respect to the current state of networks is also transparent.
- *Access:* In order to support native IPv6, the access that connects to IPv4/IPv6 domains need to be IPv6-aware. Service provider edge elements (provider edge routers) can exchange routing information with end users. Hence dual stacking is a mandatory requirement on the access layer as shown in Figure 1.

Figure 1: 6VPE Overview



The IPv6 VPN solution defined in this article offers many benefits. Especially where a coexistence of IPv4 and IPv6 is concerned, the same MPLS infrastructure can be used without putting additional stress on the provider router. Also the same set of *Multiprotocol Border Gateway Protocol* (MPBGP) peering relationships can be used. Because it is independent of whether the core runs IPv4 or IPv6, the IPv6 VPN service supported before and after a migration of the core to IPv6 can be done independent of the customer VPN.

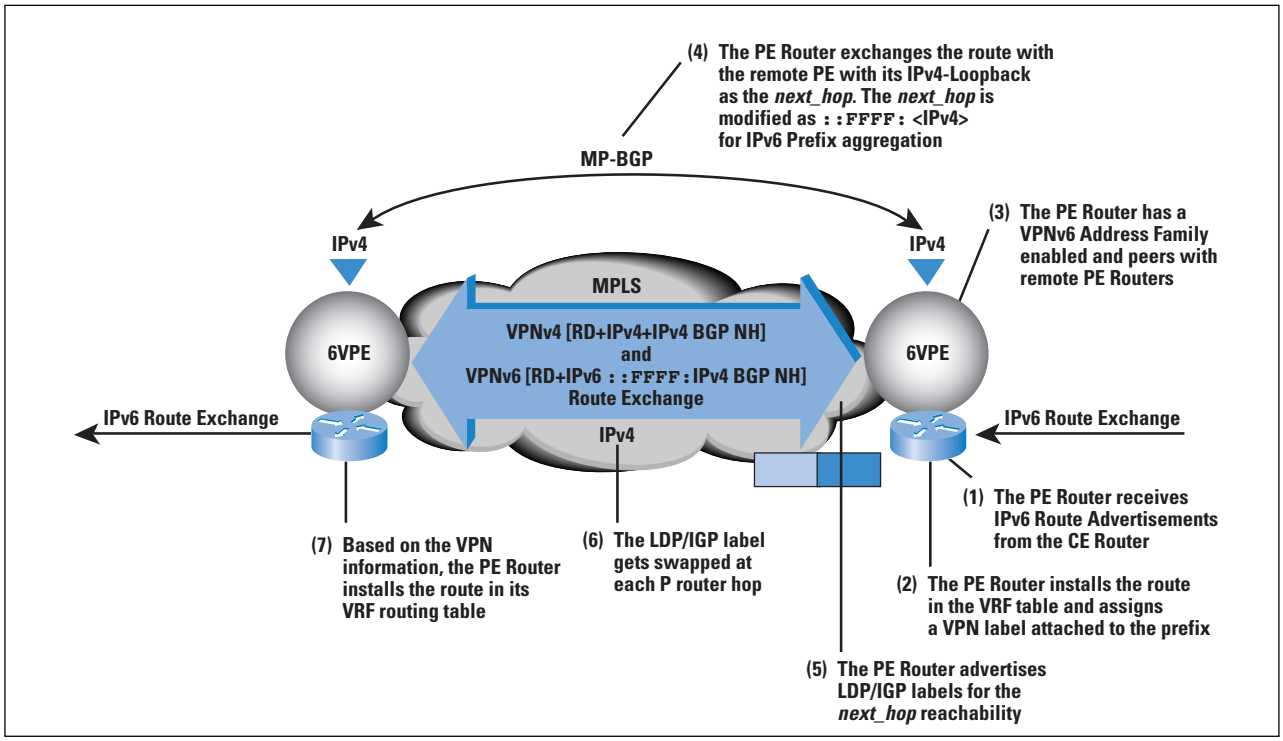
Within the MPLS core, the backbone *Interior Gateway Protocol* (IGP) (*Intermediate System-to-Intermediate System* [IS-IS] or *Open Shortest Path First* [OSPF]) populates the global routing table (v4) with all provider edge and provider routes. As outlined in the draft for IPv4 MPLS VPN (2547-bis), 6VPE routers maintain separate routing tables for logical separation. This allows the VPN to be private over a public infrastructure.

The VRF table associated with one or more directly connected sites (customer edge devices) form close IPv6 or IPv4 speaking communities. The VRFs are associated to physical or logical interfaces. Interfaces can share the same VRF if the connected sites share the same routing information. MPLS nodes forward packets based on the top label. IPv6 packets and IPv4 packets share the same common set of forwarding characteristics or attributes, also known as *Forwarding Equivalence Class* (FEC) within the MPLS core.

6VPE Operation

When IPv6 is enabled on the sub-interface that is participating in a VPN, it becomes an IPv6 VPN. The customer edge-provider edge link is running IPv6 or IPv4 natively. The addition of IPv6 on a provider edge router turns the provider edge into 6VPE, thereby enabling service providers to support IPv6 over the MPLS network.

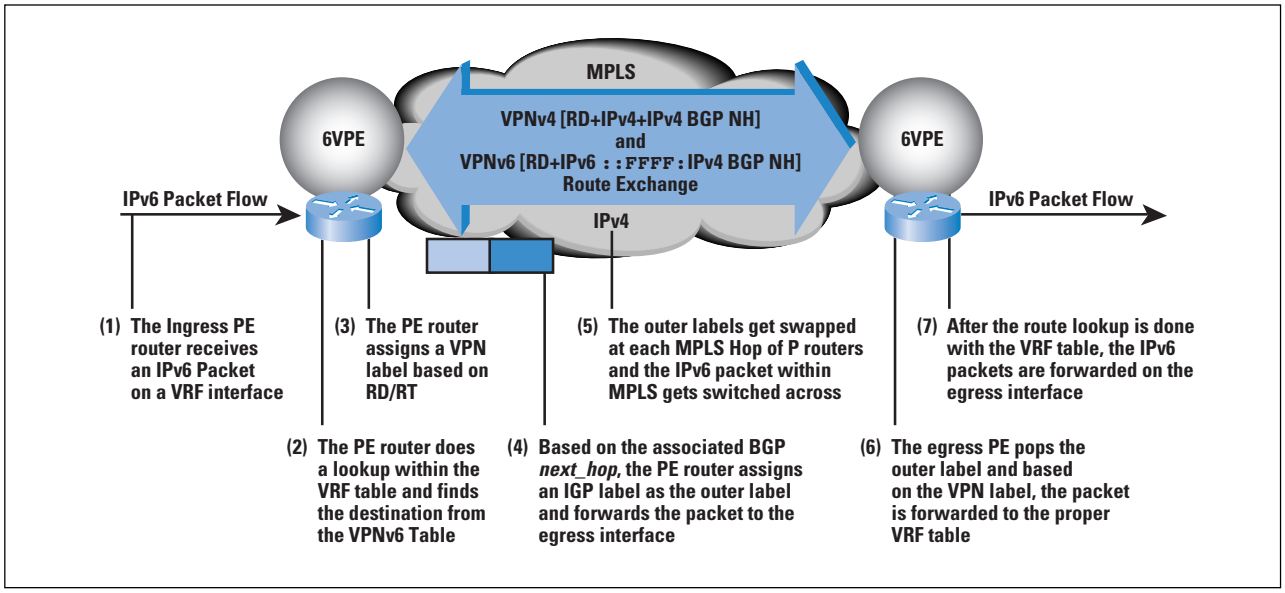
Figure 2: 6VPE Route Advertisement



As outlined in Figure 2, provider edge routers use VRF tables to maintain the segregated reachability and forwarding information of each IPv6 VPN. MPBGP with its IPv6 extensions distributes the routes from 6VPE to other 6VPEs through a direct *internal BGP* (iBGP) session or through VPNv6 route reflectors. The next hop of the advertising provider edge router still remains the IPv4 address (normally it is a loopback interface), but with the addition of IPv6, a value of **`::FFFF:`** gets prepended to the IPv4 *next_hop*. The technique can be best described as automatic tunneling of the IPv6 packets through the IPv4 backbone. The MP-BGP relationships remain the same as they are for VPNv4 traffic, with an additional capability of VPNv6. Where both IPv4 and IPv6 are supported, the same set of MPBGP peering relationships is used.

MPBGP is enhanced to carry IPv6 in a VPN fashion known as VPNv6, which uses a new VPNv6 address family. The VPNv6 address family consists of 8 bytes—a *Route Distinguisher* followed by a 16-byte IPv6 prefix. This combination forms a unique VPNv6 identifier of 24 bytes. The Route Distinguisher value has a local significance on the router, and the *Route Target* advertises the membership of the VPN to other provider edge routers.

Figure 3: 6VPE Packet Forwarding



In Figure 3, packet forwarding is explained showing end-to-end operation. When the ingress 6VPE router receives an IPv6 packet, destination lookup is done in the VRF table. This destination prefix is either local to the 6VPE (which is another interface participating in the VPN) or a remote ingress 6VPE router. For the prefix learned through the remote 6VPE router, the ingress router does a lookup in the VPNv6 forwarding table. The VPN-IPv6 route has an associated MPLS label and an associated BGP *next_hop* label. This MPLS label is imposed on the IPv6 packet. The ingress 6VPE router performs a PUSH action, which is a top label bind by the *Label Distribution Protocol* (LDP)/IGPv4 to the IPv4 address of the BGP *next_hop* to reach the egress 6VPE router through the MPLS cloud. This topmost-imposed label corresponds to the *Label Switched Path* (LSP). So, the bottom label is bound to the IPv6 VPN prefix through BGP and the top label is bound by the LDP/IGP. The IPv6 packet, now with two labels, gets label-switched through the IPv4/MPLS core router (provider routers) using the top label only (referred to as the *IGP label*). Because only the top label is of significance to the provider core, it is unaware of the IPv6 information in the bottom label.

The egress provider edge router, receives the labeled IPv6 VPN packet and performs a lookup on the second label, a process that uniquely identifies the target VRF and the egress interface. A further Layer 3 lookup is performed in the target VRF, and the IPv6 packet is sent toward the proper customer edge router in IPv6 domain.

In summary, from the control plane perspective the prefixes are signaled across the backbone in the same way as for regular MPLS/VPN prefix advertisements. The top label represents the IGP information that remains the same as for IPv4 MPLS. The bottom label represents the VPN information that the packet belongs to. As described earlier, additionally the MPBGP *next_hop* is updated to make it IPv6-compliant. The forwarding or data plane function remains the same as it is deployed for the IPv4 MPLS VPN. The packet forwarding of IPv4 on the current MPLS VPN remains intact.

6VPE Design Recommendations and Considerations

The following sections identify general recommendations that should be considered when deploying IPv6 in a service provider network:

Working with Enterprise Implementations

Typically *Customer Metropolitan-Area Networks* (C-MANs), also known as *Campus Networks* or *Customer LAN* (C-LAN) elements, form the enterprise network, whereas the 6VPE and customer edge provide the entry point into network access. IPv6 can be supported partially or fully on an enterprise network. In situations where enterprise-wide IPv6 deployment does not exist, network administrators can elect to tunnel the IPv6 traffic toward the provider's customer edge or 6VPE. This can be done with 6-to-4 tunneling methods currently[7]. So, if a site router within a C-MAN or C-LAN aggregates all IPv6 traffic and tunnels to a provider-managed customer edge or 6VPE router, then integration as well as migration becomes smooth. Therefore, it is important for the vendor and the customer to work together in determining the best approach.

Dual VRF Membership per Interface

RFC 2547 for IPv4 recommends one VRF per interface. When running dual stack on a 6VPE, multiple VRF configurations on a single physical or logical interface are required (IPv4 and IPv6). Each VRF instance configuration on a dual-stacked interface forms IPv4 and IPv6 address families. Each address family within VRF runs a VRF-aware routing protocol—such as static routing (static IPv6 unicast routing for IPv6), BGP (BGP with IPv6 enhancements for IPv6), OSPF (OSPFv3 for IPv6), or *Routing Information Protocol* (RIP) (RIPng for IPv6).

MTU Requirements

One important piece of information within the network elements is the capacity of the interface to transfer the size of datagrams. This is known as the *Maximum Transmission Unit* (MTU). The minimum link MTU for IPv4 packets is 68 bytes, whereas for IPv6 the minimum MTU should be 1280 bytes. While designing and planning for IPv6 support, the network elements should be examined along with interfaces and underlying network technologies to ensure the MTU requirements.

Dealing with Link-Locals

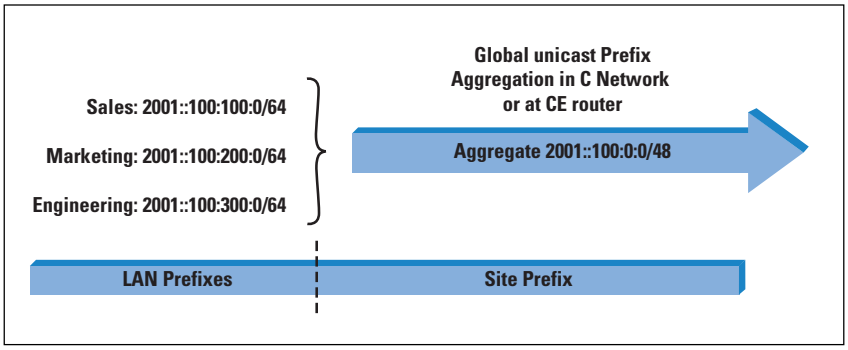
Because link-local scope addresses are defined as uniquely identifying interfaces within a single link, only those may be used on the provider edge-customer edge link.

However, they are not supported for reachability across IPv6 VPN sites and are never advertised with MPBGP to remote provider edges. As outlined in the RFC for IPv6 address assignments, the link locals (**`FE80::x`**) should not be advertised outside their local scope. Because the link-local addresses are embedded on the IPv6-enabled interface for certain local tasks, the link-local addresses are not and should not be advertised anywhere outside the local link scope, including the customer edge and 6VPE running IPv6. Globally unique aggregatable IPv6 prefixes are defined as uniquely identifying interfaces anywhere in the network. These addresses are expected for common use within and across IPv6 VPN sites. They are obviously supported by this IPv6 VPN solution for reachability across IPv6 VPN sites and advertised through MPBGP to remote provider edges.

Router Capacity Impact

Dual-stacking also introduces another task, namely hardware analysis to determine the resource capacity, that is, CPU and memory usage. Increased memory consumption may occur because of the dual-stack *Routing Information Base* (RIB). It also has implications for the *Interface Descriptor Block* (IDB) and *Routing Descriptor Block* (RDB) limits of hardware. The IDB limit is the capacity of particular equipment to support a number of physical and logical interfaces, whereas the RDB limit is the number of routing protocols and instances supported on such equipment. Typically these values (limits) are very high, but 6VPE is such an important element of the MPLS network that these facts must be considered. From a business case perspective, scalability, high aggregation, and rapid Return on Investment are expected, hence it is important to consider these factors in the design.

Figure 4: Route Aggregation



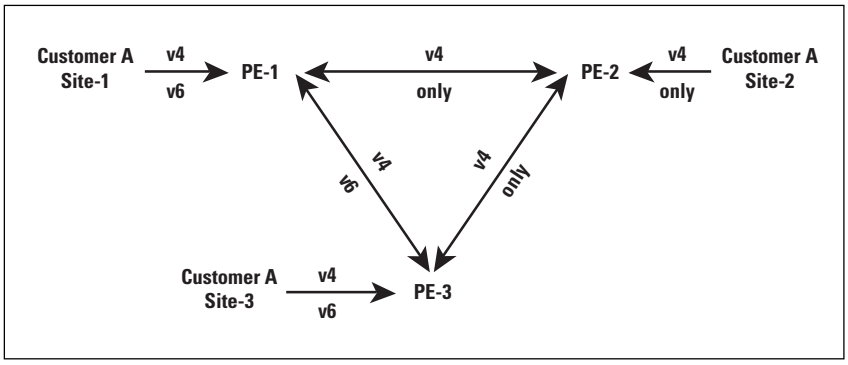
Router Memory Impact

The memory challenges can occur also when large numbers of IPv6 prefixes are advertising toward service provider network elements. In that event, the enterprise on the C-LAN or service provider on the customer edge router may elect to perform route aggregation. IPv6 prefixes can be aggregated to their higher-level significant boundary. Figure 4 shows an example of IPv6 prefix aggregation. Moreover, when a packet arrives on a dual-stacked interface (VRF-aware interface), the 6VPE router determines the packet version number by looking into the IP header. The per-packet header lookup is normally performed (it is a basic router function), but the extra work required by the router is to determine the version number. This additional task creates a longer processing cycle.

The Address Family Identifier and its Importance

All the elements referenced as dual-stacked, such as provider edge and customer edge routers, run IPv4 as well as IPv6 addressing and routing protocols. The 6VPE elements can also mix and match VPNv4 and VPNv6 peering sessions with other 6VPE routers or with route reflectors. What does the term "mix and match" mean here? It was an important enhancement to traditional BGP when MPBGP extensions were introduced. The address family within MPBGP is modular to facilitate distinct peering relationships, and is expressed using the *Address Family Identifier* (AFI). The regular BGP capabilities are exchanged after the peering sessions are turned on. In order for two provider edge routers to exchange labeled IPv6 VPN prefixes, they must use BGP capabilities negotiation to ensure that they both are capable of processing such information. When the service provider network is running VPNv4 peering sessions with other respective elements in the network, it exchanges the VPNv4 AFI capabilities with others. When the VPNv6 peering sessions are turned on, it renegotiates the capabilities and fresh peering sessions are established. The peering sessions established are based on common features if either of the peers does not agree on any of the capabilities.

Figure 5: VPNv4 and VPNv6 AFI



In Figure 5, three provider edge routers out of two need to exchange VPNv6 traffic, but all three provider edge routers need to maintain their existing VPNv4 capabilities. This is possible with the AFI configuration feature, which makes the migration steps very smooth. Service providers can mix and match VPNv4 and VPNv6 provider edge routers as required. Functions of 6VPE can be turned on when and where required. If the customer edge routers are dual-homed to different provider edge routers, the integration of customer IPv4 and IPv6 networks becomes painless. This scenario outlines hybrid environments, but it does not address the IPv4 and IPv6 communication. Consider techniques such as *Network Address Translation* (NAT) or application layer gateways for the IPv4 and IPv6 communication.

Route Reflectors for MP-IBGP

For advertising VPN membership, provider edge routers peer with VPNv4 route reflectors for scalability, thereby avoiding the need for full-mesh MP iBGP sessions among all provider edge routers. The same concept is supported for VPNv6. The same VPNv4 route reflectors can be upgraded to support VPNv6 address families.

Route reflectors can also make addition or removal of a provider edge router from a network simple and flexible. Alternatively, the BGP confederation option can also be deployed to provide MPBGP peering sessions among provider edge routers.

QoS Considerations

Service providers operating customers' MPLS VPN networks and also providing *Quality of Service* (QoS) should account for the new introduction of IPv6 and its impact. QoS and queuing of important application traffic requires distinct policies for IPv4 and IPv6, in turn possibly requiring additional operational tasks where IPv4 and IPv6 networks coexist. Other design considerations should be made to account for each individual network. Both IPv4 and IPv6 have a commonality, which is the 3-bit IP *Precedence* (or *Type-of-Service* [ToS]) field within the IP headers. Alternatively, the *Differentiated Services* (Diff-Serv)-compliant QoS models can also be employed. Irrespective of the technique, QoS is an important factor when low-speed links are concerned. However, there is no additional advantage of QoS on IPv6 versus IPv4. At some point in the future IPv6 can be different by using the flow label in the IPv6 header. QoS within the MPLS core remains *MPLS Experimental Value* (MPLS_EXP)-based and is untouched but still is effective with the addition of IPv6.

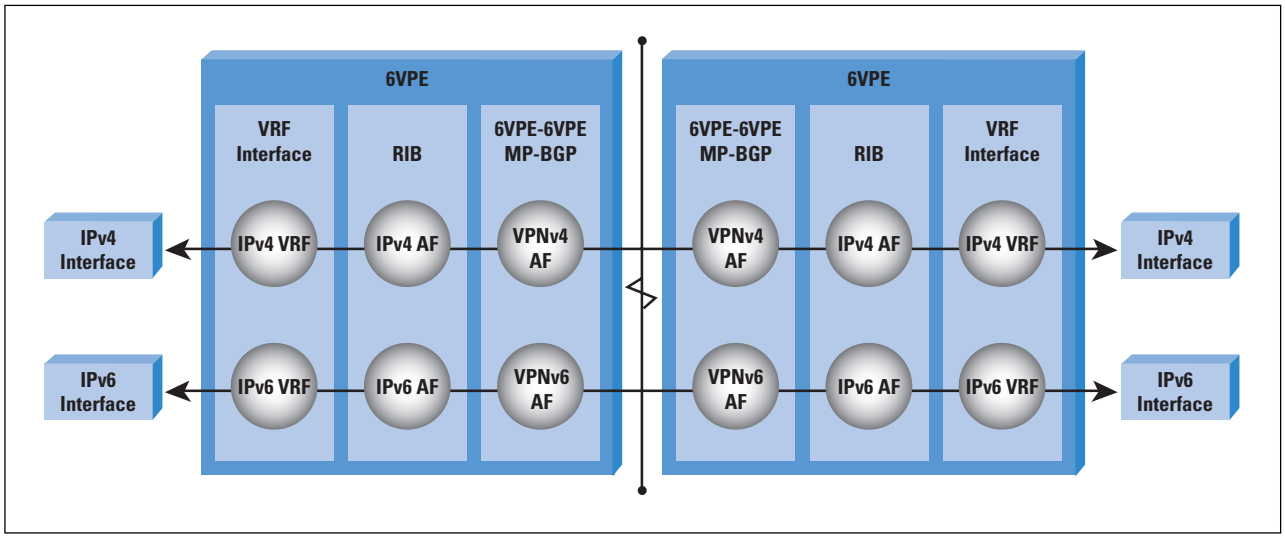
Device Management

Finally, device management is another important aspect that service providers must consider. Device management in a dual-stacked network can be done through an IPv4 or IPv6 address. Where the IPv6 VPN service is supported over an IPv4 backbone, and where the service provider manages the customer edge, the service provider can elect to use IPv6 for communication between the management tool and the customer edge for such management purposes. The management systems, including *Operations-Support-System* (OSS) servers, need to be aware of IPv6 and must run proper *Simple Network Management Protocol* (SNMP) stacks in order to perform IPv6-based management. From the VPN perspective it still remains transparent how the device and services are managed.

Enhancements to the Draft

The current MPLS VPN services that service providers have implemented are based on RFC 2547bis, the Internet Draft required to enhance the Layer 3 VPN approach further to address the IPv6 support. The "BGP-MPLS VPN extension for IPv6 VPN"[1] is the current Draft that addresses the need for IPv6 support over MPLS networks in a VPN environment. Also, to avoid an extra layer of signaling, the Draft addresses the scalable automatic tunneling of VPN-based IPv6 prefixes. The basic functions remain the same as outlined in RFC 2547. Some of the extensions outlined will require additional work in order to be effective in the service provider network.

Figure 6: Dual Mode 6VPE AFI Model



The standard RFC 2547bis introduces "address family" concepts, as well as MPBGP to carry VPN information across the MPLS network. This enables formation of a full mesh between customer sites. The provider edge routers advertise their VPN membership to other provider edge routers through direct iBGP or value(s). As shown in Figure 6, these new address families are introduced to support IPv6 within VPN, IPv6, and VPNv6. If configured for dual stacking, the interface belongs to multiple VRF instances, IPv4 and IPv6. Each instance maintains its own RIB. MPBGP is now capable of handling the VPNv6 address family to advertise the IPv6 prefix across the VPN.

Summary

"Staying abreast of the best" has always been challenging for service providers when it comes to technology deployment or support. Time to market is another challenge. This article provides a view of the service provider challenges. In this new era where explosive use of IPv6 is envisioned, it is extremely important for service providers to have a simplified, automated, fail-proof, and cost-effective network design. The Internet Draft discussed advances the capabilities to achieve this and allows service providers to take a practical approach in supporting IPv6 for customers' next-generation applications. The Draft brings service providers closer to the IPv4-to-IPv6 transition with a simple, cleaner, cheaper, and scalable solution.

For Further Reading

[1] Jeremy De Clercq, Dirk Ooms, Marco Carugi, Francois Le Faucheur, "BGP-MPLS VPN extension for IPv6 VPN," **`draft-ietf-l3vpn-bgp-ipv6.06.txt`**, February 2005.

[2] Eric Rosen and Yakov Rekhter, "BGP/MPLS VPNs," **`draft-ietf-l3vpn-rfc2547bis-03.txt`**, October 2004. (See also RFC 2547, March 1999, by the same authors.)

[3] Mallik Tatipamula, Patrick Grossetete and Hiroshi Esaki, "IPv6 Integration and Coexistence Strategies for Next-Generation Networks," *IEEE Communications Magazine,* Vol. 42, No. 1, January 2004.

[4] Bates, Chandra, Katz, and Rekhter, "Multiprotocol Extensions for BGP4," RFC 2858, June 2000.

[5] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1995.

[6] Rekhter and Rosen, "Carrying Label Information in BGP4," RFC 3107, May 2001.

[7] Carpenter, B. E., Moore, K., Fink, R., "Connecting IPv6 Routing Domains Over the IPv4 Internet," *The Internet Protocol Journal,* Volume 3, No. 1, March 2000.

TEJAS SUTHAR holds CCIE # 8423. He is working as a Service Architect at TELUS Communications Inc. in Toronto. He focuses on Converged Network designs for customers in various industry sectors. He is very active in IP-related deployments. E-mail: **`tejas.suthar@gmail.com`**

Graph Overlays on Path Vector: A Possible Next Step in BGP

by Russ White, Cisco Systems

Over the past several years, much research and thought has gone into a replacement for the current interdomain routing protocol, *Border Gateway Protocol* (BGP)[1]. For instance:

- In 2002, the *Internet Research Task Force* (IRTF) published a set of requirements for a next-generation interdomain routing protocol. In fact, several sets of requirements documents have been published in this area.
- In December 2001, *The Cook Report* noted that BGP needs to be replaced[2]:
- In October 2003, the *Workshop on Internet Routing Evolution and Design* (WIRED) presented papers arguing that BGP needs to be replaced[3].
- In December 2001, the IETF published RFC 3221[5], authored by Geoff Huston, which provided some background information toward finding a replacement for BGP.

There are probably thousands of references in magazine articles, conference proceedings, and research papers, all stating that BGP should be replaced. Of course, all these discussions wind up at the same place: It is almost impossible to replace BGP, wholesale, in the public Internet, or even in any of the private networks running BGP today.

The basic problem is you cannot take the network down, and you cannot replace the routing protocol without taking the network down. Many very clever ideas have been proposed to get around this problem—complex transition schemes, moving partitions, and all sorts of other concepts. But, in the end, the idea of transitioning from one routing protocol to another on something as large—and as distributed in both geography and ownership—as the Internet, has been a hard wall against which all the proposals for new interdomain routing protocols pile up. In an article[4] here in *The Internet Protocol Journal,* Geoff Huston states:

> "Another approach is to consider the feasibility of decoupling the requirements of inter-domain connectivity management with the applications of policy constraints and the issues of sender- and receiver-managed traffic-engineering requirements. Such an approach may use a link-state protocol as a means of maintaining a consistent view of the topology of inter-domain network, and then use some form of overlay protocol to negotiate policy requirements of each Autonomous System, and use a further overlay to support inter-domain traffic-engineering requirements."

In this article, we propose building on this concept, but in a novel way: rather than replacing BGP, or attempting to solve all the currently perceived problems with BGP at once, we attempt to address two problems in a way that does not heavily modify day-to-day BGP operation. Rather than replace BGP, enhance it to account for new re-quirements by providing new capabilities. If done right, this avoids the problem of deploying a new routing protocol altogether, because BGP is already deployed throughout the Internet.

Problems with BGP

No discussion of replacing BGP would be complete without a discussion of why so many people think BGP needs to be replaced. We need to consider three main points in this area: *convergence speed*, *policy*, and *security*. Each of these is covered in the following sections.

BGP Convergence Speed

Through various studies, and through examining the way in which BGP works, it has been shown that BGP, in an interdomain environment, always converges roughly in:

(Maximum AS_PATH – Minimum AS_PATH) × Minimum Advertisement Interval

To understand why this is so, let's examine the following small internetwork as it converges.

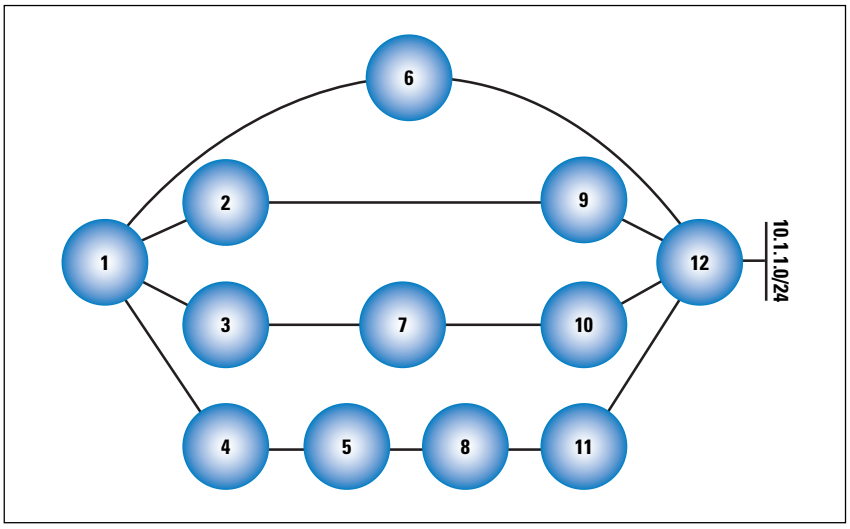


Figure 1: An Example Internetwork Using a Path Vector Protocol

Let's assume autonomous system (AS) 12 is advertising some destination, 10.1.1.0/24, and that every other autonomous system in the internetwork chooses the path to the right to reach that destination. So, for instance, AS4 chooses the path {5,8,11,12} to reach 10.1.1.0/24, AS3 chooses the path {7,10,12} to reach 10.1.1.0/24, AS2 chooses the path {9,12} to reach 10.1.1.0/24, and AS6 chooses the path {12} to reach 10.1.1.0/24.

At this point, let's examine what happens if AS12 loses its connection to 10.1.1.0/24. AS12 sends out a withdraw, which reaches AS6, 9, 10, and 11 at about the same time. These autonomous systems then send out withdraws, with the second set of withdraws reaching AS1, 7, and 8 at about the same time.

When AS1 receives this first withdraw, it examines its local table, and finds the next best path to reach 10.1.1.0/24 is through AS2, with the path {2,9,12}. AS1 does not realize that AS2 has received a withdraw for 10.1.1.0/24 at the same time it received the first withdraw for this destination from AS6. So, AS1 switches over to its next best path, and continues forwarding traffic to 10.1.1.0/24.

AS2, 7, and 8 now also send withdraws to each of their peers, including AS1, 3, and 5. AS1 now receives another withdraw, again for the path it is currently using to reach 10.1.1.0/24. AS1 examines its local tables and finds it has another path, through {3,7,10}, to 10.1.1.0/24, so it switches to that path, without knowing AS3 has just received a withdraw for this same path. AS3 and 5 now send withdraws to each of their peers, AS1 and 4. AS1 has again received a withdraw from the peer it is using to reach 10.1.1.0/24, so it examines its local tables, and finds it still has a path through {4,5,8,11,12} to reach this destination. It switches to this path, without realizing AS4 has just received a withdraw as well.

AS4 now sends the final withdraw to AS1, removing AS1's final path from its local tables. AS1 now removes all reachability information for 10.1.1.0/24, and the network is converged. Note that the actual convergence in this situation would be a bit more complicated, with AS1 sending updates at each stage, and all the other autonomous systems reconverging at each step along the way, but we have used only the simplest set of messages through the network, to illustrate the basic procedure BGP follows when converging.

This short example illustrates why BGP has the convergence characteristics described previously. BGP "hunts" through each possible autonomous-system path, from shorter ones to longer ones, until it finally converges. The rate at which it can hunt through each possible autonomous-system path is determined by the minimum advertisement interval, the rate at which new routing information is allowed to flow through the system.

This problem has several obvious solutions. The first is to simply increase the rate at which routing information flows through the system, by reducing the minimum advertisement interval. But, this plays against route flap dampening, and network stability in general, so, beyond some lower possible bound, reducing the minimum advertisement interval is not possible (without further modifications to BGP).

Another obvious solution is to simply add a "reason code" to the original withdraw. If AS12 originally stated it was withdrawing reachability to 10.1.1.0/24 because it had lost local connectivity to it, then all paths with AS12 in the path could have been discarded immediately, at the first step. The problem here is making certain the original withdraw message actually makes it through the network, from AS12 all the way to AS1. Because BGP is a very efficient protocol, many control messages of this type are actually removed from the network, through implicit withdraws, aggregation, and other mechanisms.

Policy

The second problem we encounter with BGP is its rather rough sense of policy. For instance, let's examine the following small network, and look at one specific example of where policy transmission and enforcement are problematic in BGP.

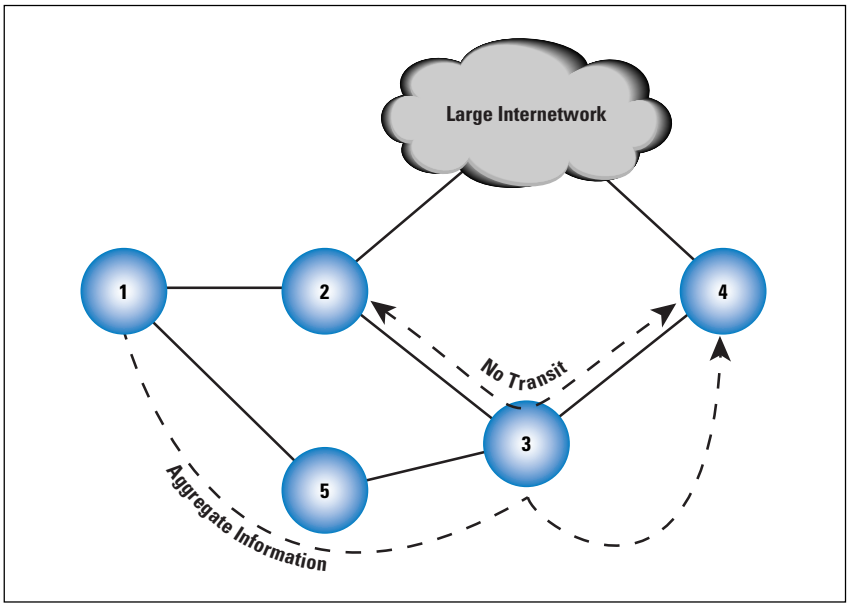


Figure 2: Issues with Policy Transmission in a Path Vector Protocol

Here AS2 has a policy that AS3 should never be used for transit. In other words, traffic originated in AS4 should always pass through the large internetwork rather than through AS3 to reach AS1. This type of situation is very common in the public Internet, such as when AS3 is actually AS2's customer. How can AS2 communicate this policy to AS4, however?

AS2 could simply mark the routing information it sends to AS3 so AS3 cannot readvertise it to AS4, but this is problematic. Simple mechanisms, such as marking the routes with the NO_EXPORT community, are easy for AS3 to simply strip off the routing information it receives. We could conceive of some way to cryptographically sign the included policy, so AS3 cannot disturb the policy and AS4 can see the policy when it receives the information from AS3, but this is problematic as well.

Suppose AS3 is receiving aggregated routing information directly from AS5, which includes some of the same destinations AS2 has advertised to AS3, but has blocked AS3 from advertising to AS4. AS3 could, conceivably, readvertise this routing information to AS4, and AS4 could prefer this shorter prefix aggregate to reach the destinations in AS1, rather than the paths through the large internetwork. AS4 would then forward traffic to AS3, which would then rely on its longer prefix routes, received from AS2, to forward this traffic to these destinations in AS1. AS3 is, contrary to AS2's policy, transiting traffic through AS2 to AS1. There is no simple answer to this problem.

Security

It has been widely acknowledged that BGP is an insecure protocol, with many areas where attackers can hijack, inject false routing information, and perform other attacks. The IETF's *Routing Protocols Security* (RPsec) working group is working on a set of documents describing vulnerabilities of BGP, and creating recommendations for systems to secure BGP. For the latest information about these Drafts, refer to the RPsec homepage at: **`http://www.rpsec.org`**

What sort of requirements are likely to come out of such an undertaking?

- Any proposed mechanism must be able to show that a specific autonomous system is authorized to originate specific routing information.
- Any proposed mechanism must be able to show that the AS Path carried in received routing information corresponds to a real path in the internetwork, beginning with the origin AS and ending in the advertising peer.

There will be many other requirements that proposed mechanisms for providing security for BGP will need to, or should, meet, but these two will be the largest areas of concern for our purposes.

Solving the Problems

Now that we have an idea of the three areas we want to solve problems in, how can we actually solve them? The most elegant solution would be a single mechanism that does not change the current semantics of BGP itself too greatly, would provide greater benefits as it is deployed throughout a large-scale internetwork, and would rely on existing—and understood—techniques within routing.

One perfect example of such a mechanism would be to simply overlay a link state-like graph of interconnectivity over the BGP protocol. This graph would provide information about the interconnections between autonomous systems, rather than between routers, and would be used to convey information about the topology and policies in the internetwork, rather than to find loop-free paths through the internetwork.

Let's go back through our three examples, and see how overlaying an internetwork connection graph would be able to solve some of the problems currently facing BGP.

Convergence Speed

Looking at our small sample internetwork again:

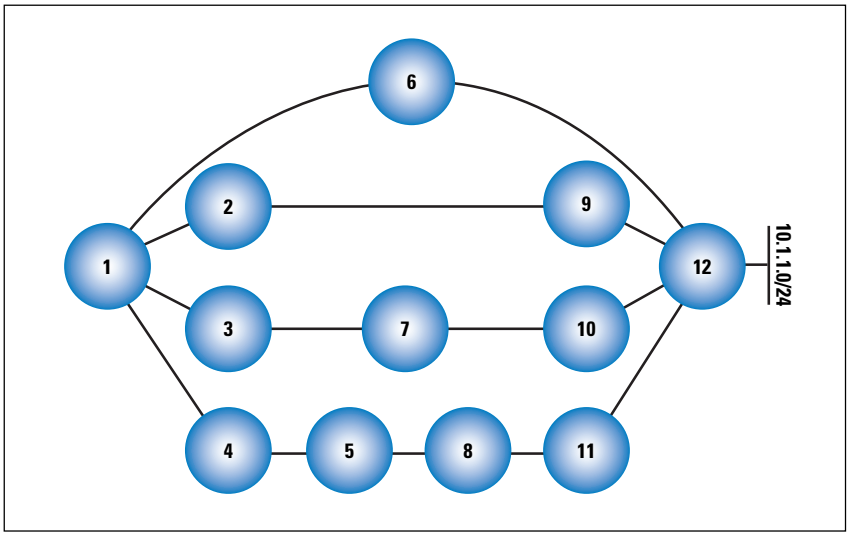


Figure 3: An Example Internetwork Using a Path Vector Protocol

What is the one thing we said would resolve the problems with BGP hunting through every possible longer autonomous-system path alternative to finally converge around loss of reachability to 10.1.1.0/24? Could AS12, somehow, communicate directly to every autonomous system in the internetwork that it has directly lost this connection, rather than waiting for AS1 to try every possible path to 10.1.1.0/24, and discover each one, in turn, withdrawn?

If we had a topological graph of the network, AS12 could simply remove 10.1.1.0/24 from its connectivity information. AS12 would then flood this information, on an interdomain basis, to all the other autonomous systems in the internetwork at roughly the same time. Thus, in the worst case, AS1 would receive this information at about the same time it received the first withdraw for 10.1.1.0/24, from AS6.

When AS1 receives this updated topology information from AS6, it will discover that AS12 is no longer connected to 10.1.1.0/24, and, therefore, it can remove every possible path to 10.1.1.0/24 containing AS12. This would allow AS1 to remove the paths {2,9,12}, {3,7,10,12}, and {4,5,8,11,12} at the same time. The internetwork now converges as soon as AS1 computes the new connectivity graph, and acts on it by examining each entry in its local tables and discarding the ones with AS12 in the autonomous-system path.

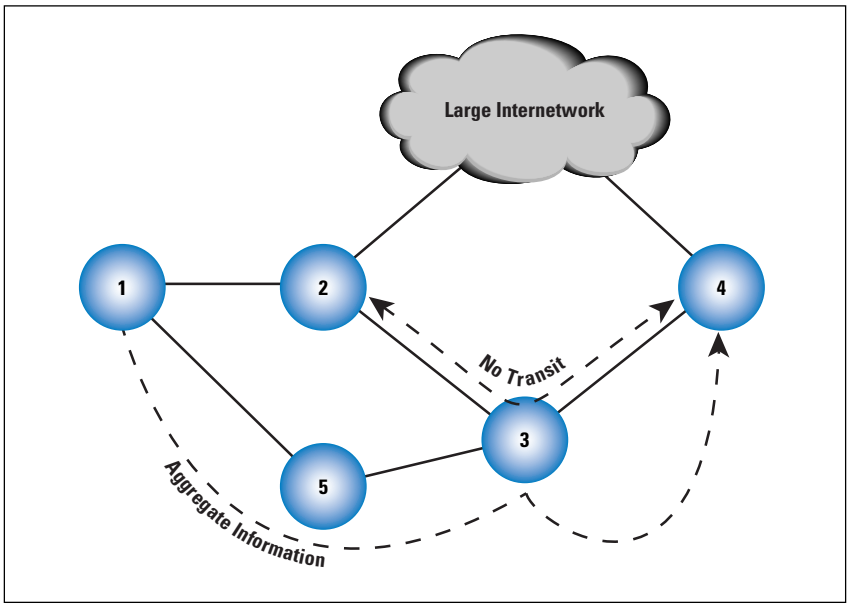
We have not changed the way BGP finds paths through the network—the path still is not valid unless we receive an advertisement from our connected peers. We have also not changed the format of any BGP updates, any peering state machines, or anything else. We have simply overlayed an interconnection graph on top of the current protocol mechanisms, which we can use to our advantage to speed up network convergence.

What about partial deployments in this situation? Suppose only autonomous systems 6, 7, 8, 9, 10, 11, and 12 are running this new extension. Would it still help us to speed up network convergence? When AS12 withdraws 10.1.1.0/24, AS6, 7, 8, 9, 10, and 11 would immediately discard any routes passing through AS12 to reach 10.1.1.0/24. At this point, they could each withdraw those routes, meaning AS1, 2, 3, and 5 would all receive a withdraw at about the same time. This short-circuits the number of possible paths for AS1 to hunt through, decreasing the amount of time the internetwork takes to converge. Even without a full deployment, we see some positive impact from this new technique.

Policy

Let's examine our policy problem after placing our interconnection graph on top of the internetwork.

Figure 4: Injecting Policy on an Interconnection Graph



Here, we see that AS2 could actually place its policy for AS3 not to transit traffic in the interconnection draft. AS4 would then be able to independently verify what AS2's policy toward AS3 transiting traffic is. AS4 could then examine the routing information it receives from AS3, and determine if it should install—or not install—routing information received from AS3, based on this policy.

Objections to an Interconnection Graph

When a link-state protocol has been proposed as a possible replacement for BGP in the past, two primary objections have been raised:

- Providers are reluctant to accept the wholesale replacement of a known working system with a new one.
- Many providers wish to hide their policies and connectivity to other providers or customers for policy reasons.

This article does not propose replacing BGP, just augmenting it, so the first argument is, to some degree, not valid against this approach. The second objection, that of using a link-state protocol for interdomain routing specifically, also does not apply, because we are not proposing changing the way BGP finds loop-free paths through the network. The proposed interconnection graph is not used for finding paths through the network, it is used only for faster signaling of path failure (by short-circuiting the slower withdraw mechanisms), and for providing a place to hang policy and security information.

Concentrating on a few smaller spaces allows us to design a smaller solution set that can be incrementally deployed in a simple way.

The second objection is harder to meet, simply because the concepts of policy within a routing system are hard to define and understand in all possible cases or respects. In fact, there are policy requirements not met by BGP today, but rather are met through contracts, packet filters, and other mechanisms (even sometimes by violating the BGP specification).

Consider two facts about this proposal that work around many of the specific objections we have heard in this area:

- The interconnection graph can be partial, in different parts of the internetwork. For instance, a given service provider might provide different views of who they are connected to to different peers, depending on their policy of revealing this information.
- The interconnection graph only contains autonomous system-level connectivity information, not specific peering-point information. For instance, two autonomous systems may be connected in a large number of places, or as few as one. The interconnectivity graph does not care about such details, only whether at least one connection exists. Such an interconnectivity graph would not reveal actual connection points between peering autonomous systems, how rich that connectivity is, nor any other information about the business relationship between the two peers.

  In fact, the types of interconnectivity information an interconnection graph could provide is already available by examining the autonomous-system paths of routes retrievable from various route view servers. Some mechanism would be required to collate this information into a usable graph, but a good deal of current research on the scaling and convergence properties of large-scale internetworks actually depends on the ability to build an interconnection graph before beginning any other work, so mechanisms to collate this data already exist, and are in use today.

Security

The internetwork interconnection graph can actually show whether a path exists from the origin to the advertising peer, through *signed certificates*. For example, soBGP[6] (**`ftp://ftp-eng.cisco.com/sobgp/index.html`**) uses this specific mechanism to validate the autonomous-system path carried in received routing information. Other research is currently being pursued in this area as well.

Summary

We have proposed a single step forward that could be used to resolve some of the problems facing BGP in the near term, and possibly provide the networking community with a path forward on other fronts as well. The concept of simply making incrementally deployable changes to BGP to solve pressing problems can provide us with options outside the normal lines of thinking: either making very small changes to BGP, making BGP more and more complicated, or simply replacing the BGP protocol, with all the deployment problems this would entail.

References

[1] Yakov Rekhter, Tony Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, March 1995.

[2] **http://www.cookreport.com/10.09.shtml**

[3] **http://www.net.informatik.tu-muenchen.de/wired/position/bruce.html**

[4] Geoff Huston, "Scaling Inter-Domain Routing—A View Forward," *The Internet Protocol Journal,* Volume 4, No. 4, December 2001.

[5] Geoff Huston, "Commentary on Inter-Domain Routing in the Internet," RFC 3221, December 2001.

[6] Russ White, "Securing BGP Through Secure Origin BGP," *The Internet Protocol Journal,* Volume 6, No. 3, September 2003.

RUSS WHITE works for Cisco Systems in the Routing Protocols Deployment and Architecture (DNA) team in Research Triangle Park, North Carolina. He has worked in the Cisco Technical Assistance Center (TAC) and Escalation Team in the past, has coauthored several books on routing protocols, including *Advanced IP Network Design, IS–IS for IP Networks,* and *Inside Cisco IOS Software Architecture*. He is currently in the process of publishing a book on BGP deployment, and is the co-chair of the Routing Protocols Security Working Group within the IETF. E-mail: **riw@cisco.com**

Book Reviews

A Brief History of the Future

A Brief History of the Future—The Origins of the Internet, by John Naughton, ISBN 0-75381-093X, 2000, Published by Phoenix, **http://www.orionbooks.co.uk**

This is a well-written book by a well-known Irish academic and journalist, which charts the growth of the Internet from a 1950s military project to the pervasive networking infrastructure that dominates the IT world today. It is relevant to the readership of this journal because it charts the growth of the technology that underpins the IP world—and it gives a sound understanding of the culture and approach that led to the development of the Internet as we know it.

Naughton takes the reader from the inception of the *Advanced Research Projects Agency Network* (ARPANET) through most of the major developments such as packet switching, mail, TCP/IP, and the Web, not only covering the technology, but also providing insights into the background of the Internet pioneers and the political environment.

Organization

The book is divided into three major sections, the first of which is largely concerned with scene setting and is aimed at bringing those less familiar with the subject area up to speed. In the first chapter, Naughton likens the evolution of the "Net" to that of amateur radio, moving on in succeeding chapters to cover basic technology and to provide some perception of scale and rate of growth.

The second part of the book covers the growth of the Internet up to the early 1990s. This starts by looking at the origins of the ARPA project, noting the influence of MIT and important figures such as Vannevar Bush, Norbert Weiner, and J.C.R Licklider. Naughton describes how ARPA was initiated and its relationship with NASA and academia, highlighting the desire to provide time-sharing systems and the breakthrough concept of the *Interface Message Processor* (IMP) as a solution to the "n-squared" problem. This is followed by two chapters that discuss the adoption of packet switching as the underlying technology, following its initial proposal by Paul Baran and further development by Donald Davies' team in the UK.

Naughton next examines how e-mail became the first "killer application" that drove up Internet usage, even telling the reader where the use of the ubiquitous "@" symbol comes from. He then considers the maturing network during the 1970s, discussing the formulation of the first *Request For Comments* (RFCs), the development of the gateway concept, and the evolution of TCP/IP. The discussion leaves the network area, concentrating on the evolution of UNIX and its impact, stressing the role of AT&T's regulatory situation. Then Naughton considers how this accelerated the development of USENET.

In a chapter called "The Great Unwashed," Naughton discusses the popularization of computing and networking, through the availability of the PC and the evolution of readily available file transfer tools such as X-Modem and the creation of bulletin board systems such as fidonet. He then considers the development of Open Source, telling the story of Linux and its derivation from MINIX.

The third section of the book deals with the emergence of the World Wide Web, tracing it back through the original ideas of Vannevar Bush and Ted Nelson, to its ultimate development by Berners-Lee at CERN. He links this to the subsequent development of Mosaic at NCSA and shows the dramatic impact this had on Internet growth.

Naughton concludes his book by looking at the prognosis for the "Net." Here he refuses to try to predict the future; instead he analyzes the forces that will drive the future of the Internet and discusses their impact in the past and hence their potential impact. At the end of the book, he provides notes and references for each chapter, a short section on the sources he consulted, and a comprehensive glossary.

Synopsis

I found this book provided excellent insights into the development of the Internet, adding a lot of perspective to the engineering field I currently work in. Naughton places appropriate emphasis on the technical, personal, commercial, and political factors that have steered its evolution. He is not afraid to disturb the reader's preconceptions by looking at things from unusual angles, and he emphasises the importance of *timing*. This is apparent when he points out that according to many sources, most of the important inventions around the Internet have come from graduate students, rather than the professors they work for. He similarly recounts the story that AT&T turned down the opportunity to run the "Net" in the early 1970s and reflects the view that if the Internet had not existed we could not invent it now.

This is an excellent read (it was nominated for the *Aventis Prize* in 2000), which helps the reader understand the How, When, Where, and Why of the Internet's development. It covers most of the major milestones in the evolution of our discipline and is very well-written.

The Author

John Naughton is Professor of Public Understanding of Technology at the Open University, and he writes a weekly column in *The Observer* Business Section, covering important developments and trends in the IT industry. He describes himself as a "Control Engineer with a strong interest in systems analysis and computer networks" and is a Fellow of Wolfson College, Cambridge.

—Edward Smith, BT, UK
edward.a.smith@btinternet.com

Eats, Shoots & Leaves

Eats, Shoots & Leaves, by Lynne Truss, ISBN 1-592-40087-6, Gotham Books, 2003.

Eats, Shoots and Leaves is a book about punctuation, but boring it is not. Informative and delightful it is. Lynne Truss includes in the book—which she says is not about grammar—wonderful examples of misused and misplaced punctuation marks. She claims to have written the book to unite us sticklers who do care about the written word, and how we communicate through it. We sticklers cringe with many misuses of punctuation, and we are cringing more and more often it seems.

Truss defines punctuation as a tool to clarify the written word, and who can argue with helps for clarification? She suggests that punctuation is dying, but then asks what would happen without it? Just imagine all the words in the first paragraph with no punctuation marks and no capital letters. You might be able to figure out its meaning with some work, but it would not be easy. Also consider, she suggests, the following:

> A woman, without her man, is nothing.
> A woman: without her, man is nothing.

Punctuation makes all the difference!

The book begins with a discussion of the apostrophe. Meaning "omission," the apostrophe was first used in the 16th century. The most common egregious misuse of this tool is found in the word "it's." It's translates "it is," but it is often used as a possessive word, as in "The keyboard is useless; some of it's keys are missing," when it should be "The keyboard is useless; some of *its* keys are missing." As a test, if you cannot substitute the words "it is" or "it has," it should be "its;" if you can, it is correctly "it's." And the same is true for you're and your. You're translates "you are," and your is the possessive ("It's your turn").

Another amusing example Truss gives is: Member's May Ball. Of course it should be Members' May Ball, because who would just one member dance with? Truss asks.

In her discussion of the comma, we learn that commas were first used 2000 years ago by Greek dramatists to show the actors where to pause or breathe. Then when printing was invented and used increasingly in the 14th and 15th centuries, a Mr. Aldus Manutius (1450–1515) developed italics, the semicolon, the comma, the colon, and full stops (we call them periods in the U.S.).

Truss is a master of the metaphor. She calls the comma the "sheepdog" of words. The comma organizes words, phrases, and groups of words that fit together. Consider one of her comma examples, a properly placed comma: No dogs, please.

Now think about that sentence without the comma: No dogs please. Now consider this: But many dogs *do* please. Thus the importance of the properly placed comma.

Truss addresses all the other marks, including semicolons, quotation marks, brackets, hyphens, parentheses, the four attention-grabbers: *italics*, the exclamation point, the dash —, and the question mark, and finally the ellipsis (the three dots ...). She tells us that, amazingly, someone actually did a PhD thesis on the ellipsis!

One chapter discusses the fact that proper use of punctuation steadily declined in the 20th century, many blaming the decline on television; and that it will continue to decline in the 21st century because of the Internet. E-mail messages cry for brevity, and brevity they get. For example, "**CU B4 8.**" "Netspeak" is, no doubt, here to stay. Language usage also is trending toward the deletion of spaces between words, so that now we say healthcare, chatroom, and the like.

And finally, Truss discusses the newest job that punctuation marks have assumed: emoticons. Examples include the smiley face :–), the sad face :–(, and many others, all made with common punctuation marks.

I thoroughly enjoyed this book, and recommend it to anyone who wants to learn while being entertained. It is a wonderful read.

—Bonnie E. Hupton, Editor
bhupton@sbcglobal.net

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

Paul V. Mockapetris Wins 2005 ACM SIGCOMM Award

Paul V. Mockapetris, Chairman and Chief Scientist at Nominum Inc., is the winner of the 2005 *ACM SIGCOMM Award*. The SIGCOMM Award is widely recognized as the higheset honor in computer networking. The Award recognizes lifetime achievement in and contributions to the field. It is awarded annually to a person whose work, over the course of his or her career, represents a significant contribution to the field and a substantial influence on the work and perceptions of others in the field. The SIGCOMM Award is presented to Dr. Mockapetris "in recognition of his foundational work in designing, developing and deploying the *Domain Name System* (DNS), and his sustained leadership in overall Internet architecture development."

Paul Mockapetris created the original DNS protocol, wrote its first implementation, and worked with others to spread the DNS across the Internet. The design of DNS, which was the first major datagram protocol of the Internet, established a number of principles for key Internet infrastructural services. Its simplicity of design and fitness for purpose have stood the test of time. The strength of its design lies in a novel combination of hierarchy and caching that gives each organization absolute control over part of the namespace while simultaneously relying on caching to make the entire system efficient. Its success can be seen from the fact that DNS now handles many orders of magnitude more names and traffic than when it was first deployed, and yet the design and structure have remained intact. As a result the DNS design and caching mechanisms are often cited as two of the cornerstones on which the success of the Internet is built.

In addition to his work on DNS, Dr. Mockapetris' career has included pioneering work on multiprocessor operating systems, virtual machines, and ring LAN technology. Further, Dr. Mockapetris played an important role in the deployment of networking technologies internationally. Starting during 1990–1993 as a program manager at ARPA, Dr. Mockapetris fostered the international deployment of multimedia conferencing, multicast, and QoS. His strong leadership in development of Internet architecture continued as Chair of the Internet Engineering Task Force during 1994–1996, as member of the Internet Architecture Board during 1994–1996, and then as member of the Federal Networking Council. Dr. Mockapetris is also a recipient of the *IEEE Internet Award* and is an ACM Fellow.

In summary, through his sustained effort in support of the Internet archicture, beginning with DNS and continuing through work at ARPA, IETF, and industry, Dr. Mockapetris has made far-reaching and influential contributions to computer networking. The 2005 SIGCOMM award recognizes Dr. Mockapetris for this lifetime record of achievement.

SIGCOMM is the *Special Interest Group* (SIG) on *Data Communication* of the *Association for Computing Machinery* (ACM). SIGCOMM is a professional forum for the discussion of topics in the field of communications and computer networks, including technical design and engi-neering, regulation and operations, and the social implications of computer networking. The SIG's members are particularly interested in the systems engineering and architectural questions of communication. For more information please visit: **http://www.acm.org/sigcomm/**

Voice over IP (VoIP) And Government Policy

Voice over IP technology has the potential to provide much cheaper telephone service, particularly internationally. More importantly, it can enable exciting new services, such as voice-enabled Web pages and integrated phone, voice-mail, and e-mail. Unfortunately, some national governments are trying to limit its use. In late April, 2005, the *Advisory Committee on International Communications and Information Policy* (ACICIP) of the U.S. Department of State issued a very useful paper describing how VoIP works, the benefits it can provide, and what governments around the world are doing to promote or hinder its development.

Michael Nelson, the Internet Society's Vice President for Policy, represents ISOC on the Committee, and is helping draft "Version 2.0" of the paper, which will report on recent developments in additional countries. If you would like to make suggestions about the paper, please submit them to Michael Nelson at **mnelson@isoc.org**

For more information, see:
http://isoc.org/pubpolpillar/voip-paper.shtml

ISOC Commentary on the Status of the Work of WGIG, April 2005

When the first phase of the *World Summit on the Information Society* (WSIS) called on the UN Secretary General to set up the *Working Group on Internet Governance* (WGIG), it was in the context of supporting the WSIS *Action Plan*. The Plan calls for concrete actions to advance the achievement of internationally agreed development goals by promoting the use of ICT-based products, networks, services and applications, and to help countries overcome the digital divide. This is, by the way, something the Internet community has worked hard to achieve since the very first days of the Internet.

These goals include those described in the *Millennium Declaration*. The 8th goal of that document is to develop a global partnership for development, which would make available the benefits of new technologies—especially information and communications technologies—in cooperation with the private sector for the benefit of all. This is the context (making the benefits of ICT available to everyone) in which we initially engaged in the WSIS and WGIG efforts. The Internet has a huge potential as an enabler bringing these benefits to people everywhere and we remain excited about the WSIS mission. However, it is not clear how WGIG's actions to date have helped support achieving such goals.

The *Internet Society* (ISOC) believes that the best way to extend the reach of the Internet is to build on those aspects that have worked well, for example, the long established open, distributed, consensus-based processes and many regional forums for the development and administration of the Internet infrastructure. Decision-making about issues such as resource allocation or IP Address Policy has always been in the hands of the Internet community, in order to be as close to those who require and use the resources as possible. It is this participative model, close to the end users, that led to the phenomenal, stable growth of the Internet. The Internet community and its bottom-up processes are constantly evolving in response to changes in needs and availability. For example, in response to moves by the African Internet community, the African countries now have their own *Regional Internet Registry* [RIR] (AfriNIC) that helps coordinate users' needs and IP Policy in that region. Latin America has the same story to tell. Support for the development of both these RIRs (educational, financial and boot-strapping of various processes) came from the global Internet community and primarily came from the other RIRs.

Developing and maintaining the Internet infrastructure are just two aspects of what has come to be referred to as *Internet Governance*. WGIG has pointed out that there are many others, and has recognized the fact that Internet Governance encompasses a much wider range of topics than IP address and domain name administration. However, much of WGIG's focus has been on Internet infrastructure, thereby missing an opportunity to focus on those aspects of the Internet's development that are less developed and that could benefit from improved, lightweight mechanisms facilitating an exchange of information between policymakers and the Internet community. Examples here are issues con-cerning inappropriate usage of the Internet—cybercrime and spam being just two examples. Much work has already been done on technical solutions to these issues, and many legal frameworks already exist for handling criminal activity such as fraud. The challenge today is to bring the lawmakers and policymakers together with the Internet community to discuss the most appropriate mechanisms to ensure the continued development of the Internet.

Many players have a role, and this clearly includes governments and intergovernmental organizations. WGIG had a clear mandate to not only develop a working definition of Internet Governance, but also to develop a common understanding of the respective roles and responsibilities of governments, existing intergovernmental and international organizations and other forums, as well as the private sector and civil society encompassing both developing and developed countries. Unfortunately an inordinate amount of time has been spent focusing on challenging current structures (those that brought us the Internet and its rapid, stable growth), rather than looking forward to the potential benefits of extended cooperation with (and based on the proven success of) existing models and structures. WGIG seems to have lost sight of this larger goal.

Also, many of WGIG's premises seem to start with an assumption that the Internet needs a hierarchical top-down governance model, thereby ignoring the decentralized, distributed structure on which the Internet was so successfully built. Not only does this "governance hierarchy" model prevent an accurate understanding of the Internet's infrastructure and development (forcing key organizations to be classed in prescribed categories that do not fit with the reality of their actions or their role in developing and supporting the Internet) but it also will very likely lead to conclusions that will harm the Internet's development and growth.

While WGIG appears to ascribe the growth of the Internet to deliberate regulatory decisions to liberalize telecommunications, in reality regulatory measures have been a relatively small factor. A more significant factor in the growth of the Internet has been the fact that the Internet architecture has enabled many tens of thousands of users to develop their own applications independent of the underlying architecture, thereby empowering people to add true value to the global Internet network. The continued expansion of the Internet to developing countries though will be greatly aided in the future by a more competitive telecommunications environment. We urge WGIG to recommend more concrete and aggressive action in this direction.

Further, WGIG has put great focus on comparing the relative merits of established treaty bodies and intergovernmental organizations to undertake a central role in the development of Internet infrastructure while very largely overlooking areas where attention and support are required and where national governments more naturally have a role to play, areas such as misuse of the Internet (cybercrime and spam to name a few). The limited perspective of this approach displays an obvious bias in the characterization of the issues and seems to pre-suppose a solution. In conclusion, we would urge WGIG to spend more time looking at what is actually being done to enable more people around the world to take greater advantage of the power of the Internet. This includes a focus on the many regional and global education activities that different Internet-related organizations are undertaking to "connect the unconnected."

These same organizations are also working to make the Internet more secure, more accessible, more reliable, more affordable, and more versatile. The development of the Internet, as well as many well-established capacity-building efforts could be jeopardized by applying a too heavy-handed approach to the operation and administration of this unique network of networks. Decentralized, lightweight governance has clearly proven itself to be a positive feature not a weakness. We want to encourage WGIG and WSIS to work with the Internet community within the already well-established Internet model to improve co-operation between policy makers and the Internet community.

In the spirit of meeting the international development goals highlighted by WSIS, any review of today's Internet model or structures must be carried out in the context of how well they have worked in the past, how well they meet the needs of the people who depend upon them today, and how well they will adapt to changing requirements in the future; and not simply focus on a comparison to other historical telecommunications or governance models. These historical models have not been demonstrated to be well suited to the Internet. For more information, see:

http://isoc.org/

http://wgig.org/

http://www.itu.int/wsis/

An interview with the new IETF Chair

IBM Distinguished Engineer and former ISOC Chairman Dr. Brian Carpenter has just taken over the role of IETF Chair. In a recent interview, Brian describes the future challenges facing the IETF and the Internet in general. The full interview is available here:
http://resources.isoc.org/20503

Upcoming Events

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in Luxembourg City, Luxembourg, July 11–15, 2005 and in Vancouver, Canada November 30–December 4, 2005. For more information see: **http://www.icann.org**

The *South Asian Network Operators Group* (SANOG) will meet in Thimpu, Bhutan, July 16–23, 2005. More info at:
http://www.sanog.org

The *Internet Engineering Task Force* (m nn) will meet in Paris, France, July 30–August 5, 2005 and in Vancouver, Canada, November 6–11, 2005. For more information, visit: **http://ietf.org**

ACM's *SIGCOMM 2005* will be held in Philadelphia, PA, August 22–26, 2005. For more information visit:
http://www.acm.org/sigs/sigcomm/sigcomm2005

The *North American Network Operators' Group* (NANOG) will meet in Los Angeles, October 23–25, 2005. For more information see:
http://nanog.org

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

September 2005 Volume 8, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Protocol transitions are never easy, particularly not when they involve something so fundamental as the *Internet Protocol* (IP). Organizations considering a move to IPv6 must consider many factors when deciding on the timing for such a deployment. One of the first questions that arises is: "When will the IPv4 address space actually run out, forcing us to use IPv6 instead ?" That question is not a new one; it was being asked in the early 1990s when the IPv6 effort was started. Several factors, such as the deployment of *Classless Interdomain Routing* (CIDR) and *Network Address Translation* (NAT), have "delayed the inevitable," and perhaps led to some complacency on the part of network operators. In this issue we examine the topic of IPv4 address space depletion in more detail. Our main article is by Tony Hain, and it is followed by a response from Geoff Huston and a roundtable discussion with Tony, Geoff, Fred Baker, and John Klensin. We would also like to hear from our readers on this important topic. Please send your comments to **ipj@cisco.com**.

As an old-time network and UNIX user, I am a big fan of tools that allow simple terminal access to remote host computers. My "Internet career" started in Norway in 1976, where I used *Telnet* to access machines in California through the ARPANET. Today, I still access remote servers through a simple terminal interface, but Telnet has been replaced by the *Secure Shell* (SSH) *Protocol* for all the obvious security reasons. SSH is used not just for terminal traffic—it also can be configured to provide secure tunnels to a variety of services such as Webpages and file transfers. Ronnie Angello explains the details in our second article.

In order to better serve our readers, we will be conducting an IPJ Reader Survey in the near future. Details will be available on our Website at **www.cisco.com/ipj**. We appreciate your cooperation in completing the survey.

Finally, let me remind you to visit the IPJ Website and update or renew your subscription.

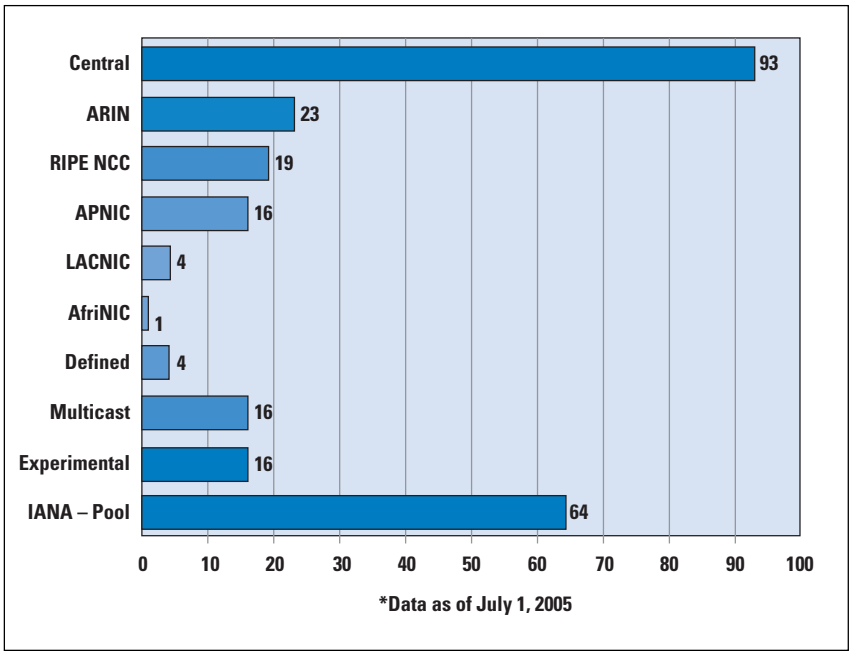
—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

A Pragmatic Report on IPv4 Address Space Consumption

by Tony Hain, Cisco Systems

When I interact with people from all around the world discussing IPv6, there continue to be questions about the projected lifetime for IPv4. This article presents consumption rate and lifetime projections based on publicly available *Internet Assigned Numbers Authority* (IANA) data. In addition, there is discussion about why the widely quoted alternative projection may be flawed, thus leading everyone to believe we have much more time than we might.

Figure 1: IANA /8 Allocations



Allocations

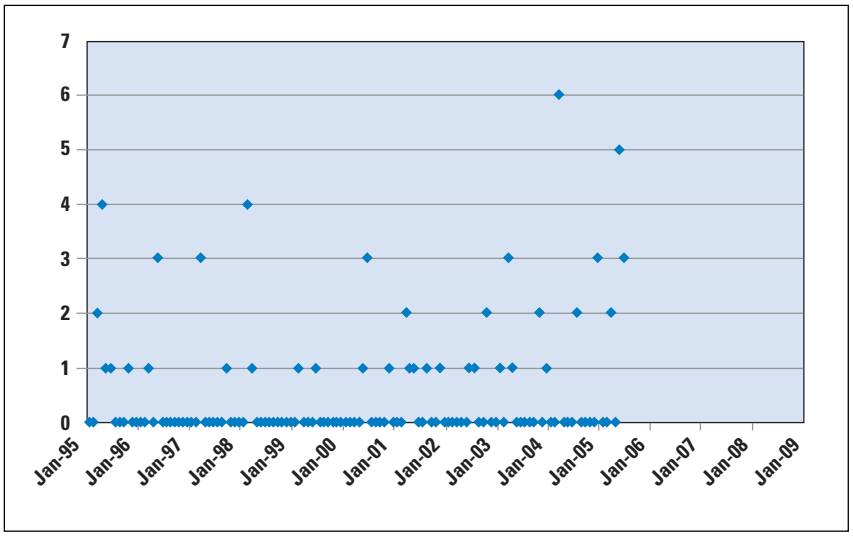
The chart in Figure 1 shows the distribution of all 256 IANA /8 allocation units in IPv4[1] as of July 1, 2005. The Central registry represents the allocations made prior to the formation of the *Regional Internet Registries* (RIRs). ARIN (North America)[2], RIPE NCC (Europe)[3], APNIC (Asia/Pacific)[4], LACNIC (Latin America)[5], and AfriNIC (Africa)[6] are the organizations managing registrations for each of their respective regions. RFC 3330[7] discusses the state of the Defined and Multicast address blocks. The Experimental block (also known as *Class E*—RFC 1700[8]) was reserved, and many widely deployed IPv4 stacks considered its use to be a configuration error. The bottom bar shows the remaining useful global IPv4 pool. To be clear, when the IANA pool is exhausted there will still be space in each of the RIR pools, but by current policy[9] that space is expected to be only enough to last each RIR between 12 and 18 months.

The projection published at **`http://bgp.potaroo.net/ipv4`**[10] is often quoted as the definitive reference for IPv4 consumption. This report presents a viewpoint consistent with that author's long-standing position that we do not need to change from IPv4 to IPv6 anytime soon, thus showing an extended lifetime for IPv4.

The approach used in the potaroo report is to take the simple exponential fit to the allocation data since 1995. As discussed later in this article, this approach includes the effects of the policy shift to *Classless Interdomain Routing* (CIDR) and subsequent digestion of prior allocations, the lull in IANA allocations to the RIRs for two full years, as well as the fact that the model used does not generate a particularly close fit to the actual run rate over the 10-year period.

Although this author agrees that over very long timeframes (20–50 years) there will be substantial variations in the consumption rate for any number of reasons, the opportunity for events that would reduce the recent rate in the timeframe of the remaining IANA IPv4 pool is not evident. That said, there are numerous things that could increase the consumption rate and exhaust the pool even sooner than this projection.

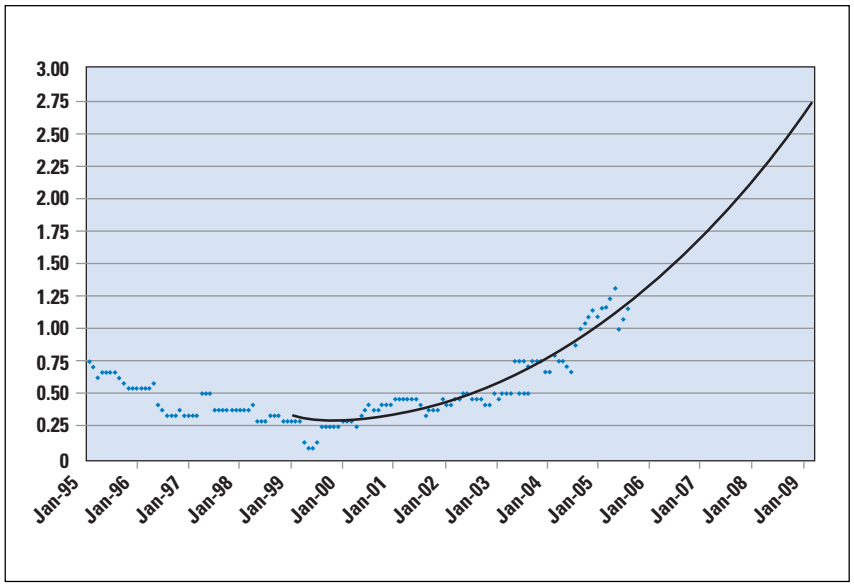
Figure 2: IANA Allocations to RIRs — Raw /8 Allocations per Month



The graph in Figure 2 shows the raw per-month IANA allocations since 1995. In raw form it is difficult to discern the trend, or develop an expectation about the overall lifetime of the remaining pool.

Taking a closer look at Figure 3, smoothing the data with a 24-month sliding window (averaging over 12 months back and 12 months forward) exposes the underlying reality that the combined rate and quantity of /8 allocations has been steadily accelerating since 2000 (the graphs for 12-, 18-, and 24-month sliding windows show the same fundamental trend). Though a few of the allocations may arguably have been "one-time" events, those are lost as statistically insignificant in the extended and continuing overall growth rate.

Figure 3: IANA Allocations to RIRs — Sliding-Window 24-Month Average



Taken by itself, the most recent allocation rate (22 /8s over the 18 months leading up to July 1, 2005) suggests that the remaining pool of 64 /8s will be exhausted in about 5 years, even if growth abruptly flattens out to hold around 1 /8 per month. Unfortunately at this point there is no reason to believe the allocation rates will slow or that they will turn downward again. All the gain of CIDR absorbing the pre-1995 allocations has already been incorporated, and there is no obvious economic bubble that might burst to lower demand within the time window of the remaining pool.

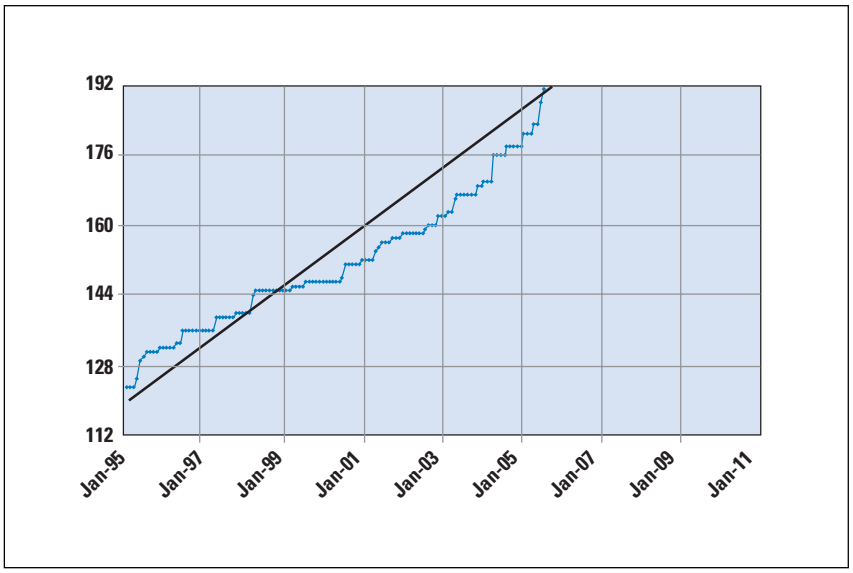
To the contrary, the following URL shows potential demand (to bring developing countries up to just 20-percent connectivity, which is half of what the existing Internet world enjoys today) that will swamp the remaining pool, even in the face of much stricter allocation policies.
`http://www.nav6tf.org/documents/e-Nations-data.pdf`

So this view of the sustained trend in allocation growth rate suggests that the lifetime of the remaining central IPv4 pool is 4 years +/-1.

Projections

Differing from recent articles and section 5 of the report at **`http://bgp.potaroo.net/ipv4`** that hint at linearity in growth, Figure 4 shows that the raw data after 1995 is clearly nonlinear. It starts with a decelerating rate through mid-1998 as the pre-1995 allocations were absorbed (precipitated by the allocation policy shift from class-based to CIDR), followed by a 2-year lull (only 1 /8 per year), then a return to accelerating growth from mid-2000 onward.

Figure 4: IPv4 Lifetime Projection — Non-Linear Nature of Raw Data



This suggests that using the past 10-year IANA data is likely to skew the projection toward a much longer period than the recent allocation data would support. Although a longer lifetime projection helps to avoid short-term panic, it can mislead people into believing there is substantial time to worry about this later, resulting in a much bigger problem when reality blindsides everyone sooner than they expected.

Figure 5: IPv4 Lifetime Projections — Order-N Polynomials, Post-2000 History Basis

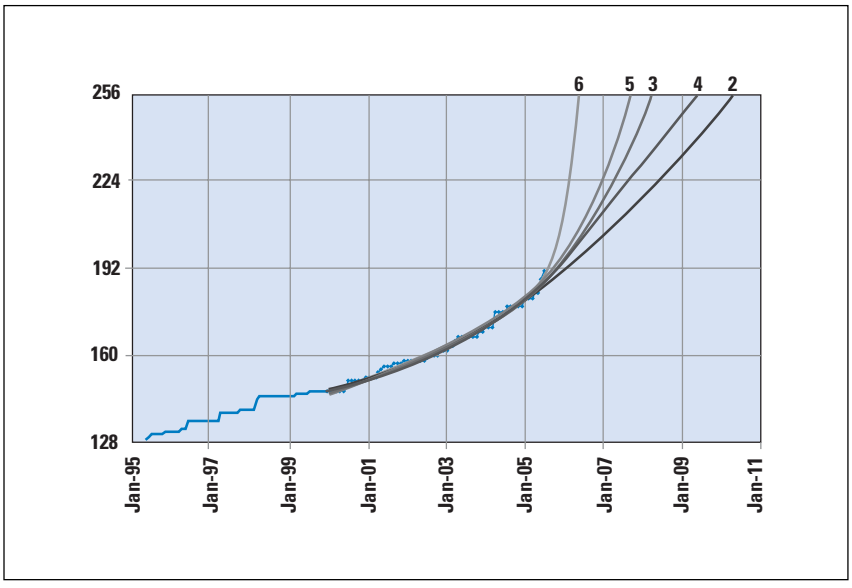
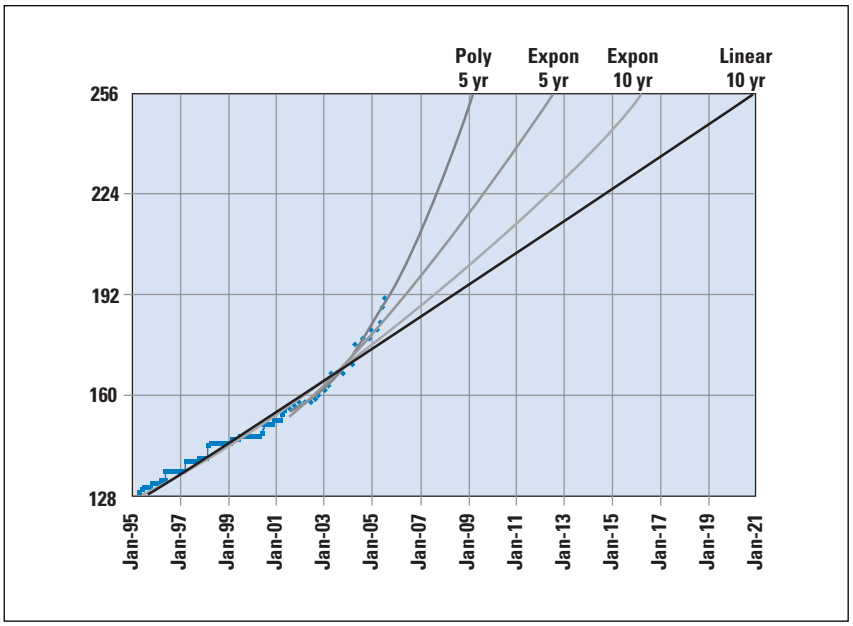


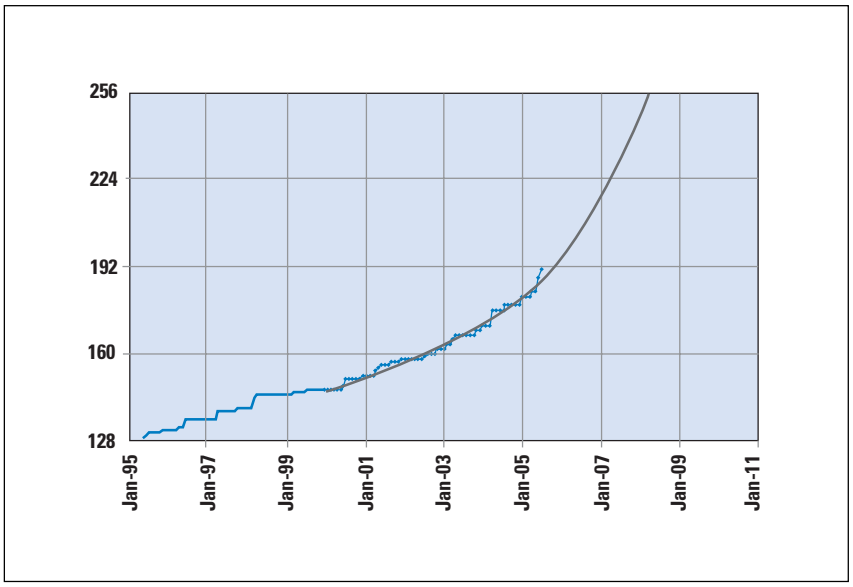
Figure 6: IPv4 Lifetime Projections — Polynomials and Exponentials



As in any statistical endeavor there are many ways to evaluate the data. The various projections in Figures 5 and 6 show different mathematical models applied to the same raw data. Depending on the model chosen, the nonlinear historical trends in Figure 6 covering the last 5- and 10-year data show that the remaining 64 /8s will be allocated somewhere between 2009 and 2016, with no change in policy or demand (though as discussed previously there are already reasons to err toward 5-year-based nonlinear models).

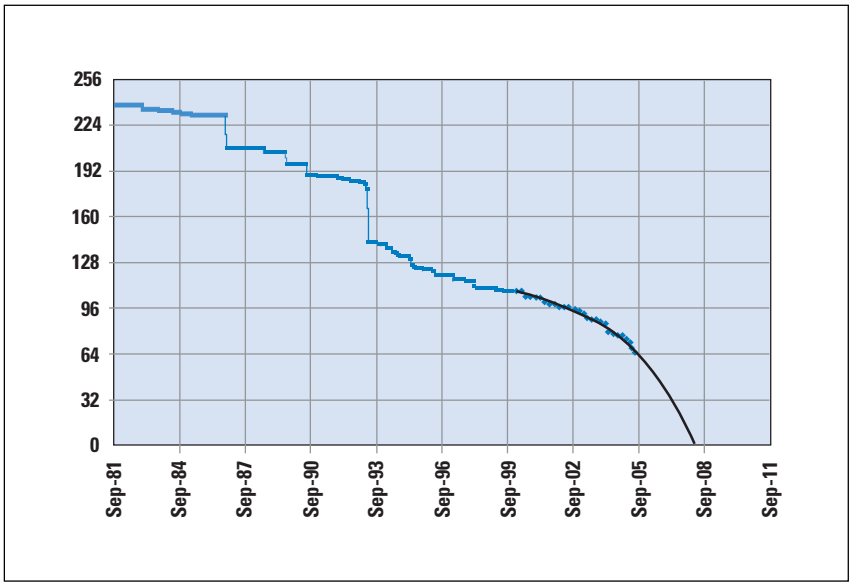
Adding to that, policy is continually changing. ARIN, for example, has recently clarified its policy allowing organizations that demonstrate they have exceeded the capacity of the private space defined in RFC 1918 to acquire IPv4 address blocks from the remaining public pool, even when it is clear these allocations will never be announced to the global Internet. The other regions already have similar policies or are likely to follow suit because the most vocal members of the RIR community have adamantly commented against expanding the private IPv4 range. This policy approach coupled with persistent demand means the actual run rate is going to continue increasing as the large organizations begin consuming public space where they had been using private to support their network growth. For example, one large enterprise has steady growth over 1 percent per month, which currently requires an efficiently managed /12 per year for its expanding network. The enterprise is less than a year from exhausting all the space provided in RFC 1918, so it was very interested in the ARIN policy that allows the enterprise to continue growing through public space. Additionally, multiple commercial service providers expect to reach the capacity of the 1918 space within 12 to 18 months, just supporting management addresses on their existing devices. This does not take into consideration their pending deployment of new services, which they expect will use several new IPv4 addresses per device with marketing targets measured in multiple millions of units.

Figure 7: IPv4 Lifetime Projection — 5-Year History Basis



The graph in Figure 7 hints at the likely outcome as word spreads about the perception of policy liberalization and the demonstrable exhaustion of the remaining global IPv4 pool landing within the *return-on-investment* (ROI) period for new equipment. It is based on the same raw historical data as the frequently quoted long-term projection on potaroo's Figure 2.4, but the more aggressive fit on the most recent data set describes a significantly higher consumption rate and shorter lifetime for the remaining pool.

Figure 8: IPv4 /8 Pool — 5-Year History-Based Projection



The graph in Figure 8 provides the exhaustion perspective, showing the entire address pool from the publication of IP Version 4[11] (note that data prior to 1995 is accurate as to where it was allocated, but with very coarse granularity as to exactly when). The projection curve is based on the IANA allocations from January 2000 onward.

Only time will tell which projection is correct, but it will already take a fairly significant stalling event to slow consumption and put the actual allocation curve back on the extended track in potaroo's Figure 2.4.

Reserved Space

There are occasionally arguments that the 16 /8s reserved in the experimental space could be used. Although this is likely to be possible for some IP stack implementations, for others it is not. At a minimum, some quick tests show that Windows 95 through Windows 2003 Server systems consider that block to be a configuration error and refuse to accept it. The operational ability to restrict the space to a select stack implementation is limited, and the amount of space there does not really help even if deployment and operations were trivial. Assuming the sustained growth trend in allocations continues, by the time the remaining 64 /8s in the IANA pool are finished the rate would be approaching 3 /8 allocations per month, so the entirety of the old Class E space would amount to about 6 months of run rate.

Reclaiming Allocations

Another debate occasionally resurfaces about reclaiming some of the early allocations to further extend the lifetime of IPv4. Hopefully this article has shown that the ROI for that approach is going to be extremely low. Discussions around the Internet community show there is an expectation that it will take several years of substantive negotiation (in multiple court systems around the globe) to retrieve any /8s. Then following that effort and expense, the likelihood of even getting back more than a few /8 blocks is very low. Following the allocation growth trend, after several years of litigation the result is likely to be just a few months of additional resource added to the pool—and possibly not even a whole month. All this assumes IANA does not completely run out before getting any back, because running out would result in pent-up demand that could immediately exhaust any returns.

Summary

Network Address Translation (NAT) and CIDR did their jobs and bought the 10 years needed to get IPv6 standards and products developed. Now is the time to recognize the end to sustainable growth of the IPv4-based Internet has arrived and that it is time to move on. IPv6 is ready as the successor, so the gating issue is attitude. When CIOs make firm decisions to deploy IPv6, the process is fairly straightforward. Staff will need to be trained, management tools will need to be enhanced, routers and operating systems will need to be updated, and IPv6-enabled versions of applications will need to be deployed. All these steps will take time—in many cases multiple years. The point of this article has been to show that the recent consumption rates of IPv4 will not be sustainable from the central pool beyond this decade, so organizations would be wise to start the process of planning for an IPv6 deployment now. Those who delay may find that the IANA pool for IPv4 has run dry before they have completed their move to IPv6. Although that may not be a problem for most, organizations that need to acquire additional IPv4 space to continue growing during the transition could be out of luck.

References

[1] **http://www.iana.org/assignments/ipv4-address-space**

[2] **http://www.arin.net/**

[3] **http://www.ripe.net/**

[4] **http://www.apnic.net/**

[5] **http://www.lacnic.net/**

[6] **http://www.afrinic.net/**

[7] **http://www.rfc-editor.org/rfc/rfc3330.txt**

[8] **http://www.rfc-editor.org/rfc/rfc1700.txt**

[9] **http://www.rfc-editor.org/rfc/rfc2050.txt**

[10] **http://bgp.potaroo.net/ipv4**

[11] **http://www.rfc-editor.org/rfc/rfc791.txt**

[12] Geoff Huston, "The Myth of IPv6," *The Internet Protocol Journal,* Volume 6, No. 2, June 2003.

[13] Geoff Huston, "IPv4: How long do we have?," *The Internet Protocol Journal,* Volume 6, No. 4, December 2003.

Another Perspective

Ed.: We asked Geoff Huston to provide some feedback on this article and he responded with the following:

Dear Editor,

There are, of course, many ways to undertake predictions, and over the millennia humanity has explored a wide diversity of them. In every case the challenge is to make predictions that end up being closely correlated to the unfolding story, and of course hindsight is always the harshest judge of such predictions.

Tony's work takes a different base point for making the projection from earlier work that I did in this area. Tony looks at the rate of allocation from the IANA to the RIRs, and bases his predictions on the trends visible in that time series of data. By contrast, I used the assumption that assigned addresses are destined for use in the public IPv4 Internet, and I used the trends visible in the amount of advertised address space as the basis for the predictions of consumption.

One of the more interesting data artifacts is the first-order differential of the rate at which the span of addresses announced in the IPv4 public Internet has increased over time.
(Figure 4.4 of **http://bgp.potaroo.net/ipv4/**)

One interpretation of this data is that there are two phases of recent activity: prior to March 2003 and post-March 2003. Prior to March 2003 the longer-term address growth rate was the equivalent of some 3.5 /8 blocks per year.

Post-March 2003 we see a different consumption growth rate, fluctuating between 5 and 8 /8s per year, with a mean value of some 7.5 /8s per year. There is no strongly obvious longer-term compound growth rate visible in this view of the data. Given some 64 /8s remaining in the IANA pool as of July 2005 and a base consumption rate of a mean of 7.5 /8s per year, the simple division yields 8.5 years, or 2014 as the time of forecast exhaustion of the IANA address pool. At that point the RIRs will be holding about 25 /8 blocks in their unallocated pools, and a further two years of allocations could be made from these pools.

So I would offer the view that the post-2003 data offers a perspective of exhaustion of the unallocated address pools in 2016, with the caveat that such a prediction assumes that the current address demand levels will continue, the actions of industry players are invariant, and the current address allocation policies will continue as they are at present.

Of course these three caveats represent relatively major assumptions about the future—and are perhaps unlikely to happen. It is likely that there will be changes in all these factors in the coming years, and these will obviously impact these predictive models.

To summarize, I observe that these different predictive approaches yield slightly different outcomes, but not beyond any reasonable error margin for predictions of this nature. Sometime in the forthcoming 5 to 10 years the current address distribution policy framework for IPv4 will no longer be sustainable for the current industry address consumption model because of effective exhaustion of the unallocated address pool.

When looking at this prediction from the perspective of the service provider enterprise, the prediction can be re-expressed as a problem relating to investment lifecycles. The ISP industry and the enterprise sector have already made considerable investments in IPv4-based infrastructure in equipment, infrastructure, and operational capability, and we are seeing some considerable reluctance to add to this with additional investment into IPv6 capability at this time. The direction of the use of various forms of NAT-based approaches and increasing use of application layer gateways in the public and enterprise environments can be seen as an effort to extend the lifetime of the existing infrastructure investment. In a volume-based market with relatively low revenue margins, this position certainly has some sound rationale from a business management perspective. But I agree with Tony here that such business approaches are ultimately short-term in nature, because they do not allow IPv4 to encompass indefinite further decades of Internet growth in a silicon-dense world.

However, in terms of understanding the next few years of a process of industry transition of protocol infrastructure into IPv6 deployment, perhaps the real issues here are more centered on competitive business factors and sector investment profiles than they are about detailed introspection of trends within various number series.

The numbers all indicate that this is not a matter that can be deferred indefinitely. Tony's call for some timely attention to the need to commence investment in IPv6-based service infrastructure is one that I hope the industry is listening to attentively.

—Geoff Huston
gih@apnic.net

A Virtual Roundtable

Ole: Let's open this discussion on the point of measurement methods. We invited John Klensin and Fred Baker to join Geoff and Tony in the discussion at our virtual round table. (We often all see each other at IETF meetings, but there is seldom enough time to gather everyone around a real table, hence this discussion took place with a few rounds of e-mail).

Geoff: As I said in my response letter, Tony's work takes a different base point for making the projection from the earlier work that I did in this area. My work has focused on the trends from the addresses used in the public IPv4 Internet, and then deriving projections on consumption based on this data. It assumes that the influencing factor for address consumption is the use of addresses in the public IPv4 Internet.

Tony: As Geoff noted, he and I have discussed over time that we are looking at different parts of the data set and coming to different conclusions. One specific point that distorts the approaches is the time delay between IANA allocation to the RIRs and the appearance of that space for public use. In particular, his comment about 5 to 8 /8s per year is based on the delayed public use data that will eventually catch up with the fact that IANA has allocated 13 /8s just since the beginning of 2005. If the allocation rates had close to linear growth, the delay would not be a big factor. Another point of distortion is the potential for some of the allocations to never show up as publicly routed.

Ole: So when do we actually run out?

Geoff: There are many specific milestones that will pass in sequence. The unallocated address pool held by IANA will exhaust first, and then the RIR pools of unallocated data will drain. At that point there is no stream of "new" addresses to fuel further growth, and that is probably a reasonable point in time to say that we have "run out." Assuming that the current business influential factors and allocation policies remain in place, then the projection models from recent data indicate that this "run-out" date is around 2016, or some 11 years from now. Of course these are unlikely assumptions as the prospect of exhaustion draws nearer, and there may be a "last-minute rush" of address allocation requests from the service provider industry that could draw in that projected "run-out" date. Such additional consumption pressures are difficult to factor in to trend-based predictive models, of course. It is also conceivable that the industry could shift its attention almost entirely to IPv6-based protocol infrastructure in the coming years, in which case the "run-out" projection for IPv4 would extend out further in time simply because of the translation of the consumption activity to the IPv6 address pool.

Tony: As I noted early on in my article, there will still be pool available at each of the RIRs when the IANA pool that I focused on is exhausted. In the past I have said we would never completely run out because nobody could afford that last address, but in light of the accelerating consumption of IPv4 coupled with the less-than-aggressive deployment of IPv6, I can see how the pool might actually run dry.

John: In practical terms, the point at which one has "run out" of address space is not tied to being the last applicant to the RIRs for an address pool. I have suggested that point will never arise: the RIRs (and, to the extent to which the *Internet Corporation for Assigned Names and Numbers* [ICANN] can make decisions, the IANA), will continually recalibrate policies to prevent "running out." Of course the inevitable consequence of those recalibrations is that, although one does not need to worry about approaching an RIR and being told "no space left," the combination of monetary, justification, and general aggravation costs is such that one does not even want to contemplate being the applicant for the next-to-last available block. That reasoning says that looking at the date on which near exhaustion is reached is relatively uninteresting. The more important question is when one enters the end game for IPv4 space because, as soon as the end game begins, the space is essentially exhausted.

I suggest that the criterion for entrance into the end game is not measured statistically but by looking at the point at which one needs to start designing networks and subnets, not in a way that is optimal from a network architecture or network management and growth standpoint, but in order to conserve address space and/or to avoid extended discussions with applicable RIRs (or one's ISP that deals with the RIR). From that point of view, we have already run out, and probably ran out a couple of years ago. Every time someone who has multiple machines is pointed to private address space because of a presumed shortage, it is an indication that we have already run out of space. Every time China manages to make a successful political point—regardless of the country's actual internal dynamics and economics—about its inability to get addresses for its population, it is an indication that we have already run out of address space. Every time an ISP decides to use private space to manage its backbone, it is an indication that we have already run out of address space.

Fred: I have made the same point, from a point of view of economics. In essence, when a commodity is common and demand is low, there are calls to squander it because it costs nothing—something one hears a lot of in the IPv6 community. When supply and demand are comparable, a market develops, and I need to tell you that I certainly pay for the IPv4 addresses at *my* house. When demand outstrips supply, we enter a regulated market of some kind, and our current allocation policies certainly reflect a regulated market. The step after a regulated market is a black market, and it is not too hard to find that either.

John: Actually, in our present situation, there is an intermediate step before things deteriorate completely into a black market. Although it is unlikely that any significant fraction of the early IPv4 academic, research, or commercial allocations could be recovered and reused, there are governmental allocations that might be recovered under significant political pressures. Unfortunately, in addition to politicizing the allocation process much more than we have seen so far, such moves might push the present users of those allocations toward NATs in ways that would make the ultimate transition to IPv6 more difficult while not gaining very much additional time for the IPv4 space.

Tony: Political pressure or not, simple logistics argues against this. Given the rate of growth in consumption, any reclaimed government space would be consumed in substantially less time than it would take to rebuild their network and release it. Even a small network sitting on a /16 would take at least a year to release that much space, and at the current spot on the escalating curve that /16 represents around 2 hours of IANA run rate. Getting back a whole /8 would logistically take several years, and then at that point on the curve the result would be about a week of run rate. If several of these government organizations have a mesh of direct interactions and head down the same path, the resulting overlap in the private address space would require creating a complex NAT system worthy of a Nobel Prize. Reclamation is a nice bar-room debate topic, but the return on investment is extremely low. If an organization were to consider rebuilding its network to release an IPv4 allocation, it would make much more sense for that organization to rebuild it as IPv6 than to move publicly addressed nodes behind a NAT.

Geoff: It would be strongly preferred by all, I would suggest, that the "black market" option be avoided. If the consequence of the exhaustion of the unallocated pool of IPv4 addresses is the trading of already-allocated IPv4 addresses, then a responsible way for the industry to support that scenario is to encourage such a market to operate with the support of some form of "clear title" that could legitimate trading transactions. Without structure and stability in a trading market, the value of the trade is meaningless, and in this case the potential for chaos in the network itself is undeniable.

Fred: We are in fact starting to see networks designed to be IPv6-only or IPv6-dominant (the latter being a network that might use IPv4 internally but offer only IPv6 services to some or all of its customers) in China, Japan, and other places. The economic argument is the one these operators are primarily giving—they state that they see a roadmap to the number of addresses that they need in IPv6, while in IPv4 they are significantly constrained. This sounds to me a lot like John's comments about network design, but the other way—rather than designing their networks to what they perceive as IPv4 addressing policy limitations, they are choosing a path that they perceive as giving them options.

We also see evidence of networks designing themselves to the limits of address allocation in IPv4, usually using multiple layers of NATs. For quite a while, for example, China Unicom used multiple layers of NAT in order to work around what the company felt was a deficiency in its ability to get IPv4 addresses from its national registry. As I understand it, the company has changed its strategy to include getting IPv4 address allocations directly from APNIC, and at the same time to deploy an IPv6 network in parallel to move away from IPv4 dependence.

John: There is another factor at work in this. Transitions are never free. If we are going to design and build out a substantially new network, we are rapidly reaching the point—some would say that we have reached it already—at which it is cheaper to design and build that network for IPv6, making whatever arrangements are needed at its interconnection points with IPv4 networks, than to build in IPv4 and face a transition later. As those decisions are increasingly made, it may both reduce pressure on new IPv4 allocations and create free pools of IPv4 space that could be recovered and reused. For example, the U.S. Department of Defense (DoD) has announced a fairly aggressive schedule for moving to IPv6. If they meet that schedule and were then willing to free up the IPv4 space that they would presumably no longer be using, it would free up the equivalent of several /8s. While I agree with Tony that this hypothetical case would be unlikely to make any significant difference in the long run, it illustrates another difficulty with trying to make assertions about what is happening by statistical projections alone.

Ole: It is frequently stated that North America is immune to the address exhaustion problem.

Tony: Well despite persistent rumors and press statements to that effect, ARIN continues to consume about 30 percent of the annual allocation from IANA. If the past allocations were sufficient to stave off global exhaustion, why the continued consumption? In any case, when the central pool is exhausted the North American region will be in the same situation as everyone else—unable to expand or acquire new IPv4 addresses.

Geoff: We are seeing growth in Internet-based services in all regions of the industry, including North America. And network growth needs to be fueled by network addresses. We are seeing a combination of a continued demand for further addresses, and the use of various forms of network configurations that attempt to make the most efficient use of already-allocated addresses. There is little data to suggest that any region, including that of North America, is in a position of immunity from these growth-related factors.

Ole: There is widespread opinion that NAT will solve the problems for a long time to come.

Geoff: The ISP industry certainly has made considerable investments there, and many millions of end users today use the Internet behind NAT devices. Given the size of this investment and the factors of inertia in large-scale service markets, it is reasonable to predict that NATs will be around for quite some time. But NATs add cost to network services. If we are talking about a network that is restricted to servicing the communications needs of people, then this is a relatively high-value activity, and the additional costs of the deployment of NATs are being absorbed within the cost base of the network service economy. And for such human activity-based services this may well continue for some time, given the existing levels of industry investment in service infrastructure that includes the use of NATs. Certainly any new application that is adopted by the Internet user population needs to work across a wide variety of NAT configurations. From this perspective it is likely that IPv4 and NATs will continue to be part of the Internet landscape for a long time to come.

But although this approach has the potential to service a portfolio of service markets for some time to come, it cannot service all forms of service markets—not in the future nor even today. It does not solve all the "problems" and certainly does not encompass all the opportunities that the Internet offers. The potential of IPv6 is one that includes an address span designed to match the full potential of the volume-driven silicon industry, both now and in a future that extends out for many decades to come. One likely scenario for IPv6 is in servicing a truly massive device-dense environment. This scenario encompasses far more than services that are primarily directed at human end users. And the associated service market will be more akin to that of a relatively undifferentiated commodity market, where simplicity and low cost are the dominant service provider discriminants. Because of their additional complexity and associated incremental cost, NATs are marginalized in such commodity markets directed at servicing device density, and it is there that the true leverage of the IPv6 address span becomes a major influential factor.

Tony: As Geoff notes, NAT has been widely available and deployed globally over the same timeframe as the recent consumption. Yet the accelerating growth trend continues, consuming to the point where only 25 percent of the total IPv4 space remains available. Although NAT does slow the rate of public address consumption from what it might otherwise be, it creates more problems than it solves. Geoff also raises the economic investment in NAT to date, which is an interesting contrast to many complaints I hear about the cost of deploying IPv6. Most people who look at what it will take to deploy IPv6 in their network are very quick to dismiss this investment in the array of costs associated with NAT. Often they insist on a demonstration of value for the IPv6 investment while at the same time they refuse to allow consideration of removing their development, and ongoing operational support costs for IPv4 NAT.

Although I agree that in the interim overlap period the costs are additive, in the long term staying on the IPv4/NAT path those costs only compound, whereas on the IPv6 path they disappear. The duration of that overlap is somewhat self-controlled as a direct trade-off between the costs for running both protocols in parallel versus the costs associated with aggressively moving the end systems and applications to IPv6.

Ole: Another area frequently discussed on various lists is that the U.S. DoD and Federal Government mandates for service availability in 2008 are just another instance of the *Government OSI Profile* (GOSIP) and that they too will disappear.

Tony: What these discussions miss is that the situation is entirely different now. In the early 1990s the U.S. GOSIP effort was directed by a strong desire to consolidate the array of protocols in use at that time toward a common one. Other governments had similar efforts that led them collectively toward a suite that was developed with international governmental input. IPv4 was an alternative to the mandate with applications already supporting it, while the OSI protocols existed in some router products but did not have many applications available.

At this point the existing government networks are already consolidated, and there is no alternative. Yes, IPv6 still has fledgling application support, but the IPv4 pool is no longer a sustainable resource to draw on, and there is no other option. So the government networks either stop growing or, as the U.S. DoD and Government agencies have announced, they will move to IPv6. This implies preparing the application community to meet the impending reality.

Geoff: Although the strategic directions of one single—but relatively large—market player does have some bearing on the direction of the global market in Internet-based service provision, I do not see evidence that this will be sufficient to influence the entire market in any particular direction. This was certainly evident in the case of GOSIP some years ago, and continues to be an aspect of the market today. The global communications sector carries the impetus and burden of massive investment in infrastructure, process, technology, services, and consumer product portfolios. The sector has already undergone a revolutionary change with the advent of the Internet over the past decade. Doubtless there is considerable reluctance on the part of many sector players to continue to invest in further change in the protocol infrastructure of Internet-based services. On the other hand, the upheavals in the service provider sector have also eliminated much historical complacency about the stability of these markets and the adequacy of the associated service portfolio. It is reasonable to suggest that this sector is now very attentive to the prospect of expanded markets and new service opportunities that can take advantage of the existing infrastructure to create new revenue streams. So I think it is the current dynamics of the service provider sector and the potential for new service markets that would be the most persuasive factor for service providers to invest in an IPv6 protocol infrastructure.

Ole: Closing thoughts?

Tony: As I said at the end of my article, now is the time to recognize that we have reached the end of sustainable growth in IPv4. For most existing organizations that can foretell they have as much space as they will need for the next decade, this is not really an internal problem. Where these organizations will have a concern is when they deal with newcomers or others that have been forced into IPv6 because of exhaustion of the pool. Those organizations that foresee expansion and growth should evaluate Geoff's analysis as well as mine and weigh their plans against the risks of either or both of us being wrong.

In any case it only makes sense to start IPv6 capability discussions with the product vendors now. Product development cycles can be lengthy, and the only way for the vendor community to mesh with an organization's deployment plans is to have sufficient notice about those plans and timeframes. It would also be wise for the organization's network architects to start thinking about the impacts of an IPv6 deployment. Both protocol versions are packet-based and the names start with IP, but there are enough differences in the details that it is worth taking a fresh look to see what might be easier or cheaper than just blindly deploying IPv6 identically to the IPv4 deployment.

Geoff: The Internet continues to present challenges to the communications sector, and I would suggest that the underlying influential factor is the combination of the silicon and software industries that continue to fuel the demand side with fascinating, innovative, and compelling uses of communications that continue to surprise us with their continual restatement of the size of the domain in which we operate. We appear to be moving beyond servicing devices that are activated and influenced primarily by direct human activity, such as e-mail and Web use, and we are now looking at various command, control, and monitoring functions that embed themselves deeply in other devices and in other elements of our infrastructure. This encompasses larger concepts such as "smart buildings" and "smart traffic control," and they reach all the way down to the level of embedding into consumer devices and even identification tags. This is not a world that can readily be serviced by an IPv4 protocol infrastructure, and we are already seeing various levels of network indirection in both NATs and various forms of overlay networks to attempt to compress this new scale of basic network addressing demands into the IPv4 environment. This appears to be a complex, and therefore costly task. But the expectation here is that the service industry is heading toward a commodity utility function, where the essential attributes of the underlying network are simplicity and efficiency. These factors suggest that the market characteristics that arise from the propulsion of the silicon and software industries are inexorably tugging the communications service industry to embrace simple, scalable, and efficient networking technologies. It is in this space that the essential attribute of IPv6, that of the size of the address pool, has its most effective leverage. Here the "run out" of IPv4 will inevitably focus our common attention on how best to engage with future needs and roles. And in this perspective the IPv6 technology has a critical and central role.

John: Tony, I think we need to assume that, when it comes down to translating the projections into an answer to the "when do we need to get serious about IPv6?" question, both you and Geoff are, to a considerable extent, wrong. Geoff's articles and projections have been interpreted by some people as containing a "there is no problem, we can continue with IPv4 until we all retire" message. Viewed from that direction, yours can be seen as "we cannot be quite *that* complacent." Instead, I think we should all be looking at going directly to IPv6 in newer network installations rather than concentrating on whether we can get enough IPv4 space for them. We also need to be examining—now, not a few years in some projected future—the applications and services for end networks and end users, not just backbone and ISP services and operations. One of my particular concerns is that we have enterprise and customer support people and protocols all over the world who are used to thinking about things in an IPv4 world, including the support advantages of "all NAT-based end networks look the same" architectures. The need to retrain them to think about things differently, and to design and build new tools for their use, may suggest a more time-consuming and expensive transition than changing over the networks themselves.

Fred: What is clear to me from this discussion, Geoff's prior analysis, and Tony's analysis here, is that there is a timeline. We are *not* debating whether IPv4 address availability is limited or whether it can be "saved" by address allocation policy, nor are we debating the economic or technical impacts of more or less draconian allocation policies. We *are* debating what constitutes the end game, when and why that end game will become important, and whether perhaps we are already seeing the first steps of it. We are also not debating whether perhaps some new architecture would be preferred over the one in IPv6; if we had an alternative on the table today we could discuss that, but experience tells us that the proposals being considered by the *National Science Foundation* (NSF) and others are sufficiently "researchy" to not be ready for wide-scale deployment in the necessary timeframe.

As such, from my perspective, there is a present call to action.

What U.S. DoD and recent congressional hearings have recommended is in keeping with the IETF's recommendation and with the IPv6 address allocation strategies of the RIRs. The simplest transition strategy involves presently procuring equipment, operating systems, and applications that are IPv6-capable in preference to systems that are limited to IPv4. At some point in the future, perhaps in the 2008–2010 timeframe, we should plan to turn on IPv6 networking capabilities throughout our networks, and this means gaining experience with IPv6 on a smaller scale in 2005–2007 in our networks, in server applications, and in user systems. Turning down IPv4 capabilities, which is the endpoint of such a transition, is a business decision that does not need to be made hastily; we should presume that coexistence will be important for a decade, and probably more.

Ole: Thank you, gentlemen!

TONY HAIN is currently the Senior Technical Leader, IPv6 technologies, with Cisco Systems. In addition to providing guidance to the various internal product teams, he was also co-chair of the IETF working group developing IPv6 transition tools. His IETF participation since 1987 includes a term on the Internet Architecture Board from 1997 to 2001. Named an *IPv6 Forum Fellow* in 2004, he is currently serving as Technology Director on the forum's North American IPv6 Task Force steering committee. Prior to joining Cisco in 2001, he spent 5 years at Microsoft, where his roles included Program Manager for IPv6 as well as Network Analyst for the CIO's office. Prior to Microsoft, he was the Associate Network Manager for the U.S. Department of Energy's Internet effort, ESnet. With this range of roles, spanning the space between the implementation technologists and senior management, he brings a real-world viewpoint to the deployment decision process. E-mail: **ahain@cisco.com**

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector, and has served his time with Telstra, where he was the Chief Scientist in the company's Internet area. Geoff is currently the Internet Research Scientist at the Asia Pacific Network Information Centre (APNIC). He served as a member of the Internet Architecture Board from 1999 until 2005, and currently co-chairs the Site Multi-homing and Routing Operations IETF Working Groups. He is author of *The ISP Survival Guide,* ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks,* ISBN 0471-378089, and co-author of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks,* ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons. E-mail: **gih@apnic.net**

JOHN KLENSIN is an independent consultant based in Cambridge, Massachusetts. He has been involved in the design, development, and deployment of ARPANET and Internet applications, and occasionally lower-layer technologies, since the late 1960s and early 1970s. He has also been intermittently involved with Internet administrative and policy issues since the early 1980s. His current work primarily focuses on internationalization of the Internet on both technical and policy dimensions. E-mail: **klensin@jck.com**

FRED BAKER has worked in the data communications industry, building network elements such as switches and routers, since 1978. His involvement with Internet technology started in 1986, and with the IETF in 1989. He has contributed to the development of OSPF, QoS, PPP, SNMP MIBs, and a variety of other technologies. He has also held a variety of management positions, including chairing various working groups, participating in the IAB, and chairing the IETF. He currently serves on the Technical Advisory Board of the U.S. Federal Communications Commission and as the Chairman of ISOC's Board of Trustees. E-mail: **fred@cisco.com**

Practical Uses of SSH Tunneling in the Internetwork

by Ronnie Angello

While the growing popularity of broadband Internet services and elevated concerns with securing *Wireless LANs* (WLANs) have become major concerns for network administrators today, *Secure Shell* (SSH) *Protocol* tunneling has proven to be a secure and effective solution for addressing various needs and concerns of both network users and administrators. Making the transition from traditional dialup remote access to a broadband solution can bring along with it some roadblocks when trying to preserve functions and security. WLANs can be difficult to secure in the enterprise, mainly because of the various client types that must connect to the network. SSH tunneling can help alleviate both of these issues.

SSH tunneling, also known as SSH *port forwarding*, is the process of forwarding selected TCP ports through an authenticated and encrypted tunnel. These tunnels can be constrained to within two points of the company's enterprise network, or it can originate on a small office or home office (SOHO) computer on a given provider's network, and transit the Internet to a server on the enterprise network. Some practical uses for SSH tunneling are outlined in this article.

A Look Back at Traditional Remote Access

Remote access is the method of connecting from a SOHO computer that resides on a remote foreign network, or has no permanent network connection, to the enterprise network or central office. Usually this involves traversing the Internet. This can be for the purpose of telecommuting, providing on-call support from home, checking e-mail while away from the office, or for the old-fashioned workaholic who must work from home. Remote access used to involve simply accessing a network through an analog phone line or possibly ISDN. In either case, the user was authenticated by an access server that resides on the enterprise network and given authorization to certain resources.

When connected to the access server, users had the feel of being connected to their company's enterprise network. They were free to browse internal Web pages and access various Windows domain resources. They could connect to the network neighborhood and transfer files to and from the work computer. They could connect directly to internal UNIX servers with SSH and use a local X-server application to access UNIX applications from the SOHO.

PC remote-control applications such as VNC, etc. could be used to access files and applications that reside on a host computer on the enterprise network without extensive configuration on the home PC. In addition to the ease of configuration for the administrator or user, fewer applications need to be installed on the home computer to accomplish work tasks from home. This approach saves software licenses in addition to valuable company resources.

Most network administrators cannot let PC configuration consume a great deal of their time because they are busy enough as it is. From a function standpoint, users felt like they were working from their office at work. It was too slow though, so it did not really matter. Then broadband services were introduced, and they offer high bandwidth, but getting the same functions is a bit more challenging. Users benefit from the extra added bandwidth, but of course the administrator has to make sure that everything works as if nothing ever changed.

Broadband Services Emerge

Many users are now migrating from their traditional dialup connections for Internet access to a technology that offers more bandwidth such as cable or DSL. Broadband wireless services are now emerging in some areas as well. These services may even be cheaper than what the company or individual was previously paying for ISDN service, and it is "always on." Most users are no longer dialing a company access server to access the resources that are vital to their job. They are now permanently connected to a foreign provider's network, and often the only choice for secure remote access to the enterprise is through a VPN. Strict policies, however, may need to be enforced on the remote SOHO computer for it to be a comfortable solution for security administrators to implement.

For those organizations without the time, money, or manpower to implement and support VPN, Linux login servers can be opened up to the Internet to authenticate users that employ SSH to access the enterprise network from these remote networks. These servers are no more than relay points to access internal systems. They should be placed in the DMZ or on a "screened" network protected by a firewall. The other internal systems are not directly accessible from the remote networks. In cases where remote access is considered a valuable resource to the organization, more than one of these servers should be implemented for load sharing and redundancy.

However, certain functions are lost. Initiating an application from a UNIX computer and displaying it to your SOHO computer with a local X server has been proven to be slow and inadequate from some remote networks. In addition, internal domain PCs and network shares are no longer accessible through the network neighborhood, and file transfer is not available without an additional secure, standalone application. The remote-control applications that access the internal PC will no longer work without opening holes in the firewall. There is a simple solution to all this that is free, secure, and effective: SSH tunneling.

Securing Broadband Remote Access

The functions described in this section can be achieved with any SSH client capable of tunneling, any Web browser that supports HTTP and *Secure Sockets Layer* (SSL) proxies, and any PC remote-control application. The first step is always to connect to the remote login server that has been made accessible to the SOHO user. When connected to this login server, the user can use SSH to access any other internal machine, or take advantage of SSH port forwarding to accomplish their other tasks.

A proxy server may already be configured on your enterprise network. This server is configured to accept connection requests for Web pages and allow the clients to view them with little network overhead. The SSH client on the SOHO computer is configured to forward the specified local source HTTP port (such as 8080) to port 80 on the remote destination HTTP proxy server. It can also be configured to forward the specified local source SSL port (such as 4433) to port 443 on the remote destination SSL proxy server.

The browser on the client machine is configured to use the HTTP or SSL proxy server **`localhost`** on the specified local port(s). When the browser attempts to download a page, the SSH client forwards the request to the specified remote proxy server on your enterprise network through the established tunnel. Internal Web pages that would normally be available only on the enterprise local intranet are available without latency and without compromising security.

The same concept can be followed for tunneling PC remote-control application data through SSH. The remote-control host service is not changed, and it is waiting for a connection attempt from a remote computer as it normally would. A new remote-control connection is configured on the SOHO computer pointing to **`localhost`**. Using any additional encryption offered by the remote-control application is possible, but not necessary. Additional encryption will add latency, and SSH provides strong encryption itself with *Triple Digital Encryption Standard* (3DES), Blowfish, etc. The SSH client is configured to forward the local source ports used for the remote-control data (that is, port 3389 for RDP) to destination ports on the host computer on the enterprise network.

Once again, all the functions that the user had when dialing up the enterprise network directly are now available. With SSH, an additional layer of security is provided. Because the desktop of the internal computer is available on the SOHO computer's desktop, users have access to all applications, files, and network resources that they would if they were physically working from their office at work. No additional software applications need to be installed on the office computer to satisfy requirements of working from home, and minimal software needs to be installed on the users' personal home computers. Some of these remote-control applications also provide a file transfer tool that can be used to transfer or synchronize files between the two PCs.

SSH Tunneling for WLAN Security

Securing WLANs has become a monumental problem today for most network administrators. Many organizations are resorting to proprietary solutions or are simply avoiding the implementation of WLANs entirely. An entire article could be dedicated to the importance of securing wireless and the details of accomplishing such a feat.

In addition to the uses described in the previous sections, SSH tunneling can also be used to supplement or replace weaker, more vulnerable encryption found in other network applications. Consider *Wired Equivalent Privacy* (WEP) encryption, for example.

Although other alternatives such as *Wi-Fi Protected Access* (WPA) are available, most WLANs have been implemented with either no encryption or with static WEP only. Static WEP has been highly criticized because of vulnerabilities in the protocol that have been discovered and widely documented. Even when implemented at the 128-bit level, there are tools circulating the Internet that exploit a well-known vulnerability that allows a hacker to crack WEP keys. Even with a WPA solution in place, there will be clients that support only static WEP. These traditional clients can be secured in the meantime by restricting network access with an *Access Control List* (ACL) and tunneling insecure protocols through SSH. Once again, the same functions can be achieved with a VPN solution, but some organizations have neither the money nor resources to implement it.

Summary

In conclusion, SSH tunneling can be used well beyond the scope of the methods explained this article. The particular uses outlined in the previous sections have been practical in my experience and have been very successful implementations. When users decide to change to a provider that offers broadband, I have found that simply providing a procedure for configuring tunneling has been successful for getting them operational from home.

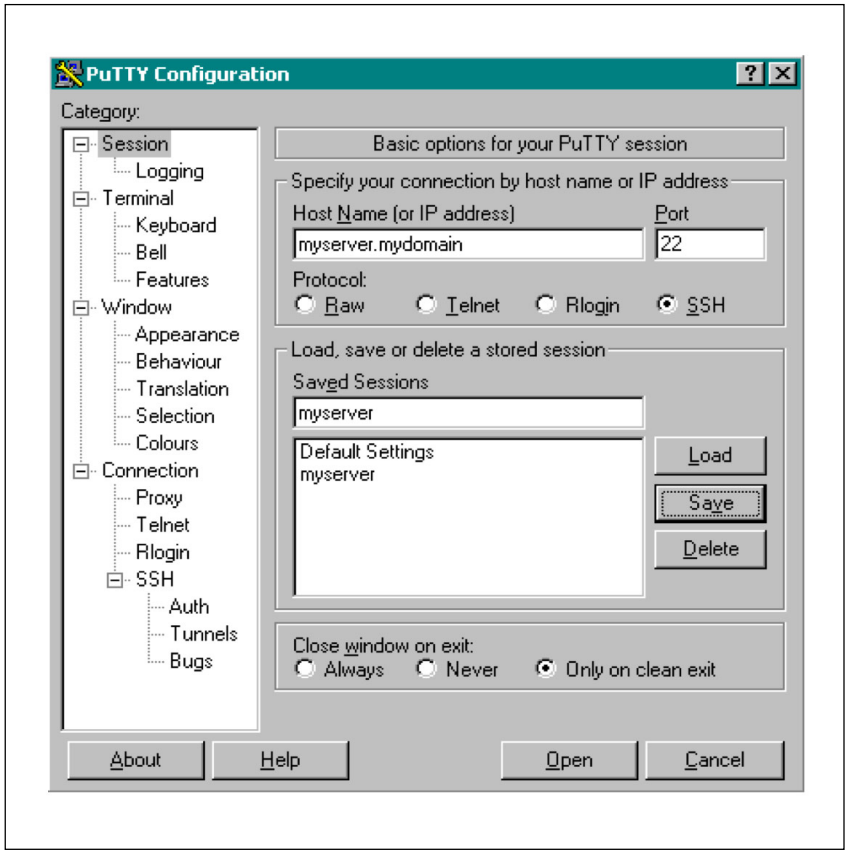
SSH tunneling should be of interest to any organization that wishes to allow its users secure access to all the resources that they may need to accomplish their job functions—especially from a remote location. While exploring possibilities to make a particular application or protocol secure, always consider SSH tunneling an option. SSH provides authentication and encryption that has been proven to be effective for any application.

Securing Remote Access to Internal PCs, Web Pages, etc.

The following is a short example procedure for configuring tunneling for this specific function. It does not include detailed instructions for configuring specific applications, but it outlines the important steps that must be followed in order for it to work properly.

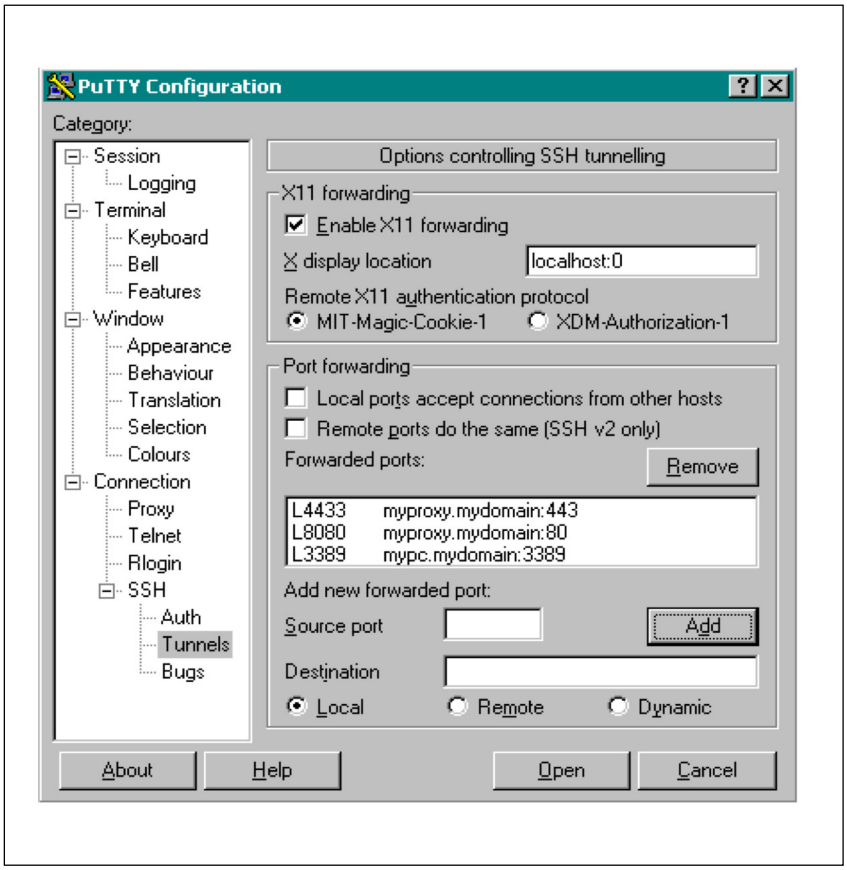
- Any SSH client that supports tunneling can be used. You can download the PuTTY SSH client (**`putty.exe`**) from: **`http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html`**
- Make sure that you select port 22 (SSH). (See Figure 1.)

Figure 1: PuTTY Configuration Screen — Sessions



- Choose your preferred encryption cipher; enable compression and X forwarding if desirable. Click "tunnels" in the tree menu. Add the local source port(s) and the remote destination port(s) for the ports that you would like to forward through the tunnel. (See Figure 2.)

Figure 2: PuTTY Configuration Screen —Tunnels



- Make sure that the LAN settings in your Web browser are configured to use the HTTP/SSL proxy server **`localhost`** on the local port that you specified.
- Make sure that your remote-control connection is pointing to the computer "LOCALHOST." If you have trouble connecting, make sure that the host service is running on the host PC.

For Further Reading

[1] The SSH (Secure Shell) Remote Login Protocol, SSH-1 Specification, T. Ylonen, November 1995.

[2] SSH-2 Specifications IETF Secure Shell working group, June 2003.

[3] O'Reilly Network Using SSH Tunneling: **`http://www.oreillynet.com/pub/a/wireless/2001/02/23/wep.html`**

[4] SSH Tunneling: **`http://www.ccs.neu.edu/groups/systems/howto/howto-sshtunnel.html`**

[5] SSH Tunnel Tiny HOWTO:
http://www.frozenblue.net/tools/howtos/?v=ssh-tunnel

[6] Secure Email Through SSH Tunneling:
http://www.slac.com/~mpilone/projects/kde/kmailssh/

[7] Mac OS X SSH Tunneling:
http://info-center.ccit.arizona.edu/~consult/macx-tunnel.html

[8] PuTTY Links:
http://cdot.senecac.on.ca/software/putty/links.html

[9] William Stallings, "SSL: Foundation for Web Security," *The Internet Protocol Journal,* Volume 1, No. 1, June 1998.

RONNIE ANGELLO, CCNP, CQS-CWLANSS, CCNA, holds an A.A.S. Degree in Information Systems Technology (Specialization in Operating Systems and Network Operations) and is currently completing degree requirements for the Bachelor of Science Degree in Information Science (Concentration in Networking and Communications) at Christopher Newport University in Newport News, Va. He recently passed the CCIE Routing and Switching Qualification Exam and is preparing for the CCIE Lab Exam. E-mail: **angello@jlab.org**

Book Review

Network Algorithmics

Network Algorithmics: An Interdisciplinary Approach to Designing Fast Networked Devices, by George Varghese, ISBN 0120884771, Morgan Kaufmann, 2004.

This is not a generic algorithms book (that is, it does not overlap much at all with Sedgewick or Coleman as an introduction to algorithms), nor is it a typical introduction to TCP/IP networking book (for example, there is no chapter defining the TCP/UDP/IP header fields, thank goodness). It might best be described as an algorithms analysis book set in the context of networking and also in the context of implementations that mix hardware and software solutions. For those familiar with Radia Perlman's book *Interconnections,* I found aspects of the writing style and approach to be similar. George Varghese—in addition to having been a networking professor for many years—has had a lot of industry experience from licensing algorithms to networking companies, to consulting with Procket Networks in the company's early days of architecting its core router, to starting a security company that was recently acquired by Cisco Systems. I have been doing architecture work at Cisco for several years and can say that George's book has real grounding in how systems are built and analyzed today.

Organization

Chapter 2 presents abstractions for networking protocols, hardware design, routers, memory technology, and Internet end nodes (servers). This is a great introduction into "systems" thinking. In section 2.2.7, "Final Hardware Lessons," one thing I thought George should have mentioned along with metrics of chip size, speed, I/O, and memory is *power*. Power is becoming a major systems concern in many platforms and deserves mention as an optimization constraint.

Chapters 3 and 4 go through a list of 15 implementation principles to use in approaching algorithmic design in systems and then give examples of these principles in action. What I find interesting about this section is that from working with George in the past, he really does believe and practice "principle"-based architecture thinking. I remember discussing several of the principles with him several years ago, and you can see how his many years of experience working in the networking field have shaped these principles. Many have probably employed some of these, but as George says in the chapter introduction, having them explicitly documented with examples is useful to help clarify our thinking. Some of the principles (and both the short examples in this chapter as well as examples cited in more detail in later chapters) are really fundamental, and I think reading through examples helped clarify in my mind when to use them.

Chapter 5 covers copying data, for example, in a server design. I really like this type of chapter, in which a subject (in this case the effect of packet copying on Web server performance) is explored in detail but with a focus on where algorithms and systems design play an important part.

My biggest question about this chapter is that I was unsure how applicable this is to, say, modern server design using Linux and with latest Gigabit Ethernet *network-interface-card* (NIC) designs. I know there was a lot of interesting work in the late 1990s, but this chapter without any data is more along the lines of an extended example of how to apply implementation principles.

Chapters 6 through 9 are not what I would consider the meat of the book; they treat the topics of implementation and analysis for servers, timers, parsing/classification of packets, and buffer management (memory allocation).

Chapter 10 covers exact match lookups. There is not a lot of meaty algorithmic discussion, but the history of scaling performance of bridges is used to elegantly show an evolution of algorithmic approaches to exact matching.

Chapter 11 is an awesome overview of the state-of-the-art in longest prefix match (used for destination address matching in routers and switches). A good read of this chapter will yield an understanding of the trade-offs in all major published algorithms, although there may be variations or tuned versions of these algorithms in use at companies like Cisco. I believe this chapter covers all the major categories of solutions.

Chapter 12 extends the prior chapter into more general packet classification (which is used in applications like extended access lists). Like the lookup chapter, this chapter addresses one of George's prime core competencies. There is good discussion on leading published approaches (Grid-of-Trie, cross producting, geometric, and decision tree-based approaches). I strongly recommend this chapter.

Chapters 13 and 14 cover packet switching (that is, architecture of fabrics like crossbars for connecting line cards in a router or switch) and then packet scheduling. These topics get a good academic treatment (after all, George is one who introduced *Modified Deficit Round Robin* (MDRR) to the industry as well as academia), and although there are gaps between what many networking markets are defining as requirements for packet scheduling and what is in this chapter, the chapter is still useful.

Chapter 15 is a short chapter that tries to treat at a high analytic level the algorithmic problems involved with routing protocols. It covers this topic without getting very specific into nonrelevant (to the analysis) networking details.

Chapter 16, which addresses measuring network traffic, was probably one of my least favorite chapters. Some of it is academically interesting but requires network level changes that I just do not think will occur. There are some cute tricks relative to counters and such, but I think they are similar to approaches already being used.

Chapter 17 is a network security chapter and seems to serve as an early introduction to the topic of algorithms in network security; this is not a major focus area of the book.

Areas for Improvement

There is always room for improvement, and I list here three areas in which this book could have been improved:

1. There is a running thread in the book of prefacing technical discussions in some cases with an example from the "normal world," like comparing packets to envelopes in the postal system. I estimate this is less than 1 percent of the content of the book and fairly easy to ignore if it annoys you.
2. I would have enjoyed better (more detailed) figures. A well-done, detailed figure can incorporate multiple concepts in the text around it and make it much clearer. On the positive side, there are numerous figures in the specifications, even if they do tend to be simple and high level.
3. Another area that I would have enjoyed seeing more on is empirical data (tables of data and graphs). I enjoy detailed empirical data of the type that Hennessy and Patterson so effectively use in their *Computer Architecture* book. There are many places (for example, Web server optimizations in Chapter 5) that I think could have benefited from detailed empirical data. However, I think folks often rely on empirical data too much when a simple analysis like the type done throughout the book could be done to help optimize the problem.

Recommended

Many chapters in this book are directly relevant to the development of networking equipment and software, as well as what is "under the hood" of networking equipment. The book is fun to read and I believe succeeds in trying to convey an organized systems approach to thinking about problems in the networking space.

—*Will Eatherton*
will@cisco.com

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

Internet Governance Report Available

The *Computer Science and Telecommunications Board* (CSTB) of the National Academies has recently published a report entitled "Signposts in Cyberspace: The Domain Name System and Internet Navigation."

A summary report, as well as links to the full report can be found at:
`http://www.cstb.org/dns/signpost.html`

From the summary: "The *Domain Name System* (DNS) enables user-friendly alphanumeric names to be assigned to Internet sites. Many of these names have gained economic, social, and political value, leading to conflicts over their ownership—especially names containing trademarked terms. Congress, in Public Law 105-305, directed the Department of Commerce to request the *National Research Council* (NRC) to perform a study of these issues. When the study was initiated, steps were already underway to address the resolution of domain name conflicts, but the continued rapid expansion of the use of the Internet had raised a number of additional policy and technical issues. Furthermore, it became clear that the introduction of search engines and other tools for Internet navigation was affecting the DNS. Consequently, the study was expanded to include policy and technical issues related to the DNS in the context of Internet navigation. This report presents the NRC's assessment of the current state and future prospects of the DNS and Internet navigation, and its conclusions and recommendations concerning key technical and policy issues."

The report was produced by the Committee on Internet Navigation and the Domain Name System: Technical Alternatives and Policy Implications, National Research Council.

First Protocols for Policy Makers Forum to be held October 28

The Internet has achieved the same global economic significance that propelled issues of international trade and finance onto the front pages of newspapers and the forefront of international policy thinking twenty years ago. This change is raising the profile of specialized issues and "obscure" policies for a rapidly expanding circle of public and private-sector stakeholders. Increased general understanding will be vital to assuring that Internet's growth, development, and coordination mechanisms continue to serve important public interests.

In recognition of this growing need for public education, Packet Clearing House is organizing a series of day-long roundtable fora to encourage sharing of technical and institutional know-how between prominent Internet architects, policy makers, and leading opinion leaders from related sectors. With the support of the *American Registry for Internet Numbers* (ARIN), the forum, to be called *Protocols for Policy Makers* (PfP), will meet for the first time on October 28, in conjunction with the NANOG 35 and ARIN XVI Internet operations and policy meetings in Los Angeles, California.
See **`http://nanog.org/arinattend.html`**

PfP will explore themes of competition, coordination, and possible conflict between new alternative Internet naming and addressing systems which are challenging the status-quo, such as the national registries recently proposed by the International Telecommunications Union and competitive private-sector "alternate roots." What outstanding problems are these new mechanisms intended to solve, and what goals might they achieve? How will these innovations contribute to the advancement of Internet public interests? What risks, costs, and complications may be imposed on the Internet by the emergence of multiple divergent systems? At PfP, these issues will be examined through a day of structured round-table discussions, interspersed with comments from leading experts on the Internet's current naming and addressing systems and prominent advocates of the current restructuring proposals. A complete agenda and list of speakers will be published shortly at **http://www.pch.net**

PfP will be open to the public, but space is very limited. For more information, or to request an invitation, please e-mail **pfp@pch.net**. Expressions of interest from potential speakers, meeting hosts, and institutional co-sponsors are also welcome. Plans for future PfP meetings are already underway, with a second meeting, tentatively titled "When Voice Goes to Bits" to focus on technical, commercial, and regulatory implications of the migration voice telephony to IP. Suggestions for future meeting themes, venues, and contributions should be directed to PfP Forum Chair Tom Vest at **pfp-sponsor@pch.net**

Jun Murai Recognized with Postel Award

Professor Jun Murai is this year's recipient of the Internet Society's prestigious *Jonathan B. Postel Service Award*. The award recognizes Professor Murai's vision and pioneering work that helped countless others to spread the Internet across the Asia Pacific region.

The Postel Award was presented during the 63rd meeting of the *Internet Engineering Task Force* (IETF) in Paris, France by Daniel Karrenberg, chair of this year's Postel Award committee, and Lynn St. Amour, President and CEO of the Internet Society.

"Jun Murai has always encouraged, inspired and helped others, particularly his students and his colleagues in other parts of the Asia Pacific region," said Karrenberg. "He has also played a key role in creating structures for Internet coordination in the region (particularly the *Asia Pacific Network Information Centre* [APNIC]), and he is widely recognized for his recent pioneering work in IPv6 implementation."

Jun Murai is currently Vice-President at Keio University in Japan, where he is a Professor in the Faculty of Environmental Information. In 1984, he developed the *Japan University UNIX Network* (JUNET), and in 1988 established the WIDE Project (a Japanese Internet research consortium) of which he continues to serve as the General Chairperson. He is President of the *Japan Network Information Center* (JPNIC), a former member of the Board of Trustees of the Internet Society and a former member of ICANN's Board of Directors.

The Jonathan B. Postel Service Award was established by the *Internet Society* (ISOC) to honor those who have made outstanding contributions in service to the data communications community. The award is focused on sustained and substantial technical contributions, service to the community, and leadership. With respect to leadership, the nominating committee places particular emphasis on candidates who have supported and enabled others in addition to their own specific actions.

The award is named after Dr. Jonathan B. Postel, who embodied all of these qualities during his extraordinary stewardship over the course of a thirty-year career in networking. He served as the editor of the RFC series of notes from its inception in 1969, until 1998. He also served as the ARPANET "Numbers Czar" and the *Internet Assigned Numbers Authority* (IANA) over the same period of time. He was a founding member of the *Internet Architecture Board* (IAB) and the first individual member of ISOC, where he also served as a trustee.

Previous recipients of the Postel Award include Jon himself (posthumously and accepted by his mother), Scott Bradner, Daniel Karrenberg, Stephen Wolff, Peter Kirstein and Phill Gross. The award consists of an engraved crystal globe and $20,000.

ISOC is a not-for-profit membership organization founded in 1992 to provide leadership in Internet-related standards, education, and policy. With offices in Washington, DC, and Geneva, Switzerland, it is dedicated to ensuring the open development, evolution and use of the Internet for the benefit of people throughout the world. ISOC is the organizational home of the IETF and other Internet-related bodies who together play a critical role in ensuring that the Internet develops in a stable and open manner. For over 13 years ISOC has run international network training programs for developing countries and these have played a vital role in setting up the Internet connections and networks in virtually every country connecting to the Internet during this time. For more information visit: **`http://www.isoc.org`**

Internet Root Servers Deployed in India

APNIC recently announced that three new Internet DNS root name servers are now operational in India.

These servers, launched in an official ceremony in New Dehli, India, on 25 August 2005, are the first root name servers deployed in India and South Asia and are already bringing significant improvements in speed and reliability to Internet users in India and the surrounding region.

APNIC has coordinated these deployments with the *Department of Information Technology* (DIT) and the respective root server operators.

F-root, operated by *Internet Software Consortium* (ISC) has been installed in Chennai; I-root, operated by Autonomica, has been installed in Mumbai; and K-root, operated by RIPE NCC, has been installed in Noida, near Delhi.

The installation of the root servers in India has been made possible by DIT, the *National Internet Exchange of India* (NIXI), and the *Internet Service Provider Association of India* (ISPAI), with financial and logistical support from APNIC. The three deployments in India bring the total number of root DNS servers in the Asia Pacific region to 24, 16 of which have been made possible with APNIC's support.

"We are pleased that India is able to contribute to the deployment of the first root name servers in South Asia," said Mr Pankaj Agrawala, Joint Secretary of DIT. "These three root servers will not only benefit the Indian Internet community, but also Internet communities in the surrounding region."

Paul Wilson, Director General of APNIC, added, "The deployment of these three root name servers in India is a positive example of Internet community coordination. The installation has involved the private sector, not-for-profit organizations, and government bodies working together to improve DNS stability and Internet response times for developing countries in South Asia."

Amitabh Singhal, Acting CEO of NIXI, said, "India is among the top ten countries in Internet usage, with over 35 million current subscribers and a five year target for 40 million, translating into more than 200 million total users by 2010. Sustainable infrastructure capacity building is imperative. As a budding intellectual capital of the world, with conducive socio-economic and political environments, India is justifiably proud of hosting three root servers, visibly putting our country, as well as the South Asian region, firmly on the world Internet route map."

More information about the participants can be found below.

- *APNIC* is one of five Regional Internet Registries currently operating in the world. It provides allocation and registration services which support the operation of the Internet globally.
  http://www.apnic.net
- *Autonomica AB* is responsible for **i.root-servers.net**, the first root name server to be installed outside the United States of America. **i.root-servers.net** has been operational since 1991 and is now anycast from more than 25 locations around the Internet.
  http://www.autonomica.se
- *DIT* operates under the Ministry of Communications and Information Technology, *Government of India* (GOI).
  http://www.mit.gov.in
- *ISC* operates one of the 13 root DNS servers as a public service to the Internet. ISC has operated F-root for the IANA since 1993.
  http://www.isc.org
- *NIXI* is joint effort between the GOI and the ISP industry to localize Internet traffic in India. NIXI has nodes in Delhi, Mumbai, Chennai and Kolkatta. **http://www.nixi.in**
- The *RIPE NCC* is one of five Regional Internet Registries currently operating in the world. It provides allocation and registration services which support the operation of the Internet globally.
  http://www.ripe.net

IETF Journal Announced

The Internet Society (ISOC) is pleased to announce the *IETF Journal,* a new publication produced in cooperation with the IETF Edu team. Our aim is to provide an easily understandable overview of what is happening in the world of Internet standards, with a particular focus on the activities of the IETF *Working Groups* (WGs). Each issue of the journal will highlight some of the hot issues being discussed in IETF meetings and in the IETF mailing lists.

The focus of this first issue will be a look back at the accomplishments of the recent 63rd meeting of the IETF in Paris.

We trust that this publication will give all those with an interest in the increasingly important Internet standards development process an opportunity to keep abreast of many of the topics being debated by the IETF. Articles will cover issues such as:

- Reports from the IETF and IAB Chair
- News from the IETF Edu Team
- Update from the IASA and the IAD
- Summary of the plenary discussions
- Highlights of IETF developments related to topics such as Routing, DNS, and IPv6
- Recently published RFCs.

The journal will be available shortly at the following URL:
`http://www.isoc.org/pubs/IETF-Journal`

Upcoming Events

The *North American Network Operators' Group* (NANOG) will meet in Los Angeles, October 23–25, 2005. For more information, see:
`http://nanog.org`

The *American Registry for Internet Numbers* (ARIN) will meet (jointly with NANOG) in Los Angeles, October 26–28, 2005. For more information, see: **`http://arin.net`**

The *Internet Engineering Task Force* (IETF) will meet in Vancouver, Canada, November 6–11, 2005. For more information, visit:
`http://ietf.org`

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in Vancouver, Canada, November 30–December 4, 2005. For more information, see: **`http://www.icann.org`**

The *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will be held in Perth, Australia, February 22–March 3, 2006. For more information, see: **`http://www.2006.apricot.net`**

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is…?"), as well as implementation/operation articles ("How to…"). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

December 2005 Volume 8, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Perhaps the greatest challenge facing the Internet is the ever-increasing amount of unwanted e-mail, commonly known as *spam*. It is tempting to compare electronic mail to its paper counterpart, but there are some important differences. First, "junk-mail" is relatively self-limiting in scope because it costs real money to print and distribute even the most modest flyer. Second, advertisers in the real world are interested in *targeting* their audience. It makes little sense for a supermarket in Boston to advertise weekly specials on produce to consumers in Tokyo. Bulk mail—when delivered by the local postal service—is also quite carefully regulated. It is somewhat rare that you cannot locate the sender of paper-based advertising. None of these observations can be applied to spam. Sending spam is more or less "free," spammers often target "the entire world," and spammers can easily hide behind fake or transient addresses.

To date, spam has been tackled largely by applying sophisticated filtering techniques for incoming e-mail, but this does nothing to decrease the amount of actual spam sent. Anti-spam legislation has been passed in some countries, but it remains difficult—if not impossible—to pursue spammers through legal means, especially in an international context. It is therefore natural to look at technological solutions to the spam problem. If we can secure our network and authenticate its users, would it not be possible to allow only "authorized and verified" senders to send e-mail? Dave Crocker examines this problem in our first article.

Of course, no simple technical solution for spam exists, and not surprisingly there are divergent views on how the problem should be tackled. Our second article, by John Klensin, looks at spam from a different perspective and suggests some possible avenues towards a solution.

Our final article looks at routing protocol testing. Russ White examines testing mechanisms and discusses guidelines for realistic testing.

Many of you have already responded to the *IPJ Reader Survey*. There is still time to participate. If you received an e-mail invitation to take the survey, simply follow the link in the message. You can also take the survey by following the survey link on the IPJ home page: **http://www.cisco.com/ipj**. If you prefer to just drop us a line with your comments and suggestions you can do so by sending e-mail to: **ipj@cisco.com**.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Challenges in Anti-Spam Efforts

by Dave Crocker, Brandenburg InternetWorking

It is said that the Internet teaches us one lesson. That lesson is "scaling." The Internet comprises perhaps one billion users, millions of machines and many tens or hundreds of thousands of independent service operators. It operates in, and between, virtually every country on the planet. It is used for personal, organizational and governmental services. Therefore, it must be compatible with many different cultures, many different styles of communication and many different methods of administration. The Internet has no central point of control and operates according to no set schedule. Hence, changes must be gradual and voluntary—when we agree on what those changes should be.

In the early 1990s, the Internet grew from a small research community into a global mass market. Imagine a small town changing into a large, undisciplined city. In a large city, most people are strangers, and the strangers have a diverse range of values and behaviors. Hence, people must use much more caution with each other. In other words, the problems are not with the original way the town operated, but with changing requirements. So, spam is merely an unfortunate—but frankly predictable—example of the Internet's success, not its failure.

This article explores the system-level complexities of the spam problem, as the intersection of social diversity, complexity of e-mail technology and operations, and specific lines of attack that seek to control spam. On the question of control methodologies, most prior work has been on analytic tools that are used by sites receiving spam, to evaluate the mail content, associated addresses or traffic flow. Recent efforts focus on assignment and assessment of an accountable identity that is responsible for individual messages or for the transit of aggregate message traffic.

The Nature of Spam

People agree that spam is a serious problem, but they have difficulty agreeing on its definition. *Unsolicited Bulk E-mail* (UBE) is probably the most useful.[1] A spammer sends a large number of messages to many different recipients who have not requested the content. (Interestingly most spammers do not care whether a particular addressee receives the message; they merely seek to get a sufficient percent of their postings delivered to some of the addressees.)

Spam can conform to Internet technical standards and can contain no technical differences from legitimate—desired—messages. Hence, spam that violates standards or has other peculiarities might be common today, but detection efforts that are based on these anomalies offer no long-term benefits. Spammers are highly adaptable and use the easiest method that works. However what spam *always* violates are our *social* conventions. Therefore, any long-term, proactive, technical responses to it, such as formulation of standards, must follow, rather than lead our social decisions about it.

Like other social problems, we probably can control spam, even if we cannot eliminate it. This means that we must adjust to having spam as a permanent part of our social landscape, even as we seek to limit it to tolerable levels. Efforts to detect and eliminate spam have been underway for quite a few years. Some techniques have shown useful, localized results, but most only for a short time. In other words, none of the many spam control attempts, over the years, has yet reduced the amount of global spam! So we must be cautious about our expectations for any new anti-spam proposal. It also is likely that controlling spam requires an array of complementary techniques and continued efforts to adapt them, as spammers continue to adapt their own methods. This means that we need to assess any new proposal in terms of its likely *incremental* benefit, rather than as a candidate to be the *Final Ultimate Solution to Solve Spam* (FUSSP).

Changing a global infrastructure takes a long time and is very expensive. Some proposals require complex technology, while others require substantial, on-going administrative effort. Worse, some impose onerous requirements on end-users. Therefore we need to ensure that the mechanisms we deploy will have significant, long-term benefit, even after spammers try to adapt to their presence. They also must have reasonable development cost, require limited, on-going administration and be sufficiently easy to use. In evaluating the likely efficacy of a proposal, a useful heuristic is to ask whether it would be desired even if spam were not a problem. If the answer is yes, then it provides general, strategic benefit, so that counteracting spam merely adds urgency to its adoption.

The Internet provides us all with vastly better access to each other. For collaboration, or the formation of specialized communities or for personal interaction, this is wonderful. For intrusions into our privacy and threats to our online security, this is problematic. Unfortunately, the benefits and the detriments are tightly coupled. Our efforts to control e-mail's problems need to be made cautiously, lest we also reduce its benefits. Worse, our efforts need to limit the damage that might be done to innovative benefits that we have not yet envisioned.

The sender of spam incurs almost no incremental cost for a single message. It is easy to think that we should simply make e-mail be the same as sending letters or making phone calls, by directly charging the sender for every message. This cost provides a barrier against abusive, bulk use. In reality e-mail is a different kind of service, with an extensive history, and it is subject to different choices. Telephones and postal service have highly centralized, formal operational authorities, and the fees charged for their use are based on offsets to direct, real expenses. By contrast, e-mail is a highly decentralized service, with correspondents' private systems contacting each other directly, rather than having to be mediated by state-regulated utilities. If additional fees are charged, they also need to be based on the costs of real services; an arbitrary "tax" will simply create its own problems. For example, who gets the money, and why?

To retain its flexibility and its ability to support new human communication uses, we must retain the current, open model of spontaneous e-mail exchanges. Therefore, over time, it is likely that Internet mail will evolve into two logical subsets. One comprises trusted, accountable participants and the other includes everyone else. Trusted participants may be subject to less stringent checks and filtering. Perhaps more importantly when there is a problem, it is likely that mail from a trusted identity will still be delivered, while the origination agent is consulted, rather than rejecting the mail automatically.

E-mail Architecture

Internet mail is based on a simple model. It distinguishes the world of users from the world of transmission. Anyone may send a message to anyone else. The basic service does not have a central authority and does not require authentication by the Originator, the Recipient or the operators. (It is worth noting that the telephone and postal services usually do not authenticate those sending letters or making calls.)

As shown in Figure 1, this model has grown to distinguish:

- *Mail User Agents* (MUA), which represent end-users
- The *Mail Transfer Service* (MTS) comprising a sequence of one or more *Mail Transfer Agents* (MTA), using the *Simple Message Transfer Protocol* (SMTP)[2,3]
- Posting new mail via a *Message Submission Agent* (MSA)[7]
- A *Notification Handler* or *Bounce Handler*, is an MUA that processes returned transmission reports such as a notice about failure. The Handler's address is specified by the MSA, during message posting.[11]
- Delivering mail via a *Message Delivery Agent* (MDA), possibly with user-specific delivery behaviors[8, 9]

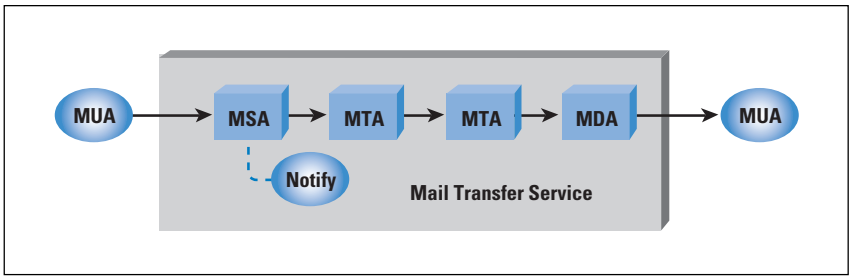


Figure 1: Internet Mail Architecture

The purpose of e-mail is to exchange messages among MUAs. For users, their e-mail client—the MUA—is all they directly experience. For most network administrators, the MTS software is their scope of concern.

The core e-mail message object also has a simple framework. Its *content* comprises:

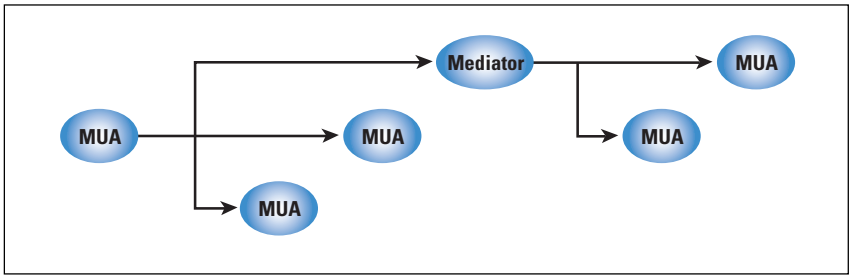
- Structured, textual meta-information, called the *header*, including *fields* for addressing, posting date, unique message identifier and a free-form description of the content[4,5]

- Lines of free-form ASCII text, called the *body,* which has evolved to support a potentially complex, structured set of multi-media, multi-character set attachments[12]

Figure 2 demonstrates a simple user-to-user example, with a message sent to three addressees, one of which is a special MUA that re-mails it to two additional recipients. The purpose of the Figure is to emphasize the user-to-user nature of e-mail and to provide a basis for considering the combinatorial explosion that marks the aggregate interactions of Internet mail components even in very simple uses. It further introduces another architectural construct:

- A *Mediator* is an MUA that re-posts messages, such as for a mailing list.[10] It preserves much or all of the original message, including author address, but can make substantial changes or additions to the content, which an MTA cannot. Therefore, a Mediator's role is user-level content responsibility, rather than MTS-level transit responsibility.

Figure 2: Simple Multi-Recipient Scenario

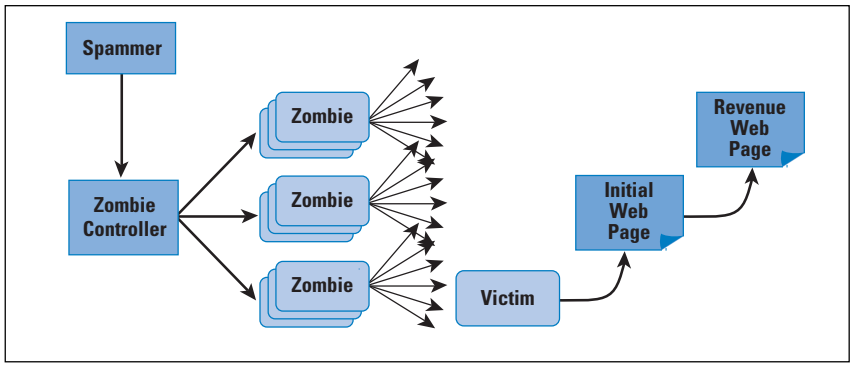


Spamming Architecture

Some spammers are legitimate businesses, engaged in overly aggressive marketing efforts, because there are no formal limits on their actions. In spite of the challenges created by needing to work at an international level, there is a reasonable expectation that legal strictures, both laws and contracts, will constrain in these businesses to a tolerable level. In contrast, *rogue* spammers actively seek to avoid accountability, to subvert barriers to their traffic, and to acquire unwitting and unwilling participation of machines owned by others. Independent of the legal details, the best social model to use for analyzing this latter group is crime. Often the activities do not violate particular laws, but what is most important is that the style of a spammer's conduct is the same as that of a criminal.

Unfortunately, the technical and operational world of spamming has also developed in scale and sophistication. Spamming used to entail one sender and one sending machine. Its performance was limited by the capacity of that machine and the bandwidth of its Internet connection. Today, rogue spammers control vast armies of compromised systems, called *zombies,* as shown in Figure 3. Zombies are owned by legitimate users who are unaware that their system has been compromised and is being used for spamming.

Figure 3: Rogue Spammer Control Network



The community of rogue spammers is remarkably well organized; it has become an extensive, underground economy. Some participants specialize in developing methods for breaking through filters. Others take over machines and turn them into zombies. Others sell the use of a zombie collection for periods of spamming. The estimated number of zombie systems is in the many tens of millions. After spam delivery, recipients often "click" to a transaction Web page. Web hosting is provided at multiple levels, in order to obscure the server side of the process, further reducing accountability.

Typically, spammers have the classic goal of selling products. However, they also can have political or religious motivations or even blatantly criminal intent, such as extortion. The ability to send very large number of messages to a specific destination gives spammers a tool that can be used to threaten an organization with a denial of service attack on their network.

Practical Efforts at Spam Control

It is tempting to believe that spam is an easy problem to solve, but history teaches us to be cautious. A web page located at **`http://craphound.com/spamsolutions.txt`** takes an irreverent approach in challenging simplistic proposals, by providing a checklist for the common weaknesses. In spite of its apparent whimsy, the checklist is surprisingly useful for screening proposals quickly.

The most common mechanism for spam control is a localized mechanism, the "filter"[14], named for its conditionally permitting mail to flow through it. Filters typically are used within the recipient's network (or Administrative Management Domain, as described later in this article.) However they may be placed anywhere along the path, notably including the MSA. Filters at the reception side cannot reduce Internet spam traffic. At the outbound side, they can. Filters have choices in the way they treat suspect messages. They can:

- Add a special annotation to the message
- Divert it into special storage
- Reject it back to its Handling Notification (RFC 2821 **`MailFrom`**) address or to the Client SMTP during the transfer session
- Simply delete it
- Accept it slowly, with "traffic shaping," to control the rate of SMTP transmission

The difficult question is: What are the criteria that a filter should use? The difficult answer is: Many. This need to support a wide, and changing, variety of decision criteria has caused filtering engines to evolve into extensible platforms for spam detection and handling modules. As the mixture and complexity of filtering algorithms become more sophisticated, the overhead they entail has grown substantially larger.

It is convenient to divide techniques into three, basic classes of criteria, although each is complex:

- *Content analysis,* such as Bayesian statistics tracking of vocabulary and content hashing, to detect bulk duplication
- *Responsible Agent assessment,* either for permission (whitelist) or rejection (blacklist)
- *Traffic analysis,* such as rates at which messages come from the same author address or IP Host Address

Content analysis is always a matter of partial success (and partial failure.) It is usually statistical and depends upon a database of training messages, to establish vocabulary norms. Spammers are constantly developing techniques for bypassing the current analysis technologies. Further, different recipients on the same e-mail service can have wildly different statistical patterns of acceptable content. This makes fine-grained filtering by their service provider problematic.

It is clear that these tools for evaluating individual messages, or aggregate traffic flow, can have significant transient utility. However they cannot be effective, long-term tools, even with continuing enhancement. Notably they have little or no effect at reducing spam at its source. These post-hoc analysis tools have two inherent deficiencies, both of which are coupled to their using heuristics, rather than reliable, accurate and objective rules. The first is one of "false positives" in which legitimate mail is incorrectly labeled as spam. As an example, this could mean that an essential business transaction is not delivered, instead being classed as junk mail. Perhaps the most insidious example of this problem occurs when spammers send mail that purports to be from a well-known, legitimate business. This is called *phishing* and results in making *all* mail with the address suspect, so that legitimate postings of essential mail are not delivered.

The second problem with using heuristics is in the nature of an "arms race" between spammers and anti-spammers who must each constantly adapt techniques, consume more resources, yet never win. It does not help that those fighting spam have been losing the war, since spammers have tended to be more aggressive, more innovative and better organized…

A different line of effort is based on the social assessment that the sender of an e-mail should be held accountable for it. The goal is to identify such an agent and then evaluate the agent's acceptability. This approach requires three enhancements to Internet mail:

- A clear sense of the boundaries between independent operational authorities
- A means of verifying an accountable identity that is associated with the message
- A means of formulating and sharing assessment information about accountable identities

Although e-mail operators often refer to *boundary* MTAs that face the open Internet, there is no accepted term for a region of e-mail components under unified authority. This article suggests a term derived from the OSI X.400 e-mail effort: *Administrative Management Domain* (ADMD) to mark these trust boundaries. They distinguish a collection of operational components subject to the same administrative policies, as discussed in [13].

An example of ADMDs is shown in Figure 4, and is derived from the scenario shown in Figure 2.

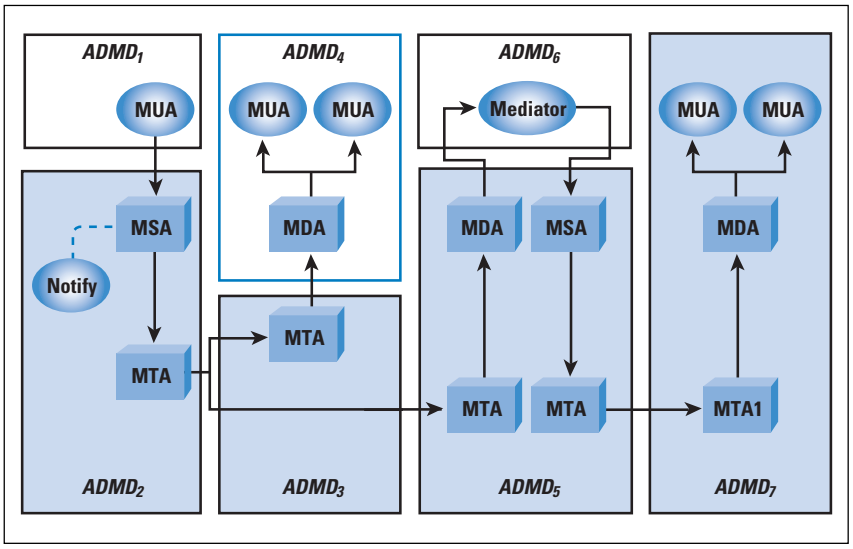


Figure 4: Independent Administrative Management Domains (ADMD)

The implied complexity of responsibilities and interactions is striking, even for this relatively modest case. For simplicity, think of the ADMDs labeled at the top of the Figure as representing users or value-added services, whereas the ADMDs labeled at the bottom could be a variety of classic Internet service (access) providers. The "boundary" agents are the ones with lines connecting over to another ADMD.

The increased diversity among Internet participants and ADMDs results in abuses such as spam. Proactive efforts to deal with these abuses require that we make changes in the nature of the trust between ADMDs and the way that that trust is enforced.

Accountability

Agent assessment seeks to hold an entity (agent) accountable for problematic e-mail. Who is a responsible agent for the content or for injecting the message into the MTS, and are they assessed as trusted or problematic?

There are two broad classes of accountable entities:

- *Content agents* comprise authors (RFC 2822 **`From`**) and those who are responsible for posting individual messages, as specified in the RFC 2822 **`Sender`** field. If the content agent is validated for a message, then the content probably reflects their intent. That is, it is unlikely that some other entity changed the content. Because the Notification Handler address (RFC 2821 **`MailFrom`**) appears in the SMTP protocol but is associated with the posting agent, it is often considered useful for analysis. Unfortunately the address often has no obvious relationship to the From field author or the Sender field posting agent, so its use for filtering can be problematic. However spammers often specify false Handling Notices addresses, in order to direct the mass of failed deliveries elsewhere. Consequently, it can be useful to validate the **`MailFrom`** address.
- *Operations agents* provide MTA or basic Internet access services. They are often held accountable for the impact of the bulk traffic their systems generate. Although they do not create the content, it is possible for them to enforce strict rules on their customers and to detect patterns of violations among them. Recommended practices for operators are beginning to obtain some consensus, such as with [15]. More are needed.

Assessment of agents can be proactive or reactive:

- *Accreditation* is the proactive registration by a sender, who aligns with a registry that extracts quality assurance commitments; any trust of the sender is therefore inherited from trust of the accreditation agency.
- *Reputation* refers to reactive evaluation of a sender's prior postings; for these, independent third parties evaluate the sender's history.

The functions that are combined, to establish useful accountability, comprise:

Identification: An identity label provides a unique reference to an entity.

Authentication: Validates the use of the identity label.

Authorization: Determines that the user associated with the identity is authorized to perform a particular function.

Assessment: Obtains an analysis of the trustworthiness or "quality" of the agency that is providing the authorization, or of the validated entity itself.

Unfortunately, many identities are involved in e-mail creation or transmission, as shown in Table 1.

Table 1: Roles for Internet Mail Identities

Type	Provided by	Identity of
MTA IP Host Address	Network-level service	SMTP client
EHLO Domain Name	RFC 2821 SMTP command	SMTP client
MTA Provider's IP Network Address	Network-level service	Site of SMTP client
Mail-From Mail Address	RFC 2821 SMTP command	Handling notices
From Mail Address	RFC 2822 header field	Author
Sender Mail Address	RFC 2822 header field	Posting agent
Received Domain Name	RFC 2822 header field	Relaying MTA site

Relative to an SMTP Server that is being asked to accept a message, the SMTP Client is an agent of the operator of the previous hop. Since the e-mail operator might be different from the operator of the IP access network that is hosting the e-mail service, it might entail a different identity. This highlights an interesting aspect of Table 1: Most of the identities associated with e-mail handling can be called "the sender." Consequently, that term has become nearly meaningless, in anti-spam discussions.

Because identity listings are made explicitly in a database, they are capable of producing almost no false positives, although there might be many identities not listed and a listing might be inaccurate. Still, there are significant challenges with the use of identity-based filtering:

- Which identity should be used and how does it relate to spamming behaviors? Note that Table 1 listed quite a few choices. In addition an author can create bad content, but the identity listed in the RFC 2822 **`From`** field of that content might not be the actual author, even if that field is validated. The message might have originated on a compromised machine and used the identity associated with it, unbeknown to the owner of the machine. Also the operator of the mail-sending network might have nothing to do with creating content, but it might be reasonable to hold the operator accountable for aggregate traffic problems.
- How is the identity validated (authenticated)? What entity is doing the validation? How does it relate to the identity being validated? And why is it trusted? Can the validation mechanism, itself, be tricked?
- How is an identity determined to be a spammer or non-spammer? What entity is vouching for the quality of that identity and why is the vouching entity trusted?

Authentication Standards

Accountability requires having an accurate, reliable identity of the agent that is to be accountable. Authenticating an identity is, therefore, a prerequisite for assessment efforts. However it does not, by itself, ensure a positive assessment. Spammers can register and authenticate their identities, too.

Early anti-spam identity schemes use the IP Address of the client SMTP MTA that is sending directly to the server running the filter. The Address is provided by the underlying network service, and therefore has been trusted. However, spammers are becoming proficient at stealing IP Address space, such as by advertising routes that use allocated-but-unused blocks of IP Addresses! Also an IP Address changes as the host changes its attachment to the Internet, and it is affiliated with operators, not authors. This makes the IP Address obscure and unreliable, when attempting to assess e-mail.

A more recent focus is on the use of Domain Names, for references that are more stable and align better with the authority boundaries of Administrative Management Domains. Broadly there are two lines of effort at using Domain Names for validating messages being relayed. One associates the identity with the systems that handle the message along its path. These "path registration" schemes include Sender Policy Framework, Sender-ID, and Certified Server Validation. The other schemes tie a Domain Name identity to the message object. These include DomainKeys Identified Mail, and Bounce-Address Tag Validation.

The *Sender Policy Framework* (SPF)[16] has evolved over time, attempting to encompass multiple identities. It primarily uses the Domain Name in the RFC 2821 **`MailFrom`** command. It queries the *Domain Name System* (DNS) with that name and determines whether the IP address of the previous-hop MTA is registered under that name. Since any SMTP server along the transit path may choose to perform this query, SPF requires that the Domain Name contain a registration for every MTA along every delivery path for a message. (A common simplification for this model is to use it only between boundary MTAs, but this considerable constraint is not specified in SPF. Rather, its use is usually characterized as being more general.) Although the software overhead for SPF is quite small, the administrative overhead can become substantial, as the number of paths increase and as paths change. In addition, some sender SPF DNS configurations can trigger a very large number of queries per addressee. Lastly, the role of the RFC 2821 **`MailFrom`** command is to specify the Notification Handler address. This address might be entirely different from other origination information, making registration of all of the MTAs in the path problematic. SPF therefore has significant administrative problems with redirected traffic, such as when going through a third-party forwarding service.

Sender-ID (SID)[17] uses a model similar to SPF, but it is based on the posting address Domain Name in the RFC 2822 **`Sender`** field (or RFC 2822 **`From`** field, if no **`Sender`** field is present.) Both SID and SPF sought IETF standardization in 2004 but the working group effort failed, due to lack of rough consensus convergence among participants and due to concerns over intellectual property claims.

Certified Server Validation (CSV)[18] covers only the current client/server SMTP hop. The client specifies an operator's Domain Name in the RFC 2821 **`EHLO`** command. The server uses this name to query the DNS. It then validates the IP Address of the SMTP client and determines whether the Domain Name administrator has authorized the client to send mail. CSV also specifies a standard mechanism for querying an assessment service about the client's Domain Name.

DomainKeys Identified Mail (DKIM)[19] specifies an accountable Domain Name that applies to a message during transit. It uses public key cryptography to digitally sign the message and provides guidance when the signing Domain Name differs from the Domain Name in the RFC 2822 **`From`** field.

DKIM Domain Name validation represents a significantly different goal from that of the strong authentication methods, such as [20, 21] which focus on long-term protection of message content. Also DKIM places its parametric information in a special RFC 2822 header field, rather than in the message body, so that it does not have any impact on recipient user agents that do not support DKIM. Although public key cryptography has relatively high computational cost, e-mail processing is usually i/o-bound, so that the real-world use of DKIM appears to have little impact on the aggregate message-handling capacity of a server.

Bounce Address Tag Validation (BATV)[22] attacks the problem of misdirected handling notices, such as bounces. It permits the creator of an RFC 2821 **`MailFrom`** bounce address to digitally sign it. When the bounce agent of that creator receives a message purporting to be a bounce, the agent can validate the address. Standardization of its format is needed so that e-mail intermediaries—such as some mailing list software—can determine the "core" of the mailbox portion. Since the creator of the signature semantics is the only consumer of the signature semantics, any signature algorithm can be used, including one based on symmetric keys. For convenience—and an existence proof—the BATV specification provides an example algorithm already in use.

Collaboration Support

Fighting spam must be a collaborative effort, which will benefit from using tools and standards that aid in exchanging information and performing coordination. To this end, standard methods of reporting spamming events, of characterizing particular spam, and of sending spam control data can be helpful. Some work in that direction is already underway.[23] Fighting spam requires global operations collaboration; this will be aided by services to facilitate interactions between network administrators speaking different languages. It is also likely that there should be standards for the syntax and semantics of whitelists and blacklists.

Acknowledgement

The author wishes to express particular appreciation for the unusual amount of dialogue that took place with the reviewers of this article. It produced a substantially clearer and more concise article. It also highlighted the extraordinary diversity of views on the topic, in case one had had any doubt. In fact, the article by John Klensin which follows this one is a direct result of the dialog.

References

[1] Hoffman, P. and D. Crocker, "Unsolicited Bulk Email: Mechanisms for Control," Internet Mail Consortium, UBE-SOL IMCR-008,
http://www.imc.org/ube-sol.html, revised May 4, 1998.

[2] Postel, J. B., "Simple Mail Transfer Protocol," STD 10, RFC 821, August 1982.

[3] Klensin, J., "Simple Mail Transfer Protocol," RFC 2821, April 2001.

[4] Crocker, D.H., "Standard for the format of ARPA Internet text messages," STD 11, RFC 822, August 1982.

[5] Resnick, P., "Internet Message Format," RFC 2822, April 2001.

[6] Crocker, D., "Internet Mail Architecture," Internet Draft, **draft-crocker-email-arch**, April 2005.

[7] Gellens, R. and J. C. Klensin, "Message Submission," RFC 2476, December 1998.

[8] Myers, J. G. and M. T. Rose, "Post Office Protocol – Version 3," STD 53, RFC 1939, May 1996.

[9] Crispin, M., "Internet Message Access Protocol – Version 4rev1," RFC 3501, March 2003.

[10] Chandhok, R. and G. Wenger, "List-Id: A Structured Field and Namespace for the Identification of Mailing Lists," RFC 2919, March 2001.

[11] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)," RFC 3461, January 2003.

[12] Freed, N. and N.S. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies," RFC 2045, November 1996.

[13] Clark, D., Wroclawski, J., Sollins, K., and R. Braden, "Tussle in Cyberspace: Defining Tomorrow's Internet," ACM SIGCOMM, 2002.

[14] Showalter, T., "Sieve: A Mail Filtering Language," RFC 3028, January 2001.

[15] Hutzler, C., Crocker, D., Resnick, P., Sanderson, R., and E. Allman, "Email Submission: Access and Accountability," Internet-Draft, **draft-hutzlerspamops-05**, October 2005.

[16] Wong M., Schlitt M., "Sender Policy Framework (SPF) for Authorizing Use of Domains in EMAIL, version 1," Internet Draft, **draft-schlitt-spf-classic-02**, June 2005.

[17] Lyon J., Wong M., "Sender ID: Authenticating Email," Internet Draft, **draft-lyon-senderid-core-01.txt**, May 2005.

[18] Crocker D., Leslie J., Otis D., "Certified Server Validation (CSV)," Internet Draft, **draft-ietf-marid-csv-intro-02**, February 2005. Also see: **http://mipassoc.org/csv**

[19] Allman E., Callas J., Delany M., Libbey M., Fenton J., Thomas M., "DomainKeys Identified Mail (DKIM)," Internet Draft, **draft-allman-dkim-base-00**, July 2005. Also see **http://mipassoc.org/dkim**

[20] Ramsdell B. (ed.), "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification," RFC 3851, July 2004.

[21] Elkins M., Del Torto D., Levien R., Roessler T., "MIME Security with OpenPGP," RFC 3156, August 2001.

[22] Levine J., Crocker D., Silberman S., Finch T., "Bounce Address Tag Validation (BATV)," Internet Draft, **draft-levine-mass-batv-00**, September 2004. Also see **http://mipassoc.org/batv**

[23] Shafranovich, Y., "An Extensible Format for Email Feedback Reports," Internet Draft, **draft-shafranovich-feedback-report-01.txt**, May 2005.

Ed.: This article is a revision of "Adapting Global Email for Controlling Spam," in *Information Processing Society of Japan (IPSJ) Magazine—Special issue on Anti-Spam,* Japanese/English, Volume 46, No. 7, pp. 741–746, July 2005.

DAVE CROCKER is a principal with Brandenburg InternetWorking. He has authored or contributed to most Internet mail standards, and an assortment of e-mail products and businesses, as well as working on facsimile, security, ecommerce and EDI. He received the 2004 *IEEE Internet Award* for his work on e-mail. Dave is a contributor to the development efforts for DKIM, CSV and BATV, motivated by a strong desire to protect more than 30 years of professional investment that is being threatened by spamming. E-mail: **dcrocker@bbiw.net**

Taking Another Look at the Spam Problem

by John C. Klensin

The problem of unsolicited bulk e-mail on the Internet has been widely discussed, and many classes of solutions have been proposed. Dave Crocker's article discusses some of the background for the solutions generally, points to a semi-humorous list of ways in which proposed approaches fail, and compares several approaches based on source authentication. This article takes a somewhat contrarian view. It argues specifically that the traditional models for defining technological solutions and then letting the policy and legal communities work out the details of how to utilize them are seriously wrong in this particular case and that partially-effective methods of fighting spam actually cause more spam.

This article makes two main suggestions. First, attempts to design technological countermeasures to spam without a clear understanding of how far, and in what directions, the setters of social policy are willing to go are futile. The requirement is not just that there be social recognition that a problem exists. In order to design effective technological countermeasures with predictable and acceptable side-effects, we must first understand what measures society is willing to take—what laws it is willing to pass and enforce to make spam a criminal or civilly-punishable act—to set an appropriate context and set of boundary conditions. Without those conditions, design of technological countermeasures is likely to constitute poor engineering practice, not just futility. Second, deployment of spam counter-measures that are not completely effective largely shifts the burdens of spam from one recipient population to another while *increasing* the total amount of spam on the network.

His analysis and mine agree on several critical points. Solutions that discard important characteristics of today's e-mail environment permanently in order to make some short-term gains against spam are not acceptable. Approaches that require drastic and simultaneous changes to the ways in which e-mail works in order to function are not going anywhere. There is a difference between legitimate businesses who have decided, within the limits of existing legislation, to engage in mass, unsolicited, electronic mailings to promote their products and those bulk mailers who prefer to cover their tracks, hide linkages between sending addresses, hosts, and web sites (or create deceptive ones), and who use zombie mailers and other ways to avoid cost and detection. We also agree that spammers, or their tool suppliers, are creative, technically-knowledgeable, and able to react much more quickly than the spam-fighting community (especially the standards-based part of that community) to changes in operating conditions and countermeasures.

I suggest a further guideline to help us think about the problem: however small they might be on a per-message basis, there are costs associated with sending e-mail and costs associated with receiving it and eliminating undesirable content.

If an anti-spam "solution" is developed that permits the spammers to vastly increase the costs to the recipients without a proportionate increase in their own costs, that solution is not tenable. A serious effort to predict the impact of a proposed solution to spam, including costs to the end user and load on the network as the spammers adapt to it, should be a critical component of such efforts. But, while equivalent analyses of measures, likely responses, and countermeasures are standard with any (other) technique designed to enhance network security, they have been largely absent when new technological approaches to spam are proposed.

This is a different aspect of the so-called "arms race" problem. In a classic arms race, no one can really win, as Dave points out. But, more important, when such races stop, it is only because one party simply stops, is forced out of the game by external pressures, or becomes exhausted economically. As long as there are no economic constraints, every escalation is met with a counter-escalation, which is met with a counter-counter-escalation, and so on. It is this positive feedback cycle that characterizes a true arms race. The battle against spam demonstrates a particularly unfortunate variation on that pattern in which the incremental economic costs of trying to deploy new spam abatement measures appear to be much more severe than the costs to the spammers of the most obvious counter-measure to improved spam abatement procedures, simply sending out more traffic. This is discussed further and in context below.

Social Problems and Technological Solutions

In the technical and protocol design community, our normal model is to develop technology and then use it to inform the policy, social, and legal parts of the society who then need to sort things out on their side. One of the classic arguments for this approach, which does not seem relevant to the spam situation, is that the potential use or misuse of a technology will not, and should not, constrain its development. For spam, the situation appears to be exactly reversed: we need to understand what is feasible and plausible from social, political, legal, and regulatory standpoints in order to define the engineering solution space. If we do not know what behaviors society is willing to make illegal or subject to effective civil action and whether it is willing to enforce those laws or equivalent positions, we cannot adequately define the engineering solution space. That results, in turn, in a high risk of solving the wrong problem or an irrelevant one. Of course, recent history has shown a variety of irrelevant and costly solutions to spam proposed, and sometimes deployed.

The solution to spam is identical to the solution to most other significant social problems: society must determine that it is a problem, create effective rules prohibiting the problem, and then enforce those rules aggressively and consistently. Technical solutions that make it easier to identify spam and its sources can then be immensely useful, but they are only useful if designed to be effective within the framework set by those rules.

If, by contrast, societies are, in practice, unwilling to take effective social or legal action against spam and those who benefit from it, then this article suggests that anti-spam measures will tend to make the overall situation worse.

The question of spam beneficiaries provides a particularly good illustration of this point. So far, most legal systems in the world have taken the position that the act of spamming is the offense (if there is any offense at all). Operating a domain or web site to which the spam recipient is directed to buy a product or obtain another benefit is rarely considered a problem by either law enforcement or by the relevant ISP. While establishing cause and effect—that the spam was authorized or encouraged by the web site owner—can be quite difficult, there has, appropriately, been little examination of tools to detect or identify beneficiaries because doing so seems pointless. On the other hand, on the same theory that it is more useful to try to arrest the drug importer than the street dealer, a different set of laws about beneficiaries and spam-authorizers—those who, in at least some cases, pay the spammers to spam—might dramatically change the landscape.

Reducing Spam by the Percentages

A new technique or group of techniques that claims to be beneficial can have either positive or negative value with regard to the amount of spam that gets through, either overall or to the mailbox or a particular sample user. A technique can also result in significant increases in the amount of network bandwidth or server resources consumed if it is neutral or better with regard to the end user mailbox. As long as the spammers can increase the number of messages they send out, almost arbitrarily and at low or zero marginal cost, the percentage of spam that is filtered out is ultimately irrelevant. The key measurement is how the amount of spam that gets through to some exemplar user (or a statistical aggregate of them) changes. That change pattern can be net either positive or negative. Suppose a technique is introduced that causes an initial small incremental reduction in the amount of spam delivered. The patterns of the last several years suggest that the spammers will respond by making a large increase in the amount of traffic they send out. Since the costs of doing so are very low, it would arguably be irrational for them to do anything else. If the increased volume is enough larger than the amount of spam the new technique was able to stop, there is a net loss to the Internet overall: the small improvement may represent a percentage decrease in the amount of spam that gets through, but the amount seen by the representative user increases and the percentage claims are largely irrelevant.

Unless whatever methods that are used in an attempt to reduce the amount of spam actually stop it at, or very near, the point of origin, the net effect on users is to shift the amount of spam received from those who have deployed the latest and most effective countermeasures to those who have not yet done so. The total amount of spam-related traffic on the network just continues to rise. And, since most countermeasures have costs—either in processing time or in software licensing fees—the cost burdens on end users also continue to rise.

This would seem to argue for methods that cut off spam traffic close to the source, but attempts to design such methods have been fairly unsuccessful, sometimes because of another policy problem: the spammers argue that some people like receiving unsolicited bulk commercial e-mail so that cutting off bulk traffic near the point of origin prevents legitimate and desired traffic from transiting the network. Source-oriented techniques include not only technical approaches but efforts—by law or social pressure—to hold ISPs and mail providers responsible for all traffic emanating from their networks, thereby encouraging them to refuse to have spammers as customers, to aggressively enforce terms and conditions of service, and so on. The strongest advocates of the "blacklist" variation of those techniques continue to claim that they are very effective although some others in the community are not completely convinced.

The House-Burglar Analogy

In the absence of a coordinated approach that is oriented toward legal or social enforcement, most anti-spam techniques appear to induce more spam on the network. They do this by making simply sending much more traffic out the most rational behavior for a spammer who is faced with an abatement technique to adopt. They may enable shifting the burden of dealing with that spam from one person to another—in the same way that aggressive locks and alarm systems on one house slightly increases the relative burglary risk to the less-protected neighbor—but, as Dave's article points out, we have no realistic plan for making it too expensive for the spammers to simply increase output.

Deterrents to burglary work moderately well because they increase the costs (in time, sophistication of the required tools, and so on) to the burglar. Equally important, they increase the risks of being caught and punished. In the present spam environment in most countries, we have no effective mechanism to increase costs and, at least statistically, the odds of being effectively punished even if caught are insignificant.

Shifting Burdens and Creating Preferred Classes of E-mail

The argument Dave presents for authenticated mail is ultimately that it can get expedited handling while non-authenticated mail is put aside for other methods of spam detection. That approach could be immensely effective at expediting receipt of some mail by the recipients who apply the needed checks, at least until the spammers begin authenticating their mail in a way that tricks the trust-establishment techniques. Prioritization of some messages and content will be effective as long as the fraction of such messages remains relatively small relative to the total number of messages received. As the percentage rises, one probably ends up either trusting all mail from a particular source, regardless of the author, or with a situation quite analogous to "whitelists," although one that is much harder to trick than the original. Either is subject to attacks and scaling problems.

There is also the risk of abuse by providers who conclude that mail that cannot be authenticated well enough that their users can prioritize it should simply be rejected and who then define the conditions for adequate authentication in terms of a small circle of cooperating mail providers. Even if the types of authentication outlined in Dave's article are used only as intended, the costs to recipients will rise, perhaps rapidly, over time as percentages of messages bearing authentication information rises and sender authentication and authorization become just one more tool to distinguish probably-desired messages from probably-undesired ones.

Maybe there is not Enough Spam Yet

One of the depressing consequences of the reasoning discussed previously is that perhaps we have yet to see sufficient spam for governments and regulatory bodies to take the spam problem seriously—seriously enough to deploy effective laws and enforcement mechanisms. If spam-fighting methods shift the burdens of receiving spam away from those who have the resources to protect themselves they may simply place the spam impacts on others who have fewer resources. That pattern may, in turn, also reduce pressure on governments to take effective action and to do so in a way that would make the design constraints for effective technological approaches clear. If a collection of anti-spam methods have the effect of simultaneously increasing the amount of total spam on the network and of decreasing pressures on societies and governments to take effective action, are they really ones we want to deploy?

Conclusions

This article presents a rather grim view of the future if we continue on our present course. If we fail to examine the actual actions that societies and their governments are willing to take to deal with spam and spammers and to treat those actions and their limitations as design constraints on the technical and engineering approaches, we are likely to continue to see an ever-increasing amount of spam on the network. Spammers will not only adopt technical countermeasures to new techniques but they will also take advantage of their ability to simply increase message volumes (at almost no cost) to counter the effects of those techniques on the percentage of spam that is delivered. It may be time to finally deal with the spam problem as the difficult social issue that it is, rather than permitting societies and governments to continue to believe that a technological "silver bullet" is right around the corner and that no real social or political action, or commitment of law enforcement resources, is needed.

JOHN KLENSIN is an independent consultant based in Cambridge, Massachusetts. He has been involved in the design, development, and deployment of ARPANET and Internet applications, and occasionally lower-layer technologies, since the late 1960s and early 1970s. He has also been intermittently involved with Internet administrative and policy issues since the early 1980s. His current work primarily focuses on internationalization of the Internet on both technical and policy dimensions. E-mail: **klensin@jck.com**

Caveats in Testing Routing Protocol Convergence

by Russ White, Cisco Systems

In general, the main problems we find when testing routing protocols lie in generating accurate (or rather, realistic) data, as well as understanding the limitations of tests geared towards measuring routing protocol performance. Three areas of specific interest are covered in this article: defining convergence, taking realistic measurements, and creating realistic data.

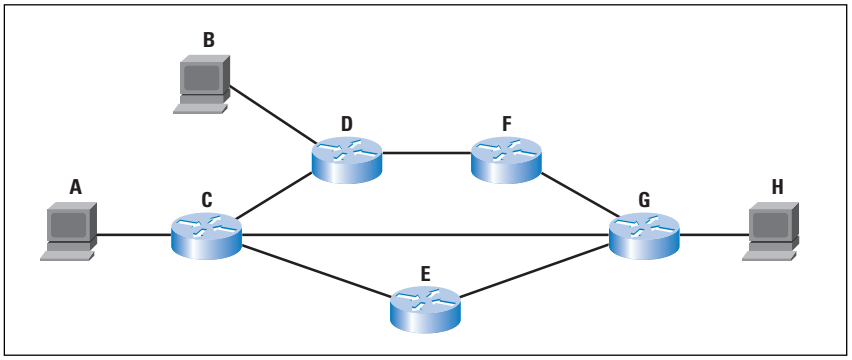
Defining Convergence

The first problem we face when trying to test routing is to define *convergence*. It seems like a simple question, but it's not, because there are so many different ways to measure convergence:

- How long does it take to begin forwarding traffic once a topology change has occurred?
- How long does it take for every router in the network to adjust to a topology change that has occurred?
- How long does it take for the forwarding information on a specific router to be updated once a topology change has occurred?
- How long does it take for the routing protocol to adjust to a topology change?

Each of these questions is actually completely different, as a short examination of the network in Figure 1, below, shows.

Figure 1: Test Network



Assume A is the traffic source for a test, and H is the sink, or the convergence measurement point. To measure the convergence time of this network, you send a stream of traffic from A to H; when the traffic stabilizes, the C to G link is taken down, and the length of the gap in traffic at H is measured. In this environment, we assume the path fails off of the C to G link, and onto the path through E.

This test assumes the traffic between B and H, or between A and B, will not be impacted by the link between C and G failing, but we do not know this will always be the case. In fact, it's possible that D and F will end up forming a *microloop* until they receive all the information needed to converge without the C to G link.

This microloop could last longer than C requires to recompute a path to H, so while the traffic from A to H may be successfully delivered, the network may not be in a fully converged state. The topic of microloop formation and avoidance is beyond the scope of this article.

In this small network, the time it takes for A to continue forwarding traffic to H may not be the same as the time it takes for the entire network to stabilize after the topology change. How long it takes for A to be able to reach H, and how long it takes for all the routers in the network to adjust to the topology change are two different questions. In this case, the concept of convergence is unclear, with several possible meanings; to properly build and understand the results of the test, we need to better understand the question being asked.

You could alter the test so only A, C, E, G, and H are in the network. This would provide a "clean" test of just the failover capabilities of the routing protocol being tested, as it's implemented on the specific routers in the network, across the specific link types connecting the routers, in the simple failover situation. While the limited topology does limit the number of outputs being measured in the test, it also limits the closeness of the tested network to a real network design. The test can provide some very specific data points, but, once the test topology is simplified, it cannot provide a true picture of convergence in a larger, more complex topology.

Another option is to refine the test procedure so the traffic between B and H is tested as well as the traffic between A and H. Measuring traffic flow from every possible connected end point to every other possible connected end point on the network provides a number called *goodput,* which is the relation between the traffic injected into the network versus the traffic the network delivers across all paths.

Although this type of testing does provide more data in a more complex topology, it also has its drawbacks. For instance, if you are trying to compare two different implementations of a single protocol, or compare two different routing protocols, this test not only counts the amount of time required for the routing protocol to converge, it also tests the amount of time required to note the topology change, the time required to install the newly computed routes into the local routing table, and the time required to pass the changes from the routing table to the local forwarding tables. This might—or might not—be a good thing.

Isolating just the routing protocol can provide information about the performance of a specific implementation of the protocol in specific network designs, and under certain conditions. Including platform and media-specific issues—such as the installation of information into a local table—may cloud the picture. For instance, if the routing protocol can converge in milliseconds, but it takes seconds to determine that the link between C and G has failed, any changes in routing protocol convergence time will be lost in the much larger link failure detection time, reducing the value of the test.

In short, numerous tradeoffs are involved in designing a test to measure routing protocol convergence; you need to begin with the right questions, and understand the tradeoffs in the various tests you could, or might, run. There's no "simple" way to run a single test that will give you all the information you need to know to understand all possible implementations of a routing protocol on all possible platforms.

In the same way, it's important to keep these types of limiting factors in mind when reading, or using, test results provided by outside companies. It's fairly easy to look at a specific test for one measure, such as the number of neighbors a specific implementation of the *Border Gateway Protocol* (BGP) can support in specific conditions, and attempt to generalize those test results to much larger and varied real world networks. Quite often, the mapping isn't all that simple.

Taking Realistic Measurements

Assume you determine you want to test for protocol convergence by checking the routing tables at each router in the network in Figure 1, rather than trying to measure convergence by measuring traffic flow through the network. How would you go about doing this? There are two general types, or classes, of tests, that you could consider:

- *Black Box:* Treat the device as a black box, only using outside signals and controls, and never any output provided from the device itself.
- *White Box:* Use available output provided from the device itself, possibly with tests using signals outside the device, to determine when specific events on the device occur.

Obviously, black box testing is much more difficult, maybe impossible in some conditions, but, at the same time, can provide more "objective" measures of a devices' performance. Examples of black box tests for the *Open Shortest Path First* (OSPF) protocol are outlined in RFC 4061, RFC 4062, and RFC 4063. White box testing typically depends on *debug* and *show* commands to provide timestamped infor-mation about when specific events occur, such as when the routing protocol has received information about the topology change, when the routing protocol has finished computing the best path to each destination, and other events.

For simplicity, the network is reconfigured with a test measurement device, as shown in Figure 2, below.

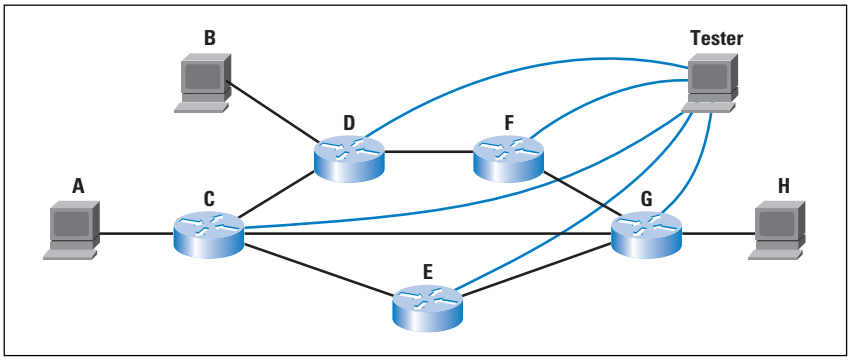


Figure 2: Reconfigured Test Network

Some mechanism is used to determine when the routing protocol on each router has computed the correct routes; the network is connected, and allowed to converge. The link between C and G is taken down, and the time between the link failure and the correct routes being computed on C, D, E, F, and G is taken as the total convergence time in the network. This appears to be a straight forward test; what sorts of problems can we run in to here?

There are two possible mechanisms for determining when each device has correctly computed the routes after the C to G link fails:

- Some sort of "continuous output," such as a *debug*, can be configured on each router, and the results collected and analyzed.
- The Tester can poll each device, using *show* commands, or some black box testing technique, to determine when device has recalculated the routes correctly.

Let's examine each of these techniques separately.

Gathering Results from Continuous Router Output

The first, and simplest, mechanism is to gather the results from each router through debugging information provided by the protocol implementation which is generally used for troubleshooting and monitoring the routing protocol. There are three primary issues related to using this information you need to be aware of:

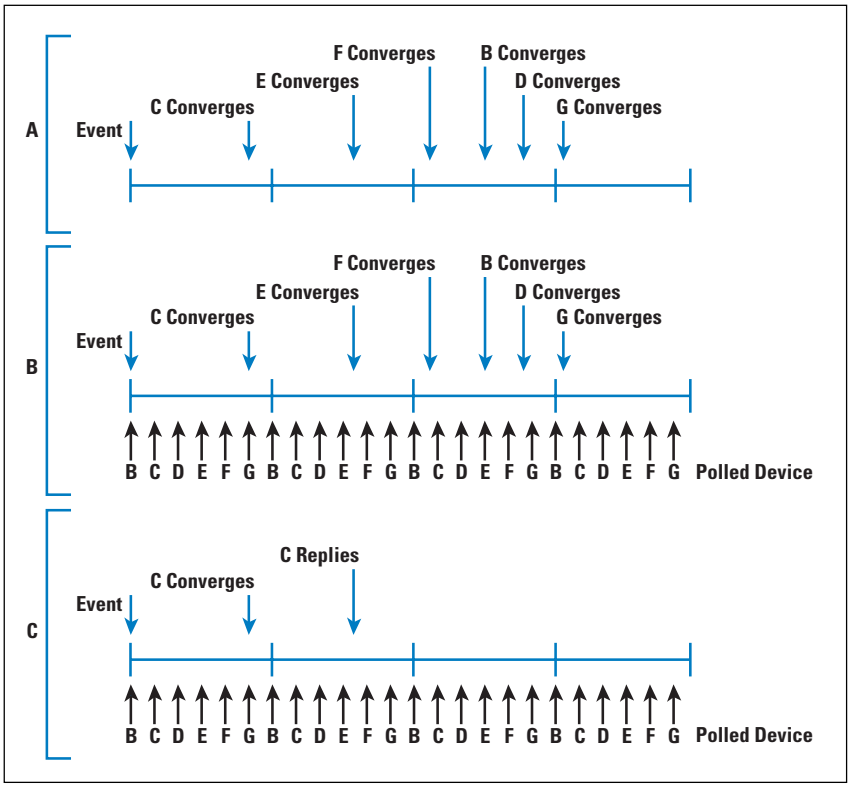
- The continuous stream of information provided by the device being tested can actually impact the test results, primarily because of the processor cycles required to record and display this information. In some situations, the additional cost is negligible, and in others, it's simply not important (for instance, if the test is designed to show the differential between two situations, rather than provide absolute convergence times).
- If the timestamps injected by the devices being tested in the network are relied on, then the time clocks of every device must be synchronized. This synchronization must generally be within about 1/10th or less of the total variation in the test time for the results to be meaningful. In other words, if the timeclocks on all the devices are synchronized within one second of each other, and the results of the test are expressed in milliseconds, the actual test results are going to be lost in variations in the synchronization of the timeclocks.
- If the devices feed their information to the Tester, and the timestamp on the Tester is used to compare the event times within the network, the timestamps can be skewed by the packet processing requirements of the devices, as well as queuing delays in the Tester. Most routers prioritize routing traffic over switched traffic, and switched traffic over management traffic. There could be significant lags between an event occurring, and the router actually building a packet noting the occurrence of that event. Again, this is a matter of time differentials; if the test results are expressed in milliseconds, queuing delays alone can bury the results in noise.

We need to be careful when using *debug* or other continuous output to measure network convergence times in any given test, then. Quite often, we need to compare the granularity of the test results with the measurement technique used, and consider how much noise the measurement technique is actually likely to inject into the testing environment, compared to the test results granularity.

Polling Devices

Another common technique is to run some sort of process on the Tester which polls each device, either using some black box or white box measurement, to determine when each device finishes recalculating routes after the topology change has occurred. This type of test is also constrained by various factors that might not be obvious when you are designing a test, or examining the results of a test that uses it. Assume events in the network occur as Figure 3 illustrates.

Figure 3: Poll Testing Scenario



In Figure 3A, we assume that the Tester is able to poll every device in the network at the same time, once a second. The test shows the network converged at 4 seconds after the event, although the last router to converge, G, does so just after the 3 second mark. There can be a variation of the entire polling interval in the actual results without the test showing any difference in the convergence time of the network, implying that the polling interval must be much faster than the expected (measured) test results for the results to be meaningful. We normally suggest that the polling interval be about 10 times faster than the expected measurement rate, or that the Tester should poll every 1/10th of a second in this test, if the results are to be measured in seconds.

However, in real test environments, a test device cannot actually poll every device in the network at the same time. Instead, the Tester will poll one device periodically, rotating through the polled devices, so the longest time between any specific device being polled is the polling rate. We can call this rotating polling *serialization,* and the time it takes to rotate through all the devices the *serialization delay*. Here, we've spread out the polls across the total one second polling time, to illustrate, in Figure 3B. Three anomalies show up in this illustration:

- The total time for the network to converge is still just over three seconds, while the recorded test time is still in the four second range. This is similar to the problem we noted when we assumed the Tester was polling all the devices in the network at the same time.
- It appears, from our test results, that E and F have converged at about the same moment. In reality, their convergence is separated by almost one second. In some extreme cases, the devices may actually converge in the opposite order from the order they appear to converge.
- If the convergence order of D and G were to be reversed, the network would appear to converge almost a half a second faster, although the actual convergence time would remain constant. This could cause a widely diverging set of test results over multiple runs in what is, actually, a fairly consistent network convergence time.

Adding the serialization delay of polling isn't enough, however, to understand polling in real test environments. We also need to remember that each device which is polled must also answer each one of the polls, thereby introducing another variable amount of delay into the test results. For instance, in Figure 3C, C is polled once before and once after it converges. If we take the time that C answers as its convergence time, then we are also including processing time on C, which is variable, into C's total convergence time. However, if we take the polling time as C's convergence time, it's possible that the poll was received before C converged, and was processed, and answered, after C converged, skewing the results in the opposite direction.

Unfortunately, there are no simple answers to these problems. Instead, when you are designing a test, or examining the results of a test, the mechanism used to determine convergence, the rate at which that mechanism is used, and the reported final results, should be taken together, and considered closely. A test which reports results in milliseconds, but polls a large number of devices from a single test device, should be examined closely for serialization delay errors.

Use Real-Life Configuration Parameters and Prefix Attributes

Finally, we need to consider what is probably one of the most widely disregarded concerns in testing routing protocol implementations: building accurate and repeatable data sets to feed into the test. Let's examine a common test, to help in understanding this problem.

A network engineer sets up a router connected to a router testing device using a SONET link. The router tester is then configured to feed one million routes, through BGP, to the router being tested. The test is run, and the amount of time it takes for the router to accept and install all of the routes into its local tables is measured. The router is disconnected (we'll call this first router A), and another router (B) is connected. The same test is performed. In the end, the network engineer proclaims A has a better BGP implementation than B, because A accepted and installed the routes fed to it faster than B.

This sort of test, and these results, should raise a lot of red flags for anyone who's ever tested routers before. Many questions here are not answered:

- Were both routers tuned to optimum parameters for this specific test? Most routers are installed in a number of different situations in various networks, and most will perform better if they are tuned to fit the role they are playing in the network. This is similar to tuning a server for database use, or web server use.

- BGP is very sensitive to the data transmitted from one router to another; BGP implementers are generally aware of this, and use differing models of BGP behavior in different networks to tune their implementations. Specifically, in the case of BGP:
  - What percentage of the prefixes transmitted were of specific prefix lengths? What percentage of the routes transmitted were /24s, /23s, and so on?
  - How many different attribute sets were represented in the routing information transmitted? What number of unique attribute sets were included in the routes? For each attribute set, what percentage of the table did that attribute set represent?

Each of these questions can, and should, be compared to real world measures in the network the router is going to be installed in. There are some instances where protocol implementers have tuned their implementation for use in an Internet *Point of Presence* (POP), for instance, and the implementation doesn't fare as well as a route reflector, or the other way around. For some vendors, this tuning could even be on a platform by platform basis, making the job of characterizing a specific implementation through a simple test, like that described above, very difficult.

Conclusion

Designing, executing, and evaluating the results of a test attempting to measure network convergence is much more complex than it appears on the surface. In any given test situation, we need to ask:

- What was the test designed to measure? Is it measuring the appropriate outputs, in the correct ways, to actually measure this?

- What is the granularity of the test results and the actual network events, compared with the measurement techniques used in the test? Will normal test results get lost in the noise introduce by the measurement techniques?
- What is the data set used to build the test? Does it accurately reflect the data the routing protocol implementation will be handling in a real network (or more specifically, the real network the router will be installed in).

When designing, or evaluating, test results, there's a strong tendency to be dogmatic about the results, to say some specific test proves, in some way, a specific vendor, platform, protocol, or implementation, is "better." When evaluating tests in the real world, however, we need to be cautious of such statements, and try to examine the entire environment, considering test results with skepticism, and try to understand their limits—as well as their results.

For Further Reading

[1] V. Manral, R.White, A. Shaikh, "Benchmarking Basic OSPF Single Router Control Plane Convergence," RFC 4061, April 2005.

[2] V. Manral, R. White, A. Shaikh, "OSPF Benchmarking Terminology and Concepts," RFC 4062, April 2005

[3] V. Manral, R. White, A. Shaikh, "Considerations When Using Basic OSPF Convergence Benchmarks," RFC 4063, April 2005.

RUSS WHITE works for Cisco Systems in the Routing Protocols Deployment and Architecture (DNA) team in Research Triangle Park, North Carolina. He has worked in the Cisco Technical Assistance Center (TAC) and Escalation Team in the past, has coauthored several books on routing protocols, including *Advanced IP Network Design, IS–IS for IP Networks,* and *Inside Cisco IOS Software Architecture.* He is currently in the process of publishing a book on BGP deployment, and is the co-chair of the Routing Protocols Security Working Group within the IETF. E-mail: **riw@cisco.com**

Book Review

Running IPv6

Running IPv6, by Iljitsch van Beijnum, ISBN 1-59059-527-0, Apress, 2005. **http://www.apress.com/**

I've read a lot of books about emerging standards that read like "How I spent my summer vacation at a Standards Body." *Running IPv6* is *not* one of those. While van Iljitsch van Beijnum has been an active part of the IPv6 standards community, he has clearly done the homework of making it all work together. Weighing in at a compact 265 pages, *Running IPv6* really gets right to the point. The reader is assumed to have a working knowledge of IPv4.

Organization

The book starts off with a fairly typical introduction that explains why the author believes IPv6 is necessary. I find such introductions tedious, because if you've already forked out US $44.95 for the book, the chances are that you're already motivated enough. This is, however, the only tedious chapter in the book.

What follows is a well written and organized primer for network administrators that covers how to configure end hosts, how get address space allocated, set up tunnels, and configure routers and the *Domain Name System* (DNS). The author covers in detail Linux, Windows, MacOS, Cisco's IOS (as well as that of other routing vendors), and Bind. We next move on to applications, IPv6 internals, transition strategies, and transit services.

Throughout, van Beijnum provides practical tips and advice on some of the pitfalls he found so the reader can avoid them. I particularly liked one case of whether to use eui-64 for the lower 64 bits of the address, pointing out the conflict between reducing configuration information (a good thing) and reduced readability (a bad thing).

The book primarily highlights differences between IPv4 and IPv6. This is important because it helps competent IPv4 administrators build on their existing knowledge. I know the last thing I want read about is how routing works when routing itself hasn't changed between versions. And I enjoyed reading, for instance, how *Dynamic Host Configuration Protocol Version 6* (DHCPv6) and stateless address auto-configuration differ from DHCPv4. I did not need nor want a primer in DHCP, but I did want to know about prefix delegation, which is not present in DHCPv4.

The author wastes no time on fluffy protocol niceties. Who cares, for instance, *how* a flow identifier is selected? What's important is that firewalls of the future may take advantage of it to determine flow direction, a major advance. Packet formats and semantics are only provided as they are needed by engineers to determine whether each component is performing correctly. The book is perhaps, therefore, best commended for what it lacks.

Unfortunately it lacks some subject matter I would like to have seen. Although van Beijnum covers how some common user applications, such as *telnet, ftp,* Web browsers and servers, and media players can use IPv6, business applications folks will be disappointed as there is no discussion of Oracle, SAP, or the like. The same is true for network management applications. And this may be a key roadblock to deployment of IPv6, as no self-respecting IT manager would deploy a service that cannot be managed. Such an obvious absence begs the question of whether those applications are IPv6 capable. On the bright side, you can try just about everything mentioned in the book because just about every tool mentioned either comes with the operating system or is freely available on the Internet. This book is not just theory.

A Must Read

It therefore shouldn't surprise anyone that I consider *Running IPv6* a "must read" for network engineers who have not yet played with IPv6. Even though Network Management Systems and business applications aren't covered, necessary protocol internals, semantics, operations, and troubleshooting are covered, therefore giving the reader a good knowledge base.

—Eliot Lear, Cisco Systems, Inc.
lear@cisco.com

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Letters to the Editor

A Pragmatic Report on IPv4 Address Space Consumption

Ole,

Thanks for a great round up in IPJ Volume 8, No. 3 on IPv6. This really helps focus where the state of the discussion needs to be in terms of addressing IPv6 deployment. You might be interested to know that this edition of the IPJ received tremendous interest in the UK. Within 24 hours of it arriving on your website, it was being distributed widely by several mailing lists serving communities from the *Ministry of Defence* (MoD) to important communications industry membership organisations. I received it myself at least three times from different lists!

Over recent months, I've seen a continuing trend to try to sideline IPv6 as not relevant to a particular discussion. IPv6 is either too low level for applications providers to think about, or too far off, or doesn't support some essential infrastructure service today. Some communities feel they have more than adequate IPv4 addresses to meet their foreseeable needs. These factors continue to drive debate on "if ever" rather than on "when" and "how" to deploy. That is, if the debate happens at all. All those who are investing in future IP-related services and networks need to read this edition of the *Internet Protocol Journal* for a reality check.

Tony Hain's article provides a compelling addition to the work you've already published by Geoff Huston on the analysis of IP address allocation, and is important food for thought that I think justifies increasing the urgency with which IPv6 support is treated. The discussion you hosted between Tony, Geoff with John Klensin and Fred Baker I think dealt very clearly with why the debate needs to be focused on the *how* and the *when* rather than on the *if*.

In the UK, we are seeing some significant investments made to enable IP level infrastructure with the intent of delivering profoundly new services into the twenty-first century, but none of these major investments appears to have included a vision for IPv6. So I think the point that was made concerning the current failure in making like-for-like investment decisions between v4 and v6 is hugely important for Chief Information Officers and Chief Financial Officers to take to their boards, or we will continue to find people investing for the past, rather than as they apparently believe, their future.

—*Christian de Larrinaga*
cdel@firsthand.net

Ole,

The analysis undertaken by Tony Hain and debated by some recognised experts makes it abundantly clear that the deployment of IPv6 is an immediate natural growth path to sustainability and global mass-market penetration of the Internet, beyond its worldwide current rate of less than 15%.

Tony has presented his study in the recent *IPv6 Forum Summits* (Seoul, Taipei, San Jose and Canberra) and obviously took a lot of people by surprise as previous studies maintained the suspense that the deployment of IPv6 should be an incremental transition and not an imminent and real migration. It was therefore decided to responsibly and morally act on this and renew a global Call to Action to set 2008 as a milestone of inevitable smooth transition in a softer form as a Y2K or Yv4 (The Year when IPv4 addresses will become hard to get) and get engineers to plan for it.

A global worldwide press release was published October 11, 2005 and can be read on the web site of the IPv6 Forum:
http://www.ipv6forum.org

The IPv6 Forum would like to recognise the work of *The Internet Protocol Journal* in watching diligently this space for the past couple of years and for initiating and orchestrating the constructive and consensual debate included at the end of the study, a contribution we trust is of great significance to the global good of the Internet.

—*Latif Ladid, IPv6 Forum President*
latif.ladid@village.uunet.lu

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

March 2006 Volume 9, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Autonomous Systems Numbers (ASNs) play an important role in the routing architecture of the Internet. An *Autonomous System* (AS) is, according to RFC 4271, "… a set of routers under a single technical administration, using an *interior gateway protocol* (IGP) and common metrics to determine how to route packets within the AS, and using an inter-AS routing protocol to determine how to route packets to other ASs." AS numbers are—like IP addresses—a finite resource, and predictions exist for when the AS number pool will be depleted. In our first article, Geoff Huston explains how ASNs work, and introduces us to the 4-byte ASN scheme that will allow for future growth beyond the currently predicted depletion date.

Our second article looks at another aspect of Internet routing and addressing—the IPv4 number space itself. Designers and operators of internets are often required to perform various address calculations in order to properly configure their networks. Russ White takes us through several exercises and introduces some "tricks of the trade" to make such calculations easier.

Our articles on spam in the last issue of IPJ prompted some feedback from our readers, and promises of more articles from other authors. This problem space clearly has more than a single solution. We look forward to bringing you more coverage of this topic in future editions.

The second issue of the *IETF Journal*, published by the Internet Society, is now available. Some people have asked me if I think of this new journal as a "competitor" to IPJ. I am happy to say that the *IETF Journal* is very much complementary to IPJ and covers important news from the IETF that we hope our readers will find interesting. You can access the *IETF Journal* by visiting: **http://ietfjournal.isoc.org**

The *IPJ Reader Survey* will soon close. We are grateful to the many readers who took the time to tell us about their reading habits, ideas for future articles, and other suggestions. Of course, we always welcome your feedback on any aspect of IPJ. Just drop us a line via e-mail to: **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Exploring Autonomous System Numbers

by Geoff Huston, APNIC

So what are *Autonomous System Numbers* (ASNs), and what role do they play in the technology of the Internet? This article explores the role of ASNs as a critical element of the Internet routing architecture. We will first explore how the AS number space is structured, examine how ASNs are used in the interdomain routing environment and then look at the consumption rate of these numbers, and finally examine our options when we get to the point of likely ASN pool exhaustion. However, in order to put this into context, a brief overview of Internet routing architecture follows.

Internet Routing Architecture

Internet routing architecture is structured as a two-level hierarchy. The environment is first partitioned into *domains* with each domain using an internal routing environment. These network domains use an interior routing protocol (commonly referred to as an *Interior Gateway Protocol* [IGP]), which maintains a complete mapping set for the current internal topology of the domain, together with the set of "best paths" between any two points within the network domain. Although this approach of having a routing protocol automatically maintaining a comprehensive view of the current topology can be made to work within even quite large routing domains, such an approach does not scale to the size of the entire Internet. Fine-grained topology information is useful only in "local" situations, and is best omitted when forming a larger view of the network. Commonly used interior routing protocols include *Open Shortest Path First* (OSPF), *Intermediate System-to-Intermediate System* (IS–IS), and *Enhanced Interior Gateway Routing Protocol* (EIGRP).

The second level in the routing hierarchy is the *interdomain* routing domain. The interdomain routing environment describes how domains interconnect, but avoids the task of maintaining transit paths within each domain. In the interdomain space, a routing path to an address is described as a sequence of domains that must be transited to reach the domain that originates that particular address prefix. Today this interdomain space is maintained using Version 4 of the *Border Gateway Protocol* (BGPv4).

Each routing domain is a single administrative domain, operated within a uniform set of routing policies, and is operated independently from any other domain. The domain is in effect an autonomous unit in the overall routing architecture, and is termed an *Autonomous System* (AS). Each of these ASs is uniquely identified using an *Autonomous System Number* (ASN).

What Is an Autonomous System?

One of the best definitions of an Autonomous System can be found in an IETF document, RFC 4271[4] that describes BGPv4:

> "The classic definition of an Autonomous System is a set of routers under a single technical administration, using an *interior gateway protocol* (IGP) and common metrics to determine how to route packets within the AS, and using an inter-AS routing protocol to determine how to route packets to other ASs. Since this classic definition was developed, it has become common for a single AS to use several IGPs and sometimes several sets of metrics within an AS. The use of the term Autonomous System here stresses the fact that, even when multiple IGPs and metrics are used, the administration of an AS appears to other ASs to have a single coherent interior routing plan and presents a consistent picture of what destinations are reachable through it."

The AS Number Pool

ASNs are drawn from a 16-bit number field, allowing for 65,536 possible values.

AS 0 is reserved, and may be used to identify nonrouted networks. The largest value—AS 65,535—is also reserved. The block of ASNs from 64,512 through 65,534 is designated for private use. ASN 23,456 is reserved for use in ASN pool transition. The remainder of the values, from 1 through to 64,511 (less 23,456), are available for use in Internet routing. The number space is unstructured, because there are no internal fields in the number structure, nor is there any aggregation or summarization capability for ASNs.

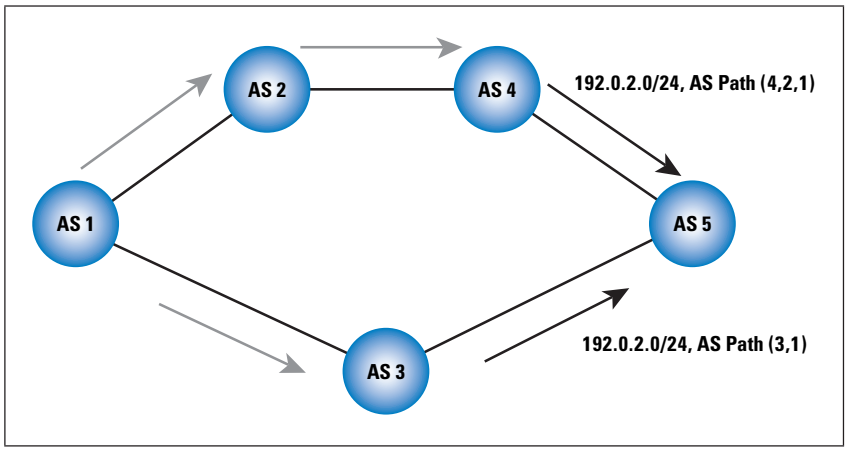
How AS Numbers Are Used in BGP

The interdomain routing space is constructed using two components: *address prefixes* and *AS numbers,* which are used as domain identifiers. Every prefix has an originating domain, known as the *Origin AS* from which reachability for the prefix is propagated across the interdomain space.

As the routing advertisement is propagated across the interdomain space, each prefix accumulates an associated "AS path." When an address prefix advertisement transits a domain, the domain effectively "signs" the prefix advertisement by prepending its ASN to the AS path associated with the address prefix. At any point in the network the AS path describes a sequence of connected domains that forms a path from the current point to the originating domain. This setup is shown in Figure 1, where AS1 originates an advertisement for the address prefix **`192.0.2.0/24`**. At AS5, the AS receives two BGP advertisements for this prefix. One has the AS path (4, 2, 1), and the other has the AS path (3, 1).

The left-most number in the AS path list is the ASN of the adjacent AS from which the address prefix advertisement was received. The sequence of numbers indicates the sequence of ASs though which this update was propagated. The right-most, or final ASN, is the AS number of the AS that originated the address prefix advertisement, or *Origin AS*.

Figure 1: AS Path Generation in BGP



The AS path serves two purposes in interdomain routing: that of a *path length* metric and a *loop detection* mechanism.

The AS path is used as a path metric in the BGP path selection algorithm. When a domain receives two different BGP advertisements for the same address prefix, the default BGP selection process is that of selection of the advertisement of the minimal-length AS path, with each AS in the path counting as a single unit of "cost." In the case of the example network in Figure 1, AS5 prefers to use the path through AS3 to reach the originating AS1, in preference to the longer path of AS4 and then AS2.

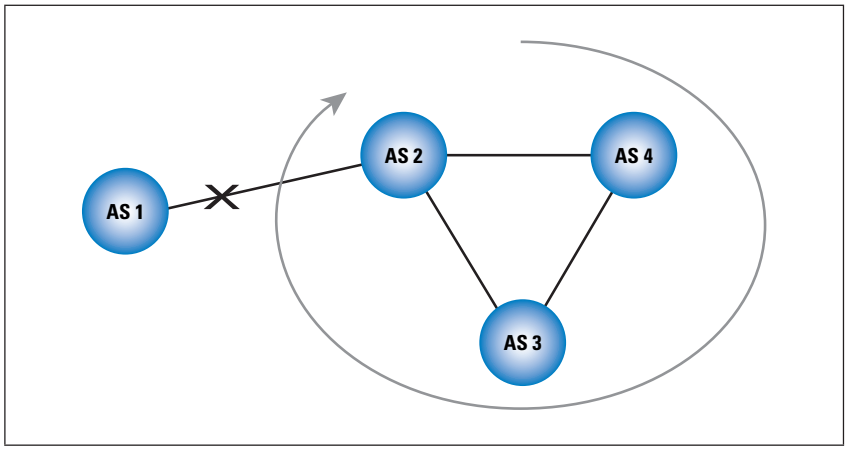
Although enumerating the AS path vector within the routing protocol is one way of passing the path cost through the routing domain, it may appear that the best path selection function could just as easily be supported by carrying a simple path cost metric of a domain transit counter, similar to that used by other distance vector routing protocols, such as *Routing Information Protocol Version 2* (RIPv2). However, the problem with distance vector protocols is the "count-to-infinity" dilemma.

To illustrate the need for explicit AS path enumeration in BGP, consider what happens when the AS path vector is replaced by a simple path cost metric. In the configuration shown in Figure 2, AS1 originates a routing advertisement toward AS2. AS2, AS3, and AS4 are interconnected in a simple loop configuration. When AS2 receives AS1's advertisement with a path cost of 1, it passes the advertisement on to both AS3 and AS4, with a path cost of 2. Both AS3 and AS4 select as their best path this advertisement from AS2 with a path metric 2, corresponding to the AS path (2, 1).

Now if the connection between AS1 and AS2 is broken, then AS2 no longer sees AS1, and withdraws its best path to the prefix through AS1. AS2 then stops advertising a path to AS3 and AS4. But AS3 is already advertising a path to AS4, with a metric of 3, corresponding to the AS path (3, 2, 1). Upon the withdrawal of the advertisement from AS2, AS4 then selects this as its next best path, with a path cost of 3. AS4 then advertises this prefix to AS2 with a path cost of 4, corresponding to the AS path (4, 3, 2, 1).

At this point, without the explicit AS path in the advertisement, AS2 cannot deduce that this advertisement is, in fact, a loop. Accordingly, AS2 accepts this path with a metric of 4 as its best path. AS2 then advertises this to AS3 with a metric of 5, corresponding to the AS path (2, 4, 3, 2, 1). AS3 updates its best path to AS1 with this new metric and then sends an update to AS4, and so on. This process continues around the loop until the path cost metric reaches some defined maximal value. The higher the maximal value for the path cost metric, the longer the time taken to detect the loop condition. The smaller the maximal path cost metric, the smaller the span of network that the protocol can encompass. Setting the maximal path cost parameter requires some considerable care, and the operation of the protocol can be extremely slow to converge in terms of loop detection.

Figure 2: Loop Formation in Distance Vector Protocols



This form of loop can be averted by replacing the path cost counter with a fully enumerated AS sequence. Continuing the example in Figure 2, when AS2 withdraws its route to AS3 and AS4, AS4 still selects the other route it has heard, but this time the selected prefix has the path (3, 2, 1). When AS4 attempts to pass this advertisement to AS2, AS2 sees its own value in the associated AS path and rejects the advertisement. At the same time AS3 withdraws its advertisement to AS4, and at that point the prefix is dropped from the entire routing system. In this way the AS path acts as an efficient routing loop detector.

The use of ASNs and AS path vectors in BGP provides an effective solution to this classic problem of loop detection, as well as providing a simple and effective path-selection process.

Who Needs an AS Number?

Not every network needs to have its own ASN. The guiding principle is that ASNs are used to express distinct interdomain routing policies, and not every network has the requirement to express its own unique set of routing policies.

In the case where a network has a single upstream connection, the routing policies of the network are precisely the same as those of its upstream service provider, and there would normally be no need for the network to use a distinct ASN. Even if the network domain uses BGP for its upstream connection, the originating domain can use a private ASN (from the number range 64,512 – 65,534) to support the BGP session to the upstream network. The upstream network strips off the private ASN when it readvertises the prefix, and the upstream network appears to the rest of the Internet as the originating AS. Even if the AS has "downstream" networks it can still use a private AS, even when the downstream ASs are using public ASNs. The stripping of the private AS removes only the instances of the private AS from the AS path, and not the public ASNs (Figure 3).

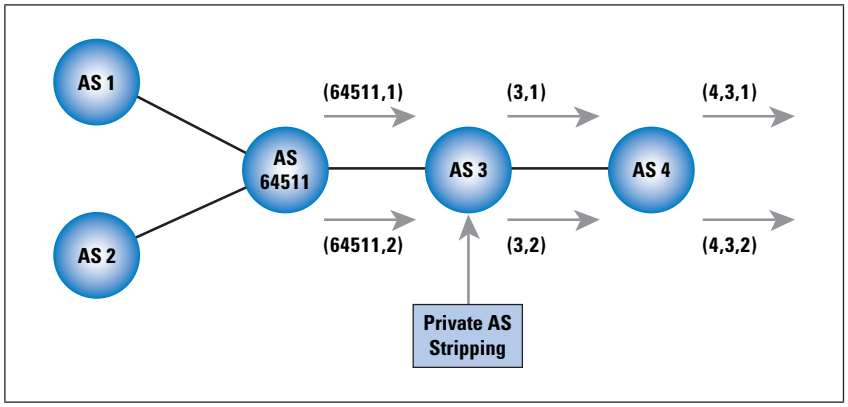


Figure 3: Use of Private AS Numbers

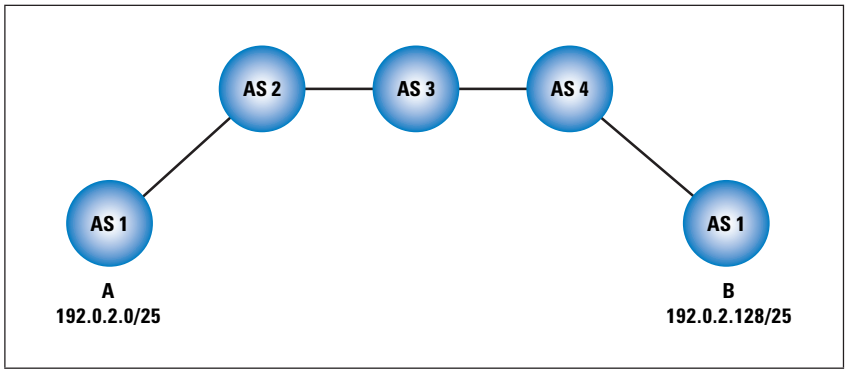
In the case where a network has two or more upstream transit connections, it is more likely that the network will use its own unique ASN. It is not always the case that a distinct ASN is required here, and the distinguishing factor is that of the network wanting to express particular routing policies. Where the network has no particular preference as to which of the upstream services should be used for incoming traffic, the network can also use a private ASN for each of its routing sessions. In such a case the external routing view would be that the prefix appears to be originated from multiple ASs.

In the case where there are multiple paths to reach the network, and where these paths need to be distinguished in the routing system by different AS paths that have the same originating AS (that is, there is a need to express a routing policy), then the network needs to use a unique ASN within the interdomain routing system.

Can an ASN Be Split Across Separated Subdomains?

There are many cases of dispersed networks that exist in multiple locations. If these locations are all administered by a single entity, it may be desirable to use a single ASN across all these domains. This scenario is possible, but considerable care needs to be exercised when designing the routing configuration. Figure 4 shows two distinct subdomains of AS1, and they are not interconnected internally.

Figure 4: Split AS



AS1 (A) advertises the prefix **`192.0.2.0/25`** to AS2, and this advertisement is propagated to AS2, AS3, and AS4. When AS4 passes this advertisement to the other segment of AS1 (B), this router rejects the advertisement because the associated AS path (4, 3, 2, 1) indicates that the route has already passed through AS1. Similarly, the first segment of AS1 (A) rejects the advertisement of **`192.0.2.128/25`** from AS2, because its path (4, 3, 2, 1) also indicates that a loop has formed. To restore complete connectivity between the distinct parts of AS1, AS1 needs to configure static routes at its edges. If AS1 (A) configures a static route to **`192.0.2.128/25`** pointing toward AS2, and AS1 (B) similarly configures a route to **`192.0.2.0/25`** through AS4, then the configuration enables full connectivity.

In more complex configurations where each of the segments of the network is multiply connected, the static route configuration becomes more complex. However, with very careful configuration, a single ASN can be distributed across multiple distinct networks.

AS Path Prepending and Path Poisoning

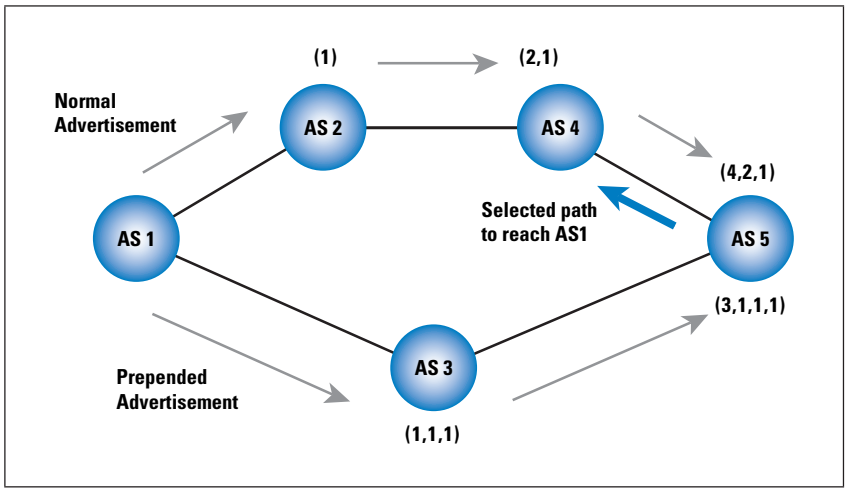
The basic mechanism of path preference in BGP is that of the AS *path length*. Where there are two advertised paths to reach a particular address prefix, the default selection algorithm in BGP is to prefer the advertisement with the *shorter* AS path length.

A multihomed domain may wish to have other domains prefer one particular path over another to reach it. This may be because the local domain wishes to optimize its traffic costs between the multiple upstream providers, balance the traffic load across multiple paths, or set up various forms of primary and backup relationships across the multiple provider upstream paths.

Although such policy preferences are often set up using BGP *communities*, BGP community signaling requires the cooperation of multiple parties in consistent interpretation of the community values. A more coarse form of expressing such policy preferences can be achieved through AS *path prepending*, a technique of deliberately extending the AS path length of a prefix advertisement by adding additional ASNs into the AS path of an advertised prefix. Normally the form of AS path prepending uses the local ASN to perform the prepending.

In the example in Figure 5, AS1 wants to express the policy to prefer incoming traffic through AS2, and use the link to AS3 only as a backup. To achieve this with AS path prepending, AS1 prepends itself twice in the AP path of the advertisement passed to AS3, in order to artificially lengthen the AS3 transit path. AS5 would have normally used the shorted AS path through AS3 to reach AS1. As a result of AS1 artificially lengthening its path to AS3, AS5 now selects the transit path through AS4 and AS2 to reach AS1.

Figure 5: AS Path Prepending



Of course AS path prepending is a very imprecise technique, and can often produce surprising results in real-world situations. A more deterministic method of traffic engineering uses additional signals attached to address prefix advertisements, through BGP communities.

A more subtle, and more controversial, prepending technique is that of so-called AS *path poisoning*, where an AS uses some other value to prepend in the AS path. In Figure 6, AS1 wants to express the policy that under no circumstances should AS5 use the transit through AS3 to reach AS1. In this case AS1 could use AS5 as the prepending value in its advertisement to AS3.

When AS5 receives this advertisement, the presence of its own ASN in the AS path means that it will not accept this advertisement, and prefers the transit path through AS4 and AS2. The difference between these two examples is that in the case where the connection between AS1 and AS2 is broken, none of AS2, AS4, or AS5 can possibly reach AS1 when this AS path poisoning technique is being used.

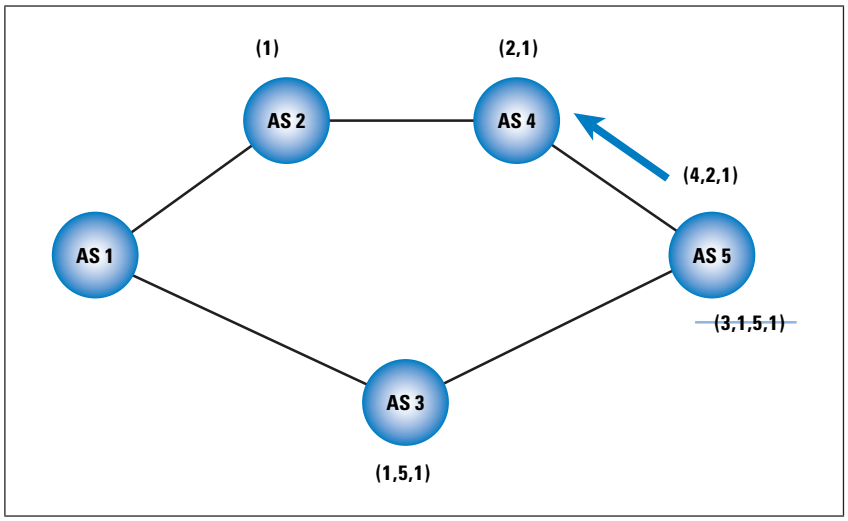


Figure 6: AS Path Prepending with AS Path Poisoning

AS Number Consumption

In this section we will look at the rate of consumption of ASNs, and estimate when they may be fully consumed. Of the 64,510 available AS numbers, as of January 2006 we have already allocated some 40,000, or well over half of the number pool. Two immediate questions arise—how long do we have before the number pool is completely exhausted, and what are our options for an expanded number pool that can encompass a larger interdomain routing environment?

The Factors for AS Number Consumption

Before looking at these two questions in further detail, it would be useful to understand the factors that affect AS number consumption.

From one perspective it is counterintuitive to assume that the Internet will evolve from tens of thousands of distinct routing domains to one of hundreds of thousands or even millions of distinct routing domains. It may appear that there is a reasonable level of correlation between the number of active *Internet Service Providers* (ISPs) in the Internet and the number of advertised ASNs. If forecasting a future demand for hundreds of thousands or even millions of ASNs, it would appear that we are forecasting continued fragmentation of the service provider industry with large numbers of small enterprises that, collectively, compose the Internet. This scenario does not appear to be likely.

The ISP industry is one with an underlying factor of economies of scale. Larger ISPs generally have access to more efficient use of resources and are more capable of sustaining a market share at competitive prices, with reasonable operating margins because of these economies of scale. Smaller providers tend to service niche markets, and in general are highly susceptible to pricing pressures in the competitive supply market. The overall result is strong pressure for continued aggregation in the service provider market, tending to aggregate to a smaller number of larger providers.

If the number of ASs in use is roughly commensurate to the number of service providers, then this view of the market dynamics would lead to a view that the service provider population is either in a state of equilibrium where the entrance of new niche-oriented players is much the same as the rate at which smaller players are aggregated into larger providers, or one of relatively small growth based on the larger dynamics of continued expansion of the Internet on a global basis.

In practice this has not been the case, and we see a continuous rate of consumption of new ASNs. This rate appears to be some 3,500 ASNs per year, and this consumption rate appears to have been steady since 2002 (see Figure 7). Accordingly, it appears that some additional factors affect AS number consumption rates.

One of these factors is the practice of *multihoming* at the edge of the network. Many end-site networks have business-critical needs for assured Internet connectivity, and a common way to achieve this connectivity is by using the services of two or more upstream providers. In such situations the end site may want to express different routing policies to each upstream provider, and it does so by using its own ASN and expressing these routing policies using BGP to each of its upstreams.

AS numbers are also used in other contexts. In *Multiprotocol Label Switching* (MPLS) Layer 3 networks, one form of generating the *Route Distinguisher* value for a VPN client network is through the use of concatenating the VPN host's AS number with a serial number. To what extent this semiprivate use of AS numbers in a VPN context contributes to the consumption rate of ASNs is difficult to assess, simply because the use of these numbers is not generally visible.

Even within the public Internet there are other contributory factors to AS number consumption. ISPs with diverse product portfolios may wish to express different routing policies for various product families, or express different routing policies in different regions of network coverage. Again this can be achieved through the use of distinct AS numbers of each routing policy set.

An associated contributory factor for AS Number consumption is that there is little incentive for AS Number return and recycling. With the current framework there is no direct cost to maintain an AS number allocation, and the overall characteristic of AS number allocation appears to be a "once and forever" allocation model. When AS numbers are no longer required, AS numbers generally do not return to the unallocated pool for subsequent reallocation.

Taken together, these factors lead to the conclusion that continued AS number consumption is based on a larger set of considerations than the dynamics of the service provider industry.

Accordingly, we can be a little more confident in making the assumption that the factors that have affected AS number consumption in the recent past will continue to be factors in the near-term future, leading to some further confidence in a predictive technique that uses recent consumption data to generate trends that can made predictive forecasts of future demands. We will apply this technique to AS number consumption data to make some forecasts of the time by which the current AS number pool will be exhausted.

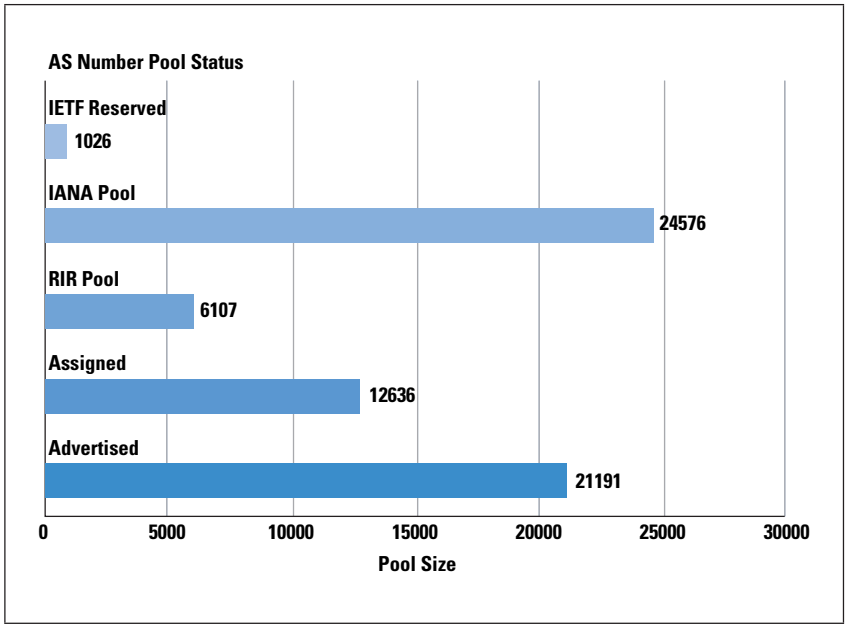
AS Number Pool Status

There are 65,536 AS numbers. As noted already, some 1,026 numbers are reserved and unable to be used in the public Internet, leaving 64,510 for use in the public Internet.

The pool of AS numbers is administered by the *Internet Assigned Numbers Authority* (IANA), and blocks of 1,024 numbers are allocated to the *Regional Internet Registries* (RIRs) periodically when the RIR's pool drops below a threshold level.

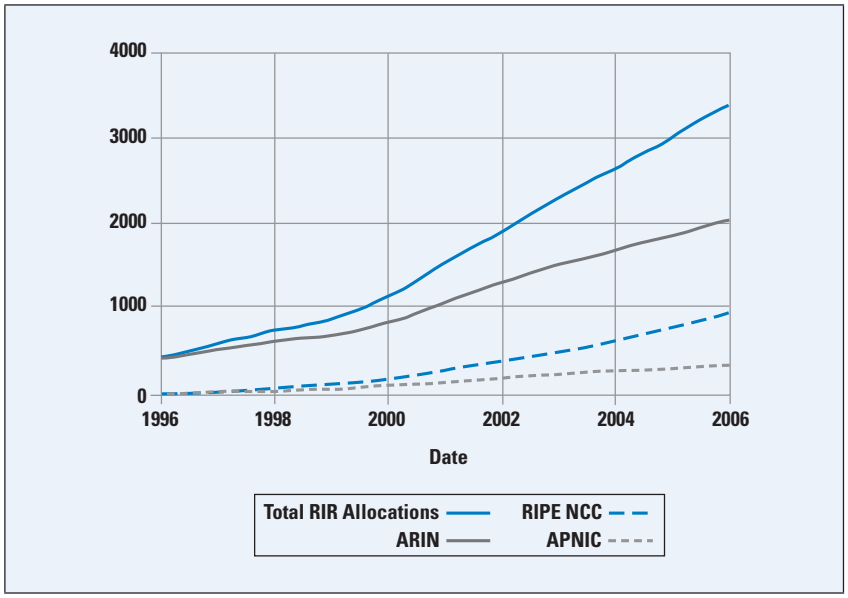
Of the 39,934 AS numbers that have been allocated by IANA by January 2006, there is a further classification of AS numbers. A working pool of numbers is held by the RIR for current assignment to ISPs. Of the assigned AS numbers, some are visibly used in the interdomain routing table of the public Internet, but others are not visible in the Internet. The breakdown of AS numbers into the RIR pool, assigned but not advertised, and assigned and advertised, as of January 2006, is shown in Figure 7. Of the 34,827 assigned AS numbers, some 21,191 are advertised; 12,636 have been allocated in the past, but are not currently advertised in the BGP routing table.

Figure 7: AS Number Status of Advertised, Unadvertised, and Unallocated Pools



The RIRs allocate ASNs to ISPs and end-user networks. A second time series can be generated, showing the cumulative sum of the RIR AS allocations (Figure 8). Not surprisingly, the time series shows the effects of the Internet boom across the period from 1999 through to late 2001 as a sharp upward trend in allocations. The subsequent market correction is also evident as a visible change in the AS allocation rate by early 2002.

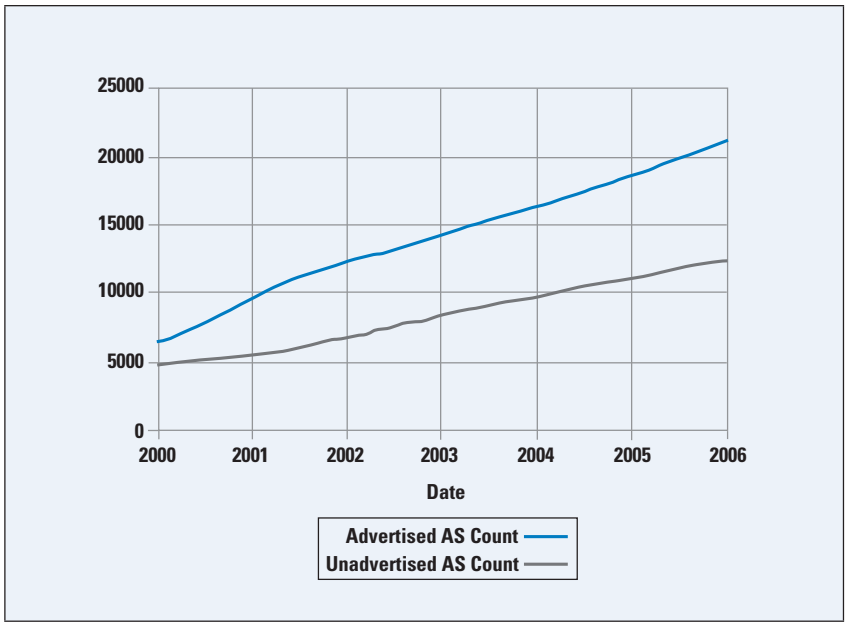
Figure 8: RIR Allocations



BGP AS Advertisements

In addition to allocation rates, a further source of ASN data is the interdomain routing table. The number of distinct ASs advertised in the interdomain routing space of the public Internet has been measured regularly since 1997. The time series of this count of advertised ASNs, and the complementary number of unadvertised ASNs, is shown in Figure 9.

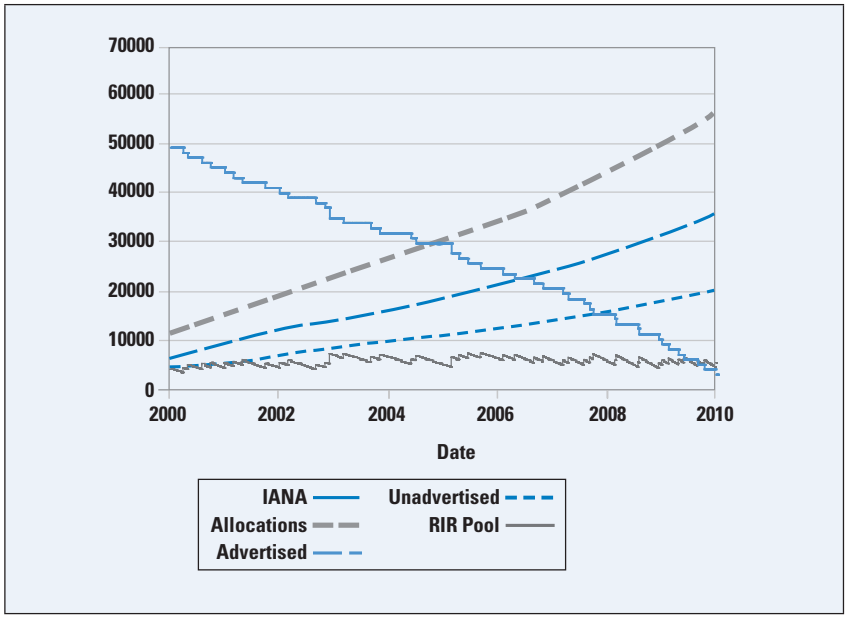
Figure 9: Advertised and Unadvertised AS Numbers



AS Number Consumption Projections

At this point, it is possible to make some projections on AS number consumption. The technique here is to use the past three years' consumption data (taking a starting point of January 2003) and derive an associated exponential function as a best fit to the 3-year data series in order to generate a trend function. This trend function is then projected forward in time to forecast the point in time when the resource reaches a certain threshold point. A considerable amount of detail is associated with this exercise, including the use of an exponential function as the best fit to the past 3 years' ASNs use rates (see **`http://www.potaroo.net/tools/asns/`**). However, for the purposes of this article it is appropriate to proceed to the outcome (Figure 10).

Figure 10: A Predictive Model of AS Number Consumption



From this model it appears that we are looking at steadily accelerating consumption of ASNs, and a projected date of late 2010 of exhaustion available AS numbers to allocate to ISPs.

The implication is that this model indicates that by late 2010 either the Internet should be using a new protocol for interdomain routing that does not rely on AS numbers at all, or, more likely, that the Internet should be using a version of BGP that supports the use of larger AS numbers that are drawn from a number pool significantly larger than 16 bits. The first option appears to be somewhat unrealistic, to say the least. And the second option, although simpler and very much the preferred path, is still going to take some time to deploy, particularly considering the growing size of the interdomain space of the Internet and the diversity of these component domains.

When contemplating a transition to a larger ASN pool, it should be remembered that every day there are more networks that will need to undertake a transition to a longer ASN field in their deployed instances of the BGP protocol.

The steps in this transition path appear to include:

- The completion of the relevant protocol standards for a larger ASN field in BGP
- The production of code in available implementations of BGP that support this protocol standard
- Various forms of testing this code, both in terms of its correct operation and interoperability and in terms of the correctness and viability of the relevant transition steps

- Developing the necessary infrastructural support system to manage the distribution of this new number pool
- A process of deployment of this protocol so that the deployment of larger ASNs can commence well before the point at which the existing AS number pool is exhausted

Even an aggressive schedule of transition across such a large and diverse network as the Internet will take many years to reach the final step. It also appears that a prudent course of action would see us reach that position not by 2010, but by 2008 at the latest, allowing us a margin of some 2 years (and some 10,000 remaining AS numbers) to complete the task.

32-Bit AS Numbers

In this part of the article we will look at the current proposal for a larger AS number pool. As of October 2005, the document defining this proposal is an IETF Internet Draft: **`draft-ietf-idr-as4bytes-12.txt`**. The proposed approach is to expand the size of the AS number pool space from 16 to 32 bits. In number terms this expands the number space from a pool of 65,536 numbers to 4,294,967,296 numbers. In terms of the current use of ASNs, the current scaling properties of the BGP routing protocol, and the use of ASs in the context of interdomain routing, a pool of some 4.3 billion numbers would easily encompass a network environment of significantly greater levels of domains, and interdomain interconnection density. Such a pool size would exceed some current guesses of the scaling capabilities of the BGP protocol by up to a further two orders of magnitude.

It is also proposed to preserve the first block of 65,536 32-bit ASNs to align with the allocations of the 16-bit numbers.

> Let's use a new form of terminology here for 32-bit ASN values, where the first 65,536 ASNs are numbers that use the form "**`0.0`**" through "**`0.65535`**." The second set of 65,536 numbers would be written as **`1.0`** through to **`1.65535`**, and so on. So here we will be using a number format of <*upper16 bits*>.<*lower 16 bits*>.

What is the inventory of concerns that need to specifically addressed in the transition to these 32-bit AS Numbers?

Obviously there is a need for some changes to the routing protocol, and an ordered interdomain transition is unrealistic to expect. More reasonable is an expectation of a piecemeal transition of domains, where individual domains transition their BGP platform to supporting 32-bit ASs in their own time. Domains that are currently using 16-bit ASs may have less reason to undergo an early transition to 32-bit AS support, whereas those domains that are assigned a nonmappable 32-bit ASN will find that they have to support 32-bit ASNs from the outset.

32-Bit Changes to BGP

BGP has two major parts within its protocol: opening up a BGP conversation with a peer BGP speaker, and then the transfrerring of protocol objects that describe reachability of address prefixes and associated attributes of these address prefixes. Both parts include AS Number components, and in considering changes to the current protocol, both parts of the protocol require some change. The message objects that need to be considered here are, therefore, the BGP OPEN message and the BGP UPDATE message.

The changes to BGP create a "NEW" BGP implementation that is capable of supporting a 32-bit ASN environment. The essential task of the changes is to define mechanisms that all NEW BGP speakers use to speak to each other and pass all ASN values in 32-bit fields. However, the Internet is way too large to set up a "flag day" at which point the entire collection of BGP speakers will undertake a switch from "OLD" BGP to NEW BGP. Accordingly, it is also necessary to define protocol interactions in NEW BGP where the transition in the Internet will be gradual and essentially uncoordinated. NEW BGP speakers will have to set up sessions with OLD BGP speakers, and of course OLD BGP speakers will also be peering with other OLD BGP speakers. The information associated with 32-bit AS paths must be passed across sections of the network that normally support only 16-bit AS paths. In other words, 32-bit AS information needs to be passed to OLD BGP speakers and between OLD BGP speakers.

The general approach adopted for transition is preserve AS path length information across the OLD and NEW BGP boundaries, while recognizing that some 32-bit AS information cannot be cleanly mapped into a 16-bit AS path. In order to preserve 32-bit information—a necessary step to prevent loop formation for 32-bit ASs—the 32-bit information is preserved across OLD transit paths and restored upon reentry into NEW BGP realms (Figure 11).

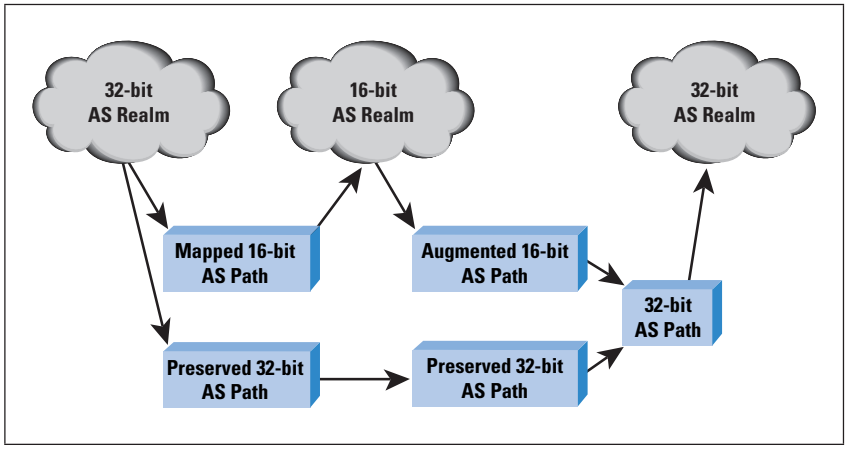


Figure 11: 16-Bit and 32-Bit AS Realms

Opening a BGP Session

The proposed approach is to initiate a NEW BGP session in a mode that is compatible with the OLD BGP protocol, and also inform the remote peer of its capability to conduct a NEW BGP conversation if the remote peer is also a NEW BGP speaker. NEW BGP speakers who open a peer session with an OLD BGP peer will ignore the NEW capability and operate their BGP peer session in OLD mode. A NEW BGP peer will respond positively to the NEW capability, and that BGP session can then operate in NEW mode.

The BGP OPEN message includes a fixed-length 16-bit *My_AS* field as well as potentially containing a capability query as part of the *Optional Parameters* section. In order to ensure that NEW and OLD speakers can communicate, this 16-bit *My_AS* field needs to be preserved in NEW BGP even when the Optional Parameters section includes the capability to undertake a NEW peering session. This may appear contradictory in the first instance, because the OPEN message then contains both a 16-bit ASN and a 32-bit AS *Capabilities Query*. The mechanism proposed for the OPEN message varies according to whether the NEW speaker is using a mappable ASN drawn from the original pool (that is, with a *My_AS* number in the range 0.0 through 0.65535), or it is using a number drawn from a higher-numbered 32-bit number block. In the first case the OPEN message would use the 16-bit mapped value in the *My_AS* field (dropping out the zero-valued high-order 16 bits of the AS value), whereas in the second case the BGP speaker would use for *My_AS* a special 16-bit value that is reserved for this purpose (AS 23456). In both cases the Optional Parameter section would include a capability code to indicate that the local BGP speaker can support 32-bit ASNs (Capability Code 65).

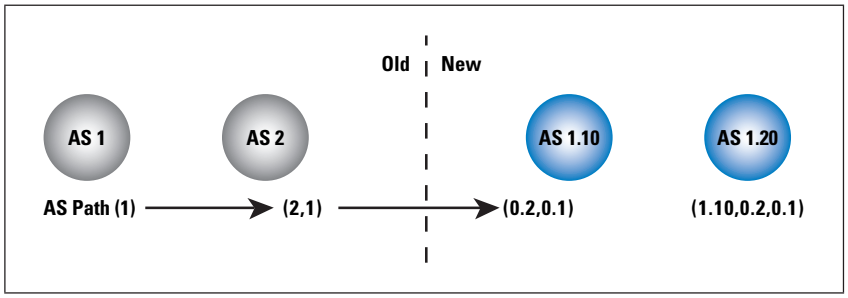
The side effect is that from the perspective of OLD BGP domains AS 23456 may appear to be connected to the interdomain network in many different locations. From the OLD BGP realm this does not present a protocol problem, although, as always, there is the potential here that this repeated use of AS 23456 as a 32-bit AS substitution token may create a somewhat confusing BGP view of the Internet from the perspective of the OLD BGP world.

The capability exchange uses a protocol described in RFC 3392. The NEW BGP speaker adds an optional capability field to the OPEN message. The 32-bit AS capability code 65 carries as its capability value the local 32-bit local ASN value. For a NEW peer this capability value is to be interpreted as the actual AS of the remote side, on the basis that the *My_AS* field in the body of the OPEN is either a truncation of the local 32-bit AS value (in the case of mappable 32-bit AS values), or the special value of AS 23456.

The BGP UPDATE Message

For a NEW BGP session (32-bit peering with 32 bits) the changes to the protocol are the use of 32-bit ASNs in the AS_PATH attribute of UPDATE messages. All 16-bit AS values are padded with a zero high-order 16 bits. If the AGGREGATOR attribute is used, it is similarly carried as a 32-bit value. So in the 32-bit peering, all 16-bit information is carried in mapped 32-bit ASNs (Figure 12).

Figure 12: OLD to NEW BGP AS Path Mapping



In this way AS path length is preserved without change when translating 16-bit AS information into the 32-bit domain.

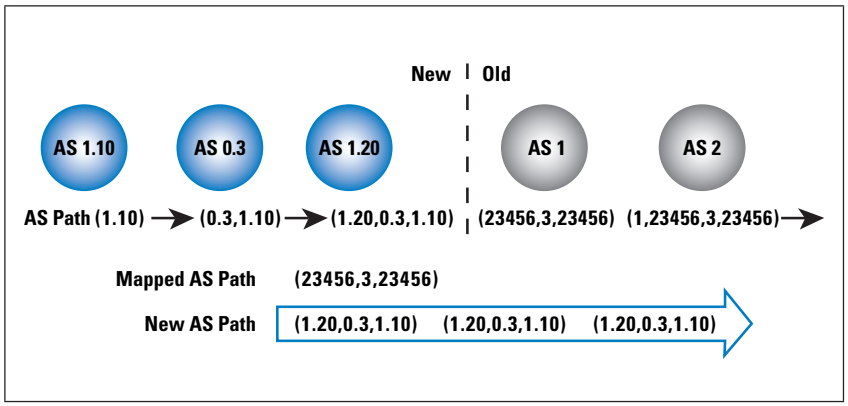
The next case is where an OLD BGP peers with a NEW BGP. We have already seen the simple case where the information is coming from a 16-bit path and there is no additional 32-bit information, and in this case the 16-bit values are simply mapped into 32-bit values, by padding the ASN values with 16 zero high-order bits. What about the reverse case where 32-bit information is being passed back into the 16-bit world?

This case has two parts: first creating an equivalent 16-bit AS path and second, packing up the 32-bit AS path information in such a way that it transits across the 16-bit domain in such a manner that that it can be reassembled in any subsequent transition into a 32-bit domain. In the first case, the equivalent path information is constructed by stripping the high-order 16 bits off the AS value, as long as this part is all zeros. Where this is not possible—and the AS path contains one or more ASNs with non-zero high-order bits—then the transition ASN, 23456, is substituted in the place of each such ASN in the AS path. In this way the AS path length metric is preserved, and the prevention of count-to-infinity loops in the 16-bit domain is avoided.

The second part to this case is packaging up the 32-bit path into the OLD BGP session in such a way that it can be unpacked at any subsequent boundary back into a 32-bit routing realm. Here the proposal calls for new transitive community attributes to be carried in OLD BGP routing realms. These attributes are defined as transitive attributes, and should be passed through the OLD BGP peering sessions without alteration. It should be noted that this is not a protocol change as such, but it does require the explicit configuration support within OLD BGP implementations of this attribute as a transitive community.

The proposed mechanism is an extended community attribute called "NEW_AS_PATH." When a NEW BGP speaker is speaking to an OLD BGP, the NEW BGP prepends its own AS value to the AS_PATH and copies this information into the NEW_AS_PATH attribute. It then translates the 32-bit AS path into a 16-bit equivalent AS path. The translation is straightforward, in that where the 32-bit AS has all zeros in the high-order 16 bits, the translation truncates the AS value to a 16-bit value, and where the high-order 16 bits are nonzero, the translation substitutes the reserved 16-bit value AS 23456 in its place (Figure 13).

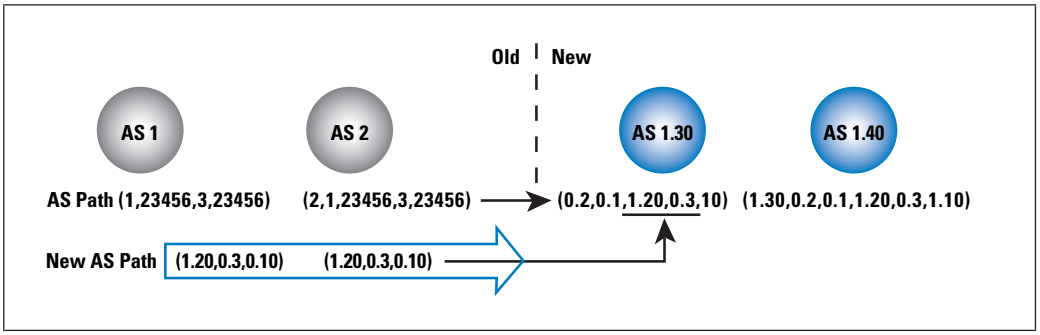
Figure 13: NEW to OLD BGP AS Path Mapping



The transit across the OLD BGP domains leaves the NEW_AS_PATH untouched, and prepends 16-bit AS values to the AS_PATH. In other words, OLD BGP behaves as it always has. The NEW_AS_PATH is passed through the OLD realm as an opaque bit block.

The next transition is one from the OLD to the NEW domain when a NEW_AS_PATH attribute is present. In this case the NEW BGP speaker takes the AS path as presented by the OLD BGP speaker and converts the 16-bit AS values to 32-bit AS values by adding 16 bits of zero padding to each entry, as before. However, in this case the NEW BGP speaker then overwrites the trailing entries with the values specified by the NEW_AS_PATH attribute. The effective result is that the 32-bit AS path that entered the 16-bit sequence is prepended with the equivalent of the 16-bit transit AS sequence. The NEW_AS_PATH attribute is then removed from the BGP Update, leaving an intact 32-bit path as the AS_PATH attribute. This scenario ensures that the resultant BGP environment can accurately detect loops in both the NEW 32-bit and OLD 16-bit realms (Figure 14).

Figure 14: OLD to NEW BGP AS Path Mapping



What if there was a routing loop that traversed a mixed sequence of NEW and OLD routing realms? The restoration of the original 32-bit AS path at the OLD-to-NEW transition ensures that the potential loop is discarded, because a 16-bit AS sees its own AS in the 16-bit AS_PATH attribute, and a 32-bit AS also sees its own value in the 32-bit AS_PATH. The transition mapping ensures that the potential routing loop is detected by BGP.

The ability to perform AS path prepending is also unaltered in this mixed NEW and OLD BGP environment. The AS simply prepends its local AS value to the AS_PATH as usual. In the case of prepending on a NEW-to-OLD boundary, the prepended AS path is mapped into the NEW_AS_PATH attribute as described previously.

Earlier in this article we noted the less common use of AS path poisoning, where the prepending uses a different ASN value in order to ensure that the particular advertisement is not learned by a remote AS. For NEW BGP speakers there is no change to this capability. For OLD BGP speakers the AS path poisoning can be directed only toward 16-bit ASs, because the OLD BGP speaker has no knowledge of the structure or content of the NEW_AS_PATH attribute.

Another part of the BGP protocol that uses ASNs is the AGGREGATOR attribute. This attribute is attached to an update message when an AS combines two or more prefixes into a single aggregate prefix (a practice that is often referred to as "proxy aggregation"). The ASN of the aggregating AS is attached to the aggregate prefix advertisement as an AGGREGATOR attribute. The same ASN translation technique applies to AGGREGATOR attribute when an advertisement is passed across a transition point. In a NEW-to-OLD transition the AGGREGATOR may be a mappable ASN, in which case the value is truncated to 16 bits and no further action is required. Otherwise the 32-bit AGGREGATOR value is rewritten into a NEW_AGGREGATOR attribute and the transition 16-bit value, AS 2356, is placed into the AGGREGATOR attribute. On an OLD-to-NEW transition the NEW_AGGREGATOR attribute is copied back into the AGGREGATOR attribute, if defined; otherwise the AGGREGATOR is padded out with leading zeros.

Transition

Transition in this scheme is relatively straightforward. NEW BGP speakers can be deployed within the network in a piecemeal fashion without any major concerns, and no changes are required for OLD BGP speakers. The size of BGP UPDATE messages is slightly longer because of the extended length of the AS PATH attribute in NEW BGP and the NEW_AS_PATH attribute that has been added in the OLD BGP environment, but it should not prove to be a major factor.

BGP loop prevention appears to be adequately addressed in all commonly encountered situations, and there appears to be no other significant transition considerations from the perspective of BGP platforms.

This scenario implies a relatively straightforward transition, in that OLD BGP speakers do not have to migrate to NEW BGP capability just because 32-bit ASNs are deployed elsewhere in the network. As long as they transmit the NEW-AS_PATH update across their domain without attempting to alter it in any way, then the 32-bit routing realm will be able to perform loop detection and shortest AS path selection in a manner that is entirely consistent with the 16-bit routing realm. Deployment of NEW BGP code is required only when the local AS is numbered from the nonmappable 32-bit ASN space.

Alternatives to AS Numbers

It is certainly a challenging task to contemplate an environment in which a 32-bit ASN space is exhausted, but one would suppose that the same consideration was in the minds of the original BGP protocol designers when they opted to use 16-bit ASNs. Of course a 32-bit number pool is not double the pool size of a 16-bit number pool—it is 65,536 times larger. That does appear to lead one to believe that this time it will be a far more challenging task to exhaust this expanded number pool.

This approach of simply extending the number space appears to offer a path of minimal disruption and minimal change in terms of operational configuration, storage, message size, and processing overheads for BGP. Nothing much has changed here except the range of the number space, and some ancillary considerations relating to transitional arrangements.

Of course, other labeling spaces remain possibilities, and a shift to a different labeling scheme could well use the same transitional approach. There is no significance in the ASN apart from its uniqueness, and any other form of name space would function equally well in terms of its role in BGP. One could use strings such as domain names, URIs, fixed-length hashes of public keys, the public keys themselves, or even IPv6 addresses as distinguishing AS identifiers.

There is no direct requirement for summarization of ASN ranges within the protocol use, no requirement within the protocol to continue to use number identifiers, and no direct requirement to stick with values that are encoded in a fixed-length field.

However, such approaches would add to the size of BGP UPDATE messages, increase the storage requirements, and, perhaps marginally, increase processing overheads for BGP. The more complex the identity space the more complex the basic task of BGP configuration and the higher the possibility of mistakes. "Borrowing" AS identifiers from another name space, such as domain names, or derived URIs, has the associated concern that the uniqueness of the space is derived from the inherent stability and uniqueness of the name space upon which the identifiers are derived. It is definitely possible that at times this trust is misplaced.

Numbers are often the simplest of identifiers. This approach represents minimal change to the installed base of BGP speakers, and there is no requirement for an existing routing domain using a 16-bit ASN and OLD BGP to make any changes to its routing environment at all. The transition appears to offer flexibility, orderly transition, and minimal disruptions to existing operational practices.

Conclusion

We are certainly running out of available 16-bit ASNs, and an industry of the size of the Internet is no longer as agile as it may have been in the past to make the necessary adjustments to alleviate this situation. At present we need to have a considerable period of advance warning of change in something as fundamental as the interdomain routing space in order to be able to integrate changes into various operational cycles of testing and transitional deployment prior to integration into production environments. The first steps that need to be taken are the completion of the technical specification of this approach in the form of an Internet standard and the production and distribution of BGP implementations that support 32-bit ASNs from the existing BGP implementation suppliers. It would be preferable to get this transition process under way in the near future, while there is still time to complete the transition well before we exhaust the current 16-bit ASN space.

For Further Reading

[1] "BGP Support for Four-octet AS Number Space," E. Chen, Q, Vohra, work in progress, (**draft-ietf-idr-as4bytes-12.txt**), November 2005.

The 32-bit AS description and the associated transition considerations. This work is expected to be completed shortly, and published as an RFC as a proposed standard document.

[2] "The AS Number Report", G. Huston, (updated on a daily basis) **http://www.potaroo.net/tools/asns**

A longer description of the numerical analysis used in the prediction of AS Number exhaustion.

[3] "ASN Missing in Action", H. Uijterwaal, R. Wilhelm, Document RIPE-353, (**http://www.ripe.net/docs/ripe-353.html**), October 2005.

Another analysis of AS Number consumption has been performed by Henk Uijterwaal and Rene Wilhelm, using the RIR AS number allocation rate as the base for the predictive exercise.

[4] "A Border Gateway Protocol 4 (BGP 4)," Y. Rekhter, Ed. T. Li, Ed. S. Hares, Ed., RFC 4271, January 2006.

[5] "Capabilities Advertisement with BGP-4," R. Chandra, J. Scudder, RFC 3392, November 2002.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for almost two decades, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector, and has served time with Telstra, where he was the Chief Scientist in the company's Internet area. Geoff is currently the Internet Research Scientist at the Asia Pacific Network Information Centre (APNIC). He has been a member of the Internet Architecture Board, and currently co-chairs two Working Groups in the IETF. He is author of a number of Internet-related books. E-mail: **gih@apnic.net**

Working with IP Addresses

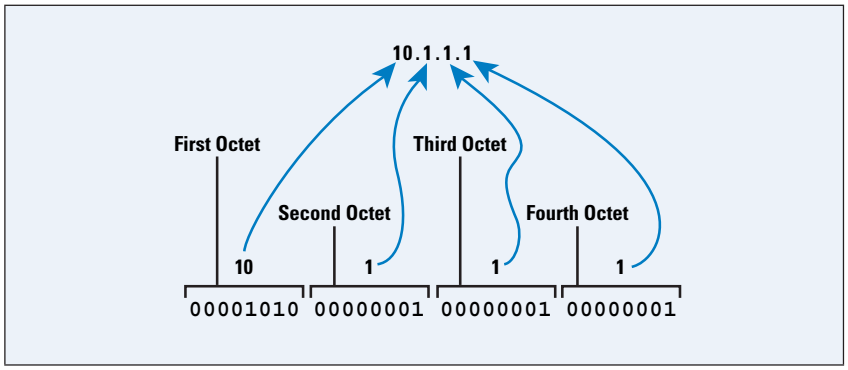
by Russ White, Cisco Systems

IP addresses, both IPv4 and IPv6, appear to be complicated when you first encounter them, but in reality they are simple constructions, and a using a few basic rules will allow you to find the important information for any situation very quickly—and with minimal math. In this article, we review some of the basics of IPv4 address layout, and then consider a technique to make working with IPv4 addresses easier. Although this is not the "conventional" method you might have been taught to work with in IP address space, you will find it is very easy and fast. We conclude with a discussion of applying those techniques to the IPv6 address space.

Basic Addressing

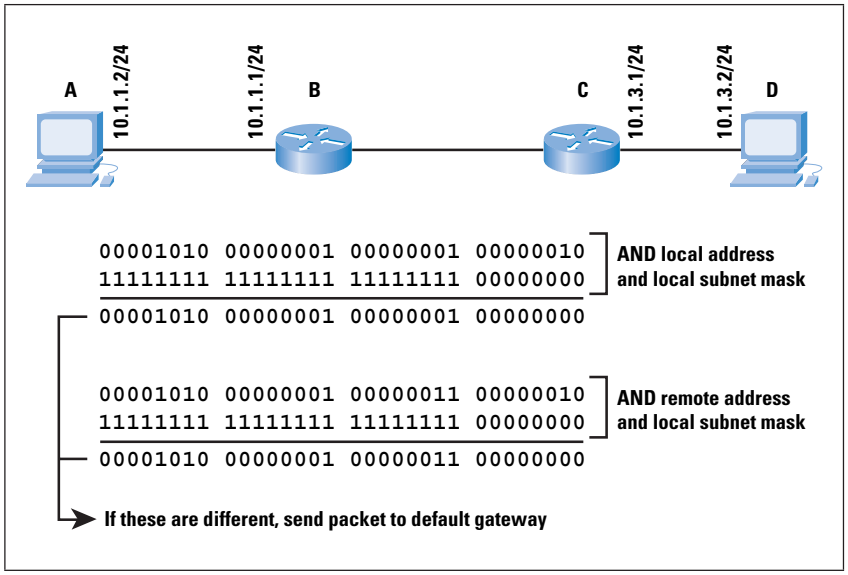
IPv4 addresses are essentially 32-bit binary numbers; computer systems and routers do not see any sorts of divisions within the IPv4 address space. To make IPv4 addresses more human-readable, however, we break them up into four sections divided by dots, or periods, commonly called "octets." An octet is a set of eight binary digits, sometimes also called a "byte." We do not use byte here, because the real definition of a byte can vary from computer to computer, whereas an octet remains the same length in all situations. Figure 1 illustrates the IPv4 address structure.

Figure 1: IPv4 Address Structure



Because each octet represents a binary (base 2) number between 0 and 2^8, each octet will be between 0 and 255. This part of IPv4 addresses is simple—but what about subnet masks? To understand a subnet mask, we need to understand how a device actually uses subnet masks to determine where to send a specific packet, as Figure 2 illustrates.

Figure 2: Subnet Masks



If host A, which has the local IP address **`10.1.1.2`** with a subnet mask of **`255.255.255.0`**, wants to send a packet to **`10.1.3.2`**, how does it know whether D is connected to the same network (broadcast domain) or not? If D is connected to the same network, then A should look for D's local Layer 2 address to transmit the packet to. If D is not connected to the same network, then A needs to send any packets destined to D to A's local default gateway.

To discover whether D is connected or not, A takes its local address and performs a logical AND between this and the subnet mask. A then takes the destination (remote) address and performs the same logical AND (using its local subnet mask). If the two resulting numbers, called the *network address* or *prefix*, match, then the destination must be on the local segment, and A can simply look up the destination in the *Address Resolution Protocol* (ARP) cache, and send the packet locally. If the two numbers do not match, then A needs to send the packet to its default gateway.

Note: ARP is a protocol used to discover the mappings between the IP addresses of devices attached to the same network as the local device and the Layer 2 address of devices attached to the same network as the local device. Essentially, a device sends an ARP broadcast containing the IP address of some other device it believes to be connected, and the device with the specified IP address replies with its Layer 2 address, providing a mapping between these two addresses.

If a subnet mask is a "dotted decimal" version of the binary subnet mask, then what is the prefix length? The prefix length is just a shorthand way of expressing the subnet mask. The prefix length is the number of bits set in the subnet mask; for instance, if the subnet mask is **`255.255.255.0`**, there are 24 1's in the binary version of the subnet mask, so the prefix length is 24 bits. Figure 3 illustrates network masks and prefix lengths.

Figure 3: Prefix Lengths

Binary Mask	Prefix Length	Subnet Mask
11111111 00000000 00000000 00000000	/8	255.0.0.0
11111111 10000000 00000000 00000000	/9	255.128.0.0
11111111 11000000 00000000 00000000	/10	255.192.0.0
11111111 11100000 00000000 00000000	/11	255.224.0.0
11111111 11110000 00000000 00000000	/12	255.240.0.0
11111111 11111000 00000000 00000000	/13	255.248.0.0
11111111 11111100 00000000 00000000	/14	255.252.0.0
11111111 11111110 00000000 00000000	/15	255.254.0.0
11111111 11111111 00000000 00000000	/16	255.255.0.0
11111111 11111111 10000000 00000000	/17	255.255.128.0
11111111 11111111 11000000 00000000	/18	255.255.192.0
11111111 11111111 11100000 00000000	/19	255.255.224.0
11111111 11111111 11110000 00000000	/20	255.255.240.0
11111111 11111111 11111000 00000000	/21	255.255.248.0
11111111 11111111 11111100 00000000	/22	255.255.252.0
11111111 11111111 11111110 00000000	/23	255.255.254.0
11111111 11111111 11111111 00000000	/24	255.255.255.0
11111111 11111111 11111111 10000000	/25	255.255.255.128
11111111 11111111 11111111 11000000	/26	255.255.255.192
11111111 11111111 11111111 11100000	/27	255.255.255.224
11111111 11111111 11111111 11110000	/28	255.255.255.240
11111111 11111111 11111111 11111000	/29	255.255.255.248
11111111 11111111 11111111 11111100	/30	255.255.255.252
11111111 11111111 11111111 11111110	/31	255.255.255.254
11111111 11111111 11111111 11111111	/32	255.255.255.255

Working with IPv4 Addresses

Now that we understand how an IPv4 address is formed and what the subnet length and prefix length are, how do we work with them? The most basic questions we face when working with an IP address follow:

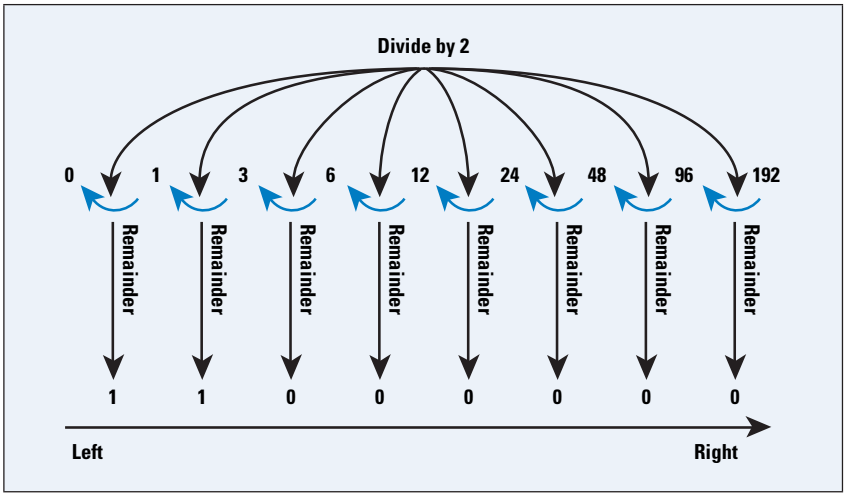
- What is the network address of the prefix?
- What is the host address?

There are two ways to find the answers to these questions: the hard way and the easy way. We cover the hard way first, and then show you the easy way.

The Hard Way

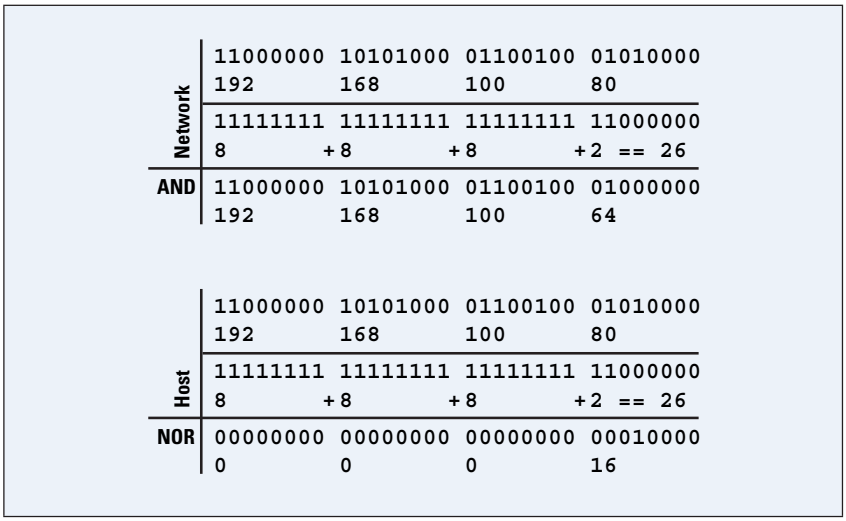
The hard way to determine the prefix and host addresses is to convert the address into binary, perform logical AND and NOR operations on the address and the subnet mask, and then convert the resulting numbers back to decimal. Figure 4 illustrates the process of converting a single octet of the IPv4 address into binary; the number converted in this case is 192.

Figure 4: Binary Conversion



The process is simple, but tedious; divide the octet value by 2, take the remainder off, and then divide by 2 again, until you reach 0. The remainders, reversed in direction, are the binary numbers representing the value of the octet. Performing this process for all four octets, we have the binary IP address, and can use logical AND and NOR operations to find the prefix (network address) and the host address, as Figure 5 shows for the address **`192.168.100.80/26`**.

Figure 5: Address Calculation



The Easy Way

All this conversion from binary to decimal and from decimal to binary is tedious— is there an easier way? Yes. First, we start with the observation that we work only with the numbers within one octet at a time, no matter what the prefix length is. We can assume all the octets before this *working octet* are part of the network address, and octets after this *working octet* are part of the host address.

The first thing we need to do, then, is to find out which octet is our *working octet*. This task is actually quite simple: just divide the prefix length by 8, discard the remainder, and add 1. The following table provides some examples.

Address	Hard Math	Working Octet
`192.158.100.80/26`	(26 ÷ 8) + 1 = 4	4
`10.1.1.48/23`	(23 ÷ 8) + 1 = 3	3
`172.31.80.10/22`	(22 ÷ 8) + 1 = 3	3

Note: Another way to look at this task is that you will ignore the octets indicated by the division. For instance, for **`192.168.100.80/26`**, *the result of dividing 26 by 8 is 3, so you will ignore the first three octets of the IP address, and work only with the fourth octet. This process has the same result.*

When we know the working octet, what do we do with it? Well, we could simply use the procedure outlined, convert the single octet to binary, perform AND and NOR operations on it with the right bits from the subnet mask, and then put it all back together to find the network and host addresses—but there is an easier way to find the network and host parts of the working octet. Start by doing the same math, only this time we want to work with the remainder rather than the result.

`192.168.100.80/26`
26 ÷ 8 = 3 with a remainder of 2

Take the remainder, and use the following table to find the corresponding *jump* within the octet; this number is the distance, in decimal form, between the network addresses within the octet.

1	2	3	4	5	6	7	8
128	64	32	16	8	4	2	1

In this chart, the first line represents the prefix length *within this octet*, the second line represents the prefix value when this bit is set to 1, the number of hosts in the subnet for this prefix length, and the *jump* between network addresses with the specified prefix length.

The number 2 corresponds to 64, so the jump is 64—there is a network at 0, 64, 128, 192, and 224 in this octet. Now all we need to do is figure out which one of those networks this address is in. This task is fairly simple: just take the largest network number that fits into the number in the working octet. In this case, the largest number that fits into 80 is 64, so our network address is **`192.168.100.64/26`**.

Now, what about the host address? That is easy when we have the network address—just subtract the network address from the IP address, and you have the host address within the network: 80 – 64 = 16. This process takes a little practice, but it is not hard when you become accustomed to the steps.

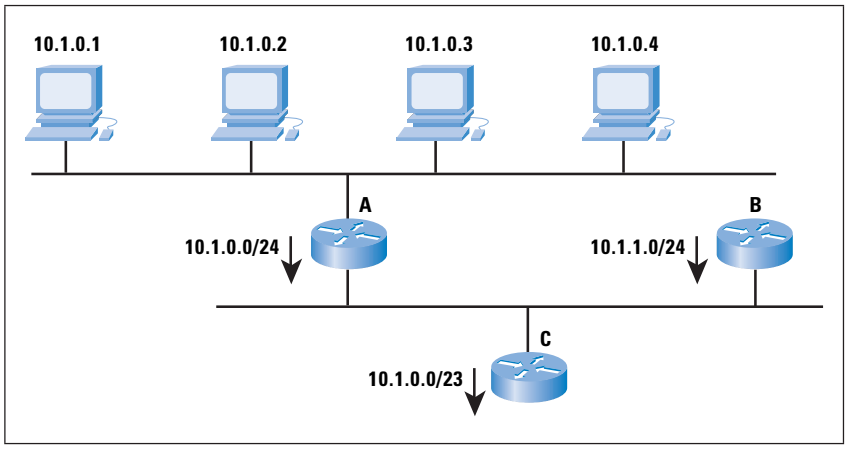
Address	Hard Math	Working Octet	Remainder	Jump	Network	Host
192.158.100.80/26	26 ÷ 8 = 3	3 + 1 = 4	2	64	192.168.100.64/26	80 – 64 = 16
10.1.1.48/23	23 ÷ 8 = 2	2 + 1 = 3	7	2	10.1.0.0/23	1 – 0 = 1.48
172.31.80.10/22	22 ÷ 8 = 2	2 + 1 = 3	6	4	172.31.80.0/22	80 – 80 = 0.10

In the second and third examples, you see that the working octet is actually the third, rather than the fourth, octet. To find the host address in these examples, you simply find the host address in the third octet, and then "tack on" the fourth octet as part of the host address as well, because part of the third octet—and all of the fourth octet—are actually part of the host address.

Summarization and Subnets

Subnets and supernets are probably the hardest part of IP addressing for most people to understand and handle quickly, but they are both based on a very simple concept—*aggregation*. Figure 6 shows how aggregation works.

Figure 6: Address Aggregation



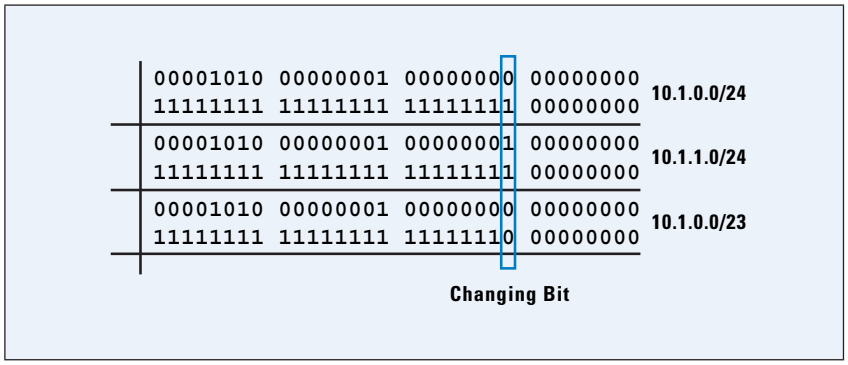
The figure shows four hosts with the addresses **10.1.0.1**, **10.1.0.2**, **10.1.0.3**, and **10.1.0.4**. Router A advertises **10.1.1.0/24**, meaning: "Any host within the address range **10.1.0.0** through **10.1.0.255** is reachable through me." Note that not all the hosts within this range exist, and that is okay—if a host within that range of addresses is reachable, it is reachable through Router A. In IP, the address that A is advertising is called a *network address*, and you can conveniently think of it as an address for the wire the hosts and router are attached to, rather than a specific device.

For many people, the confusing part comes next. Router B is also advertising **10.1.1.0/24**, which is another network address. Router C can combine—or aggregate—these two advertisements into a single advertisement. Although we have just removed the correspondence between the wire and the network address, we have not changed the fundamental meaning of the advertisement itself. In other words, Router C is saying: "Any host within the range of addresses from **10.1.0.0** through **10.1.1.255** is reachable through me." There is no wire with this address space, but devices beyond Router C do not know this, so it does not matter.

To better handle aggregated address space, we define two new terms, *subnets* and *supernets*. A subnet is a network that is contained entirely within another network; a supernet is a network that entirely contains another network. For instance, **10.1.0.0/24** and **10.1.1.0/24** are both subnets of **10.1.0.0/23**, whereas **10.1.0.0/23** is a supernet of **10.1.0.0/24** and **10.1.1.0/24**.

Now we consider a binary representation of these three addresses, and try to make more sense out of the concept of aggregation from an addressing perspective; Figure 7 illustrates.

Figure 7: Aggregation Details



By looking at the binary form of **10.1.0.0/24** and **10.1.1.0/24**, we can see that only the 24th bit in the network address changes. If we change the prefix length to 23, we have effectively "masked out" this single bit, making the **10.1.0.0/23** address cover the same address range as the **10.1.0.0/24** and **10.1.1.0/24** addresses combined.

The Hardest Subnetting Problem

The hardest subnetting problem most people face is that of trying to decide what the smallest subnet is that will provide a given number of hosts on a specific segment, and yet not waste any address space. The way this sort of problem is normally phrased is something like the following:

You have 5 subnets with the following numbers of hosts on them: 58, 14, 29, 49, and 3, and you are given the address space **10.1.1.0/24**. Determine how you could divide the address space given into subnets so these hosts fit into it.

This appears to be a very difficult problem to solve, but the chart we used previously to find the jump within a single octet actually makes this task quite easy. First, we run through the steps, and then we solve the example problem to see how it actually works.

- Order the networks from the largest to the smallest.
- Find the smallest number in the chart that fits the number of the largest number of hosts + 2 (*you cannot, except on point-to-point links, use the address with all 0's or all 1's in the host address; for point-to-point links, you can use a /31, which has no broadcast addresses*).
- Continue through each space needed until you either run out of space or you finish.

This process seems pretty simple, but does it work? Let's try it with our example.

- Reorder the numbers 58, 14, 29, 49, 3 to 58, 49, 29, 14, 3.
- Start with 58.
  - The smallest number larger than (58 + 2) is 64, and 64 is 2 bits.
  - There are 24 bits of prefix length in the address space given; add 2 for 26.
  - The first network is **10.1.1.0/26**.
  - The next network is **10.1.1.0** + 64, so we start the next "round" at **10.1.1.64**.
- The next block is 49 hosts.
  - The smallest number larger than (49 + 2) is 64, and 64 is 2 bits.
  - There are 24 bits of prefix length in the address space given; add 2 for 26.
  - We start this block at **10.1.1.64**, so the network is **10.1.1.64/26**.
  - The next network is **10.1.1.64** + 64, so we start the next "round" at **10.1.1.128**.
- The next block is 29 hosts.
  - The smallest number larger than (29 + 2) is 32, and 32 is 3 bits.
  - There are 24 bits of prefix length in the address space given; add 3 for 27.
  - We start this block at **10.1.1.128**, so the network is **10.1.1.128/27**.

  - The next network is **10.1.1.128** + 32, so we start the next "round" at **10.1.1.160**.

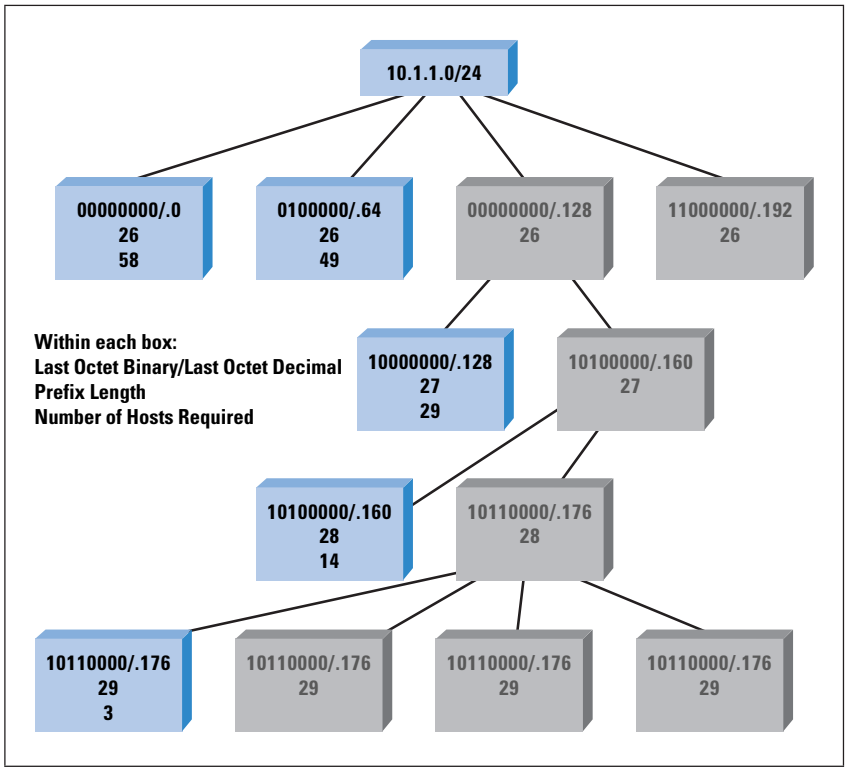
- The next block is 14 hosts.
  - The smallest number larger than (14 + 2) is 16, and 16 is 4 bits (actually equal, but it still works).
  - There are 24 bits of prefix length in the address space given; add 4 for 28.
  - We start this block at **10.1.1.160**, so the network is **10.1.1.160/28**.
  - The next network is **10.1.1.160** + 16, so we start the next "round" at **10.1.1.176**.

The last block is 3 hosts.

  - The smallest number larger than (3 + 2) is 8, and 8 is 5 bits.
  - There are 24 bits of prefix length in the address space given; add 5 for 29.
  - We start this block at **10.1.1.176**, so the network is **10.1.1.176/29**.
  - This is the last block of hosts, so we are finished.

It is a simple matter of iterating from the largest to the smallest block, and using the simple chart we used before to determine how large of a *jump* we need to cover the host addresses we need to fit onto the subnet. Figure 8 illustrates the resulting hierarchy of subnets.

Figure 8: Subnet Chart



In this illustration:

- The first line in each box contains the final octet of the network address in binary and decimal forms.
- The second line in each box contains the prefix length.
- The third line indicates the number of hosts the original problem required on that subnet.
- Gray boxes indicate blocks of address space that are unused at that level.

Working with IPv6 Addresses

IPv6 addresses appear to be much more difficult to work with—but they really are not. Although they are larger, they are still made up of the same fundamental components, and hosts and routers still use the addresses the same way. All we really need to do is realize that each pair of hexadecimal numbers in the IPv6 address is actually an octet of binary address space. The chart, the mechanisms used to find the network and host addresses, and the concepts of super and subnets remain the same.

For example, suppose we have the IPv6 address **`2002:FF10:9876:DD0A:9090:4896:AC56:0E01/63`** and we want to know what the network number is (host numbers are less useful in IPv6 networks, because they are often the MAC address of the system itself).

- 63 ÷ 8 = 7, remainder 7.
- The working octet is the 8th, which is 0A.
- Remainder 7 on the chart says the jump is 2, so the networks are **`00`**, **`02`**, **`04`**, **`06`**, **`08`**, **`0A`**, **`0C`**, and **`0E`**.
- The network is **`2002:FF10:9876:DD0A::/63`**.

The numbers are longer, but the principle is the same, as long as you remember that every *pair* of digits in the IPv6 address is a single octet.

Summary

IP addresses appear to be very complex on first approach, but their inbuilt structure actually provides easy ways to divide the problems into pieces and approach one piece of the problem at a time—the same way we design and build networks on a large scale. If you learn to use some simple techniques and understand how IP addresses are structured, they are relatively easy to work with.

For Further Reading

The following IETF *Requests for Comments* (RFCs) provide information on IP addressed and addressing structures:

[1] V. Fuller, T. Li, J. Yu, K. Varadhan, "Supernetting: an Address Assignment and Aggregation Strategy," RFC 1338, June 1992.

[2] E. Gerich, "Guidelines for Management of IP Address Space," RFC 1466, May 1993.

[3] Y. Rekhter, T. Li, "An Architecture for IP Address Allocation with CIDR," RFC 1518, September 1993.

[4] V. Fuller, T. Li, J. Yu, K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy," RFC 1519, September 1993.

[5] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, "Address Allocation for Private Internets," RFC 1918, February 1996.

RUSS WHITE works for Cisco Systems in the Routing Protocols Deployment and Architecture (DNA) team in Research Triangle Park, North Carolina. He has worked in the Cisco Technical Assistance Center (TAC) and Escalation Team in the past, has co-authored several books on routing protocols, including *Advanced IP Network Design, IS–IS for IP Networks,* and co-author of *Practical BGP.* He is the co-chair of the Routing Protocols Security Working Group within the IETF. E-mail: **riw@cisco.com**

Letter to the Editor

Dear Editor,

I read with interest the article by Dave Crocker in the December 2005 issue of IPJ (Volume 8, No. 4) titled "Challenges in Anti-Spam Efforts." However, I was surprised not to find any mention of *graylisting*, an effective anti-spam technique. The technique is not a programmatic or analytical approach to the spam problem but rather relies on exploiting the general behavioral weakness of spam delivery (that spammers typically want to try an address just once for their delivery of spam contents). The technique provides a pragmatic solution to the contemporary bulk commercial e-mail problem to a large extent.

If you are planning for a sequel of this article, I would strongly advocate mentioning the technique for the general benefit of the community.

I administrate a national ISP of considerable size in Pakistan, and the extent to which graylisting has helped us in fighting against spam is amazing.

Successful spam-fighting techniques (especially those that are still far from being widely adopted and worked upon) of today make good candidates for future efforts. My enthusiasm for graylisting is chiefly a result of the benefits our company has derived from its use, but I also want to champion its use because I think it is not widely adopted among peer ISPs because of ignorance. Hence my enthusiastic advocacy of this unsung hero in the fight against spam.

Citations:
Graylisting, **http://en.wikipedia.org/wiki/Graylisting**

—*Tee Emm, Supernet, Pakistan*
tm@super.net.pk

The Author responds:

Dear Editor,

I appreciate Tee Emm's concern that graylisting was not explicitly cited in my article.

I must use the cliché of "limited space" as my primary excuse for omitting graylistng. The tight constraints for a brief article required some difficult choices. As I mentioned at the end of the article, the people reviewing it before publication were particularly helpful (and vigorous). The question of what detail to include was a major focus. My decision was to have only a basic review of existing techniques, because the focus of the article was on future activities.

I believe the work on detection and reaction mechanisms against "bad actors" is reasonably mature. I also believe that the creation of a trust overlay for e-mail, to permit better handling of messages from "good actors," is very early and in need of much more focus.

With that said, I think I can also generate a plausible claim that graylisting is a form of "traffic shaping," which is cited in the article.

I primarily meant the traffic shaping reference to be about the technique of tracking aggregate (statistical) flow from a specific address. However, I think that graylisting constitutes a simple—albeit quite useful—mechanism that is designed to slow down the senders, to limit their impact. As Tee Emm notes, graylisting is based on the spammers' pattern of giving up, after a single failure to send the message. That is the ultimate "shaping," I think.

Certainly a summary of existing techniques is a worthy topic. It has become quite a rich topic, and matured to a level of qualifying as an area of administration and operations specialization.

As for a follow-up article, I do not have one planned, currently. If I do another one, I hope it will be about open mechanisms for achieving authentication and assessment (vetting) of good actors.

Perhaps next year.

For reference, I should note that there has been some public follow-up on the article, when CircleID reprinted a posting I made about it: **http://www.circleid.com/posts/challenges_in_anti_spam_efforts/**

Again, I appreciate Tee Emm's interest and comment.

—Dave Crocker, Brandenburg InternetWorking
dcrocker@bbiw.net

Fragments

IETF @ 20

The *Internet Engineering Task Force* (IETF) and the *Internet Society* (ISOC) celebrate the 20th anniversary of the IETF, the world's leading Internet standards development body. The IETF is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. Its principal task today is the development and publication of technical specifications and standards for basic Internet protocols. It is open to any interested individual.

The first IETF meeting was held on the afternoon of January 16, 1986, in San Diego, California. As a community-driven activity the IETF went on to pioneer a unique, open process for standards development. Open to all, and based on principles such as "rough consensus and running code," the IETF has enabled the development of standards that have supported every aspect of the Internet's phenomenal growth.

"The IETF is unique," said Brian Carpenter, IETF Chair. "Unlike other standards bodies, there is very little in the way of formal hierarchy and there are no membership requirements or fees. The IETF welcomes broad participation by anyone interested in the future technical evolution and stability of the Internet—and IETF standards are available to all, without charge."

"There is global recognition of the achievements of the IETF in its support of the development of Internet technology. As the demands on the Internet increase, the IETF clearly has a vital role to play in ensuring that Internet technologies continue to evolve in a coherent and coordinated manner," said Leslie Daigle, chair of the *Internet Architecture Board* (IAB) which provides architectural oversight of IETF activities."

"The success of the IETF has largely been due to a pragmatic, consensus-based approach to technology standards development," noted Lynn St. Amour, President and CEO of ISOC. "Many of the principles of co-operation and collaboration that were developed in the IETF are now being successfully applied in other global forums. ISOC is proud to be associated with the IETF—we value its members' accomplishments over the last 20 years and look forward to celebrating these achievements over the course of 2006."

ISOC has declared 2006 "The Year of the IETF" and will be running several activities during the year in celebration of the IETF's 20th anniversary. For more information, see: **`http://ietf20.isoc.org`**

The Internet Society is a not-for-profit membership organization founded in 1992 to provide leadership in Internet related standards, education, and policy. With offices in Washington, DC, and Geneva, Switzerland, it is dedicated to ensuring the open development, evolution and use of the Internet for the benefit of people throughout the world. ISOC is the organizational home of the IETF and other Internet-related bodies who together play a critical role in ensuring that the Internet develops in a stable and open manner. For over 13 years ISOC has run international network training programs for developing countries and these have played a vital role in setting up the Internet connections and networks in virtually every country connecting to the Internet during this time.

ISOC Welcomes WSIS Proposal

Delegates meeting at the *World Summit on the Information Society* (WSIS) in Tunis have affirmed their commitment to build on the governance mechanisms that have enabled the Internet's incredibly successful growth.

ISOC welcomes the recognition by WSIS of how the effectiveness of the existing arrangements for Internet governance has helped make the Internet the highly robust, dynamic and geographically diverse medium that it is today.

"We are delighted that there is now much broader recognition of the achievements of the organisations that support the Internet community," said Lynn St. Amour, President and CEO of the ISOC. "These organizations, along with their open, consensus-based processes clearly have a vital role to play in the further development of the Internet. It is also significant that the WSIS debate has moved beyond the details of technical administration and on to broader issues that require increased coordination by stakeholders in order to ensure the continued stability of the Internet."

The WSIS recommendation includes a proposal for a new forum for multi-stakeholder policy dialogue—the *Internet Governance Forum*. ISOC, together with partner organizations from the Internet community, has always worked to encourage full engagement in such dialogues by all those with an interest in the Internet's future. ISOC believes that the forum's success depends upon the fullest participation by all stakeholders. At the same time, ISOC is pleased to note that the proposed forum would have no oversight function and would have no involvement in the day-to-day operations of the Internet.

"ISOC will facilitate increased cooperation and information sharing amongst all parties interested in Internet governance and we look forward to playing an active role in the new forum as is expected of us by the global community," said Lynn St. Amour. "We very much hope that the Tunis summit will lead to some real and positive outcomes that will help bring the benefits of the Internet to people everywhere—especially to those who are yet to be connected."

ISOC, along with some of its partner organisations—the *Number Resource Organisation* (NRO), the IETF, *London Internet Exchange* (LINX), the *Internet Corporation for Assigned Names and Numbers* (ICANN) and the *Council of European National Top level domain Registries* (CENTR)—were present at the *ICT 4 All* exhibition held in conjunction with WSIS.

For more information about the organizations listed above visit:

http://isoc.org

http://ietf.org

http://iab.org

http://www.intgovforum.org

http://www.linx.net

http://nro.org

http://www.centr.org

http://www.itu.int/wsis

Upcoming Events

The *Internet Engineering Task Force* (IETF) will meet in Montreal, Canada, July 9–14, 2006. For more information, visit:
http://ietf.org

ACM's *SIGCOMM 2006* will be held in Pisa, Italy, September 11–15, 2006. For more information, visit:
http://www.acm.org/sigs/sigcomm/sigcomm2006

The *North American Network Operators Group* (NANOG) will meet in St. Louis, MO October 8–10, 2006. For more information, see: **http://nanog.org**

The *American Registry for Internet Numbers* (ARIN) will meet (jointly with NANOG) in St. Louis, October 11–13, 2006. For more information, see: **http://arin.net**

The Internet Protocol Journal
Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

June 2006 Volume 9, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

In our June 2000 issue we wrote: "Two protocols used in the Internet are so important that they deserve special attention: the *Internet Protocol* (IP) from which this journal takes its name, and the *Transmission Control Protocol* (TCP). IP is fundamental to Internet addressing and routing, while TCP provides a reliable transport service that is used by most Internet applications, including interactive Telnet, file transfer, electronic mail, and Web page access via HTTP. Because of the critical importance of TCP to the operation of the Internet, it has received much attention in the research community over the years. As a result, numerous improvements to implementations of TCP have been developed and deployed." We return to TCP in this issue with a look at its performance at gigabit speeds. Geoff Huston describes numerous research proposals related to TCP and discusses lessons learned by operators and researchers involved with this protocol.

My first encounter with the Internet (then called the ARPANET) took place in 1976 when I visited the *Norwegian Defence Research Establishment* (NDRE) at Kjeller, about 20 kilometers from Oslo, Norway. At NDRE, one of the researchers, named Pål, showed me a teletype terminal that was connected through the ARPANET to a host computer at SRI International in Menlo Park, California. After a few minutes, the teletype started printing messages from someone called "Geoff" on the other end of the line. Pål typed back, passing on questions from myself about the weather in California and so on. I later learned that the host computer was a PDP-10 model KA10 running the TENEX operating system. TENEX could "link" two terminals together so that anything typed on one terminal would appear on the other, and conversely. This primitive "chat" system is the forerunner of today's *Instant Messaging* (IM) environment. David Strom gives an overview of the current state of IM solutions in our second article.

The article "Working with IP Addresses" in our last issue sparked several comments, some of which are included in our Letters to the Editor section. A few readers also noticed some errors in the article, so we have included the corrections in this issue. We very much appreciate your feedback. Please send your comments to: **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Gigabit TCP

by Geoff Huston, APNIC

In looking back over some 30 years of experience with the Internet, the critical component of the Internet Protocol Suite that has been the foundation of its success as the technology of choice for the global communications system is the *Internet Protocol* (IP) itself, working an overlay protocol that can span almost any form of communications media. But I would also like to nominate another contender for a critical role within IP, namely the reliable transport protocol that sits on top of IP, the *Transmission Control Protocol* (TCP), and its evolution over time. In support of this nomination is the fact that the end-to-end rate-adaptive control algorithm that was adopted by TCP represented a truly radical shift from the reliable gateway-to-gateway virtual circuit flow control systems used by other protocols of similar vintage. It is also interesting to note that TCP is not designed to operate at any particular speed, but it attempts to operate at a speed that uses its fair share of all available network capacity along the network path. The fundamental property of the TCP flow control algorithm is that it attempts to be maximally efficient while also attempting to be maximally fair.

Previous articles on this topic, "TCP Performance"[12] and "The Future for TCP"[13] looked at the design assumptions behind TCP and its performance characteristics. The essential characteristic of TCP is that it attempts to establish a dynamic equilibrium with other concurrent sessions and opportunistically use all available network capacity. It achieves this by constantly altering its flow characteristics, continually probing the network to see if higher speeds are supportable, while also being prepared to immediately decrease the current sending rate in the face of received signals of network congestion.

In a world where network infrastructure capacity and complexity are related to network cost and delivered data is related to network revenue, TCP fits in well. The minimal assumptions that TCP makes about the capability of network components permit networks to be constructed using simple transmission capabilities and simple switching systems. "Simple" often is synonymous with cheap and scalable, and there is no exception here. TCP also attempts to maximize data delivery through adaptive end-to-end flow rate control and careful management of retransmission events. In other words, TCP is an enabler for cheaper networking for both the provider and consumer. For the consumer the offer of fast cheap communications has been a big motivation in the increase in demand for Internet-based services, and this—more than any other factor—has been the major enabling factor for the increased use of the Internet itself. "Cheap" is often enough in this world, and TCP certainly helps to make data communications efficient and therefore cheap.

Although TCP is highly effective in many networking environments, that does not mean it is highly effective in every environment. For example:

- In those wireless environments where there is significant wireless noise, TCP may confuse the outcome of radio-based signal corruption and the corresponding packet drop with the outcome of network congestion, and consequently the TCP session may back off its sending rate too early and back off for too long.
- TCP also backs off too early when the network routers have insufficient buffer space. This effect is more subtle, but it is related to the coarseness of the TCP algorithm and the consequent burstiness of TCP packet sequences. These bursts, which occur at up to twice the bottleneck capacity rate, are smoothed out by network buffers. Buffer exhaustion in the interior of the network causes packet drop, which causes the generation of a loss signal to the active TCP session, which, in turn, either halves its sending rate or—in the worst case—resets the session state and restarts with a single packet exchange. Particularly in wide-area networks, where the end-to-end delay-bandwidth product becomes a significant factor, TCP uses the network buffers to sustain a steady-state throughput that matches the available network capacity. Where the interior buffers are underconfigured in memory it is not possible to even out the TCP bursts to continuously flow through the constrained point at the available data rate.
- TCP also asks its end hosts to have local capacity equal to the available network capacity on the forward and reverse paths. The reason is that TCP does not discard data until the remote end has reliably acknowledged it, so the sending host has to retain a copy of the data for the time it takes to send the data plus the time for the remote end to send the matching acknowledgement.

Even accounting for these limitations, it is true to say that TCP works amazingly well in most environments. Nevertheless, one area is proving to be quite a fundamental challenge to TCP as we know it, and that is the domain of wide-area, very-high-speed data transfer.

Very-High-Speed TCP

End host computers, even laptop computers these days, are typically equipped with Gigabit Ethernet interfaces, and have gigabytes of memory and internal data channels that can move gigabits of data per second between memory and the network interface. Current IP networks are constructed using multigigabit circuits and high-capacity switches and routers (assuming there is still a quantitative difference between these two forms of packet switching equipment). If the end hosts and the network both can support gigabit transmissions then a TCP session should be able to operate end to end at gigabits per second, and achieve the same efficiency at gigabit speeds as it does today at megabit speeds—right?

Well, no, not exactly!

This conclusion is not obvious, particularly when the TCP Land Speed Record is now at some 7Gbps across a distance that spans 30,000 km of network. What is going on?

Let's return to the basics of TCP to understand some of the variables with very-high-speed TCP. TCP operates in one of two states, that of *slow start* and *congestion avoidance*.

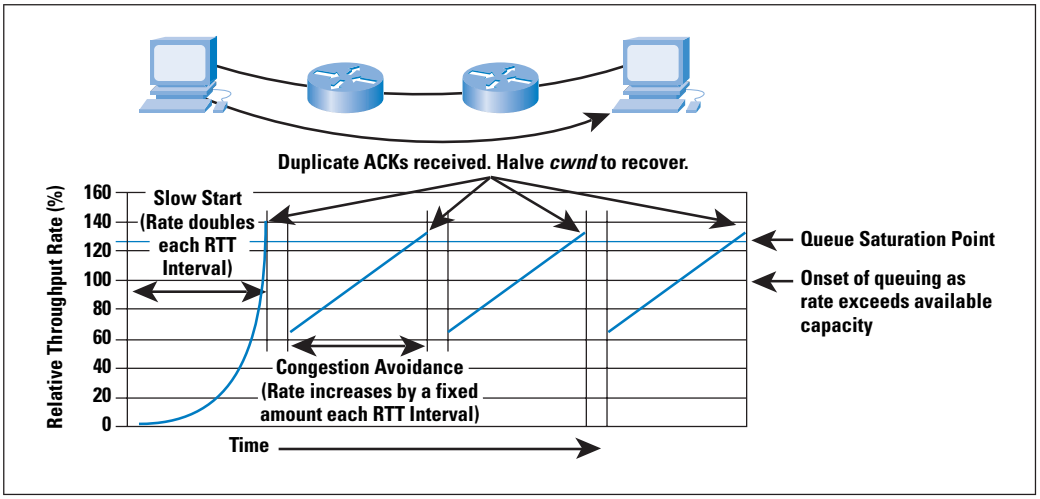
- *Slow start* mode is the initial mode of operation of TCP in any session, as well as its "reset" mode. In this mode, TCP sends two packets in response to each ACK packet that advances the sender's window. In approximate terms (*delayed* ACKs notwithstanding), this mode allows TCP to double its sending rate in each successive lossless *round-trip-time* (RTT) interval. The rate increase is exponential, effectively doubling each RTT interval, and the rate increase is bursty, effectively sending data into the network at twice the bottleneck capacity during this phase.

  Sending data into the network at twice the bottleneck data speed is possible because of the "ACK clocking" property of TCP. Disregarding the complications of the TCP delayed ACK mechanism for a second, a TCP receiver generates a new ACK packet each time a packet arrives at the receiver. The sending rate of the ACKs is, in effect, the same as the receiving rate for the data packets. Assuming a one-way data transfer, so that ACK packets in the reverse direction are of minimal size, and assuming minimal jitter on the reverse path from the receiver back to the sender, the arrival rate of ACKs at the sender is comparable to the arrival rate of data packets at the receiver. In other words, the return ACK rate is comparable to the bottleneck capacity of the forward network path from sender to receiver. Sending two packets per received ACK is effectively sending packets into the network at twice the bottleneck capacity. At the bottleneck point the switching unit receives twice the amount of data than it can transmit to the output device over a period that corresponds to the delay-bandwidth product of the bottleneck link. Hence the comment that TCP is a *bursty* protocol, particularly at startup. For this reason TCP tends to operate more effectively across network switching elements that are generously endowed with memory, or have for each output port a buffer capacity roughly equal to the delay-bandwidth product of the link that is attached to that port.

- In the other operating mode, that of *congestion avoidance,* TCP sends an additional segment of data for each loss-free round-trip time interval. This increase is additive rather than exponential, increasing the sender's speed at the constant rate of one segment per RTT interval.

TCP undertakes a state transition upon the detection of packet loss. Small-scale packet loss (of the order of 1 or 2 packets per loss event) causes TCP to halve its sending rate and enter congestion avoidance mode, irrespective of whether it was in this mode already. Repetition of this cycle gives the classic sawtooth pattern of TCP behavior, and the related derivation of TCP performance as a function of packet loss rate. Longer sustained packet loss events cause TCP to stop using the current session parameters, recommence the congestion control session using the restart window size, and enter the slow start control mode once again. (See Figure 1).

Figure 1: TCP Behavior



But what happens when two systems are at opposite sides of a continent with a high-speed path between them? How long does it take for a single TCP session to get up to a data transfer rate of 10 Gbps? Can a single session operate at a sustained rate of 10 Gbps?

Let's look at a situation such as the network path from Brisbane, on the eastern side of the Australian continent, to Perth on the western side. The cable path is essentially along the southern coast of the continent, so the RTT delay is 70 ms, implying that there are 14.3 round-trip intervals per second. Let's also assume that the packet size being used is 1500 octets, or 12,000 bits, and the TCP initial window size is a single packet. And let's also assume that the bottleneck capacity of the host-to-host path between Brisbane and Perth is 10 Gbps.

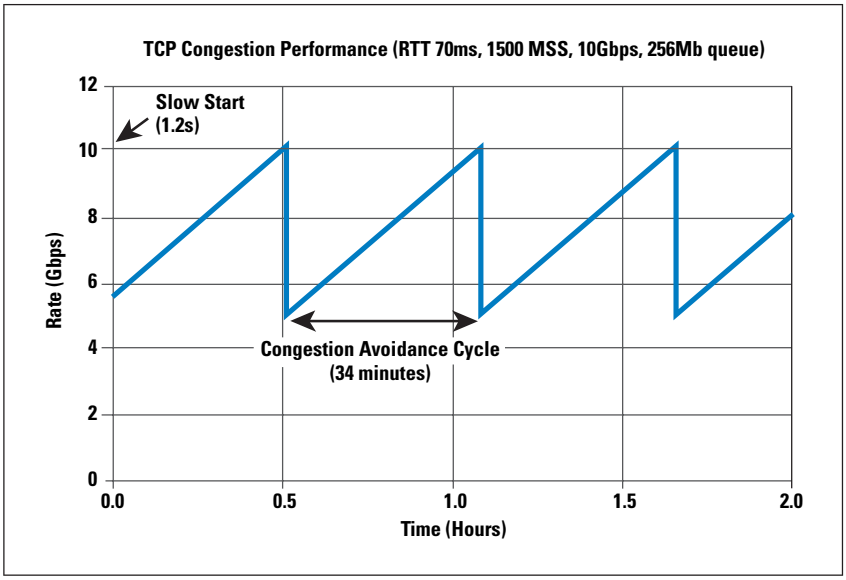
In a simple slow start model the sending speed doubles every 70 ms, so after 17 RTT intervals where the sending rate has doubled for each interval, or after some 1.2 seconds have elapsed, the transfer speed reaches 11.2 Gbps (assuming a theoretical host with sufficiently fast hardware components, sufficiently fast internal data paths, and adequate memory). At this stage let's assume that the sending rate exceeded the buffer capacity at the bottleneck point in the network path. Packet drop will occur, because the critical point buffers in the network path are now saturated.

At the point of reception of an ACK sequence that signals packet loss, the TCP sender's congestion window will halve, as will the TCP sending rate, and TCP will switch to congestion avoidance mode. In congestion avoidance mode the rate increase is 1 segment per RTT, equivalent to sending an additional 12 kilobits per RTT, or, given the session parameters as specified previously, equivalent to a rate increase of 171 kbps each RTT. So how long will it take TCP to recover and get back to a sending rate of 10 Gbps?

If this were a T1 circuit where the available path bandwidth is 1.544 Mbps, and congestion loss occurred at a sending rate of 2 Mbps (higher than the bottleneck transmission capacity due to the effect of queuing buffers within the network), then TCP would rate halve to 1 Mbps and then use congestion avoidance to increase the sending rate back to 2 Mbps. Within the selected parameters of a 70-ms RTT and 1500-byte segment size, this process involves using congestion avoidance to inflate the congestion window from 6 segments to 12. This process takes 0.42 seconds. So as long as the network can operate without packet loss for the session over an order of 1-second intervals, then TCP can comfortably operate at maximal speed in a megabit-per-second network.

What about our 10-Gbps connection? The first estimate is the amount of usable buffer space in the switching elements. Assuming a total of 256 MB of usable queue space on the network path prior to the onset of queue saturation, the TCP session operating in congestion avoidance mode will experience packet loss some 590 RTT intervals after reaching the peak transmission speed of 10 Gbps, or some further 41 seconds, at which point the TCP sending rate in congestion avoidance mode is 10.1 Gbps. For all practical purposes the TCP congestion avoidance mode causes the sawtooth oscillation of this ideal TCP session between 5.0 Gbps and 10.1 Gbps. A single iteration of this sawtooth cycle takes 2062 seconds, or 34 minutes and 22 seconds. The implication here is that the network has to be stable in terms of no packet loss along the path for time scales of the order of tens of minutes (or some billions of packets), and corresponding transmission bit error rates that are less than 10^{-14}. It also implies massive data sets to be transferred, because the amount of data passed in just one TCP congestion avoidance cycle is 1.95 terabytes (1.95×10^{12} bytes). It is also the case that the TCP session cannot make full use of the available network bandwidth, because the average data transfer rate is 7.55 Gbps under these conditions, not 10 Gbps. (See Figure 2).

Figure 2: TCP Behavior at High Speed



Clearly something is unexpected with this scenario, because it certainly looks like it is a difficult and lengthy task to fill a long-haul, high-capacity cable with data, and TCP is not behaving as expected. Although experimenting with the boundaries of TCP is in itself an interesting area of research, some practical problems here could well benefit from this type of high-speed transport.

A commonly quoted example, and certainly one of the more impressive ones is the Large Hadron Collider at CERN:

"The CERN Particle Physics lab in Geneva, Switzerland, successfully transmitted a data stream averaging 600Mbytes per second for 10 days to seven countries in Europe and the US. It was a crucial test of the computing infrastructure for the Large Hadron Collider being built at CERN. The LHC will be the most data intensive physics instrument ever built, generating 1500 Megabytes every second for a decade or more."

—*New Scientist,* 30 April 2005

TCP and the Land Speed Record

The TCP Land Speed Record was originally an informal effort to achieve record-breaking TCP transfer speeds across IP networks. The late 1980s and early 1990s saw some noted milestones, particularly with Van Jacobson's efforts in achieving sustained 10-Mbps and 45-Mbps TCP transfer speeds.

This activity has been incorporated into the Internet2 program, with the introduction of some formal rules about what constitutes a TCP Land Speed effort. In particular, the rules now have times, distances, and TCP constraints, and they call for the use of operational networks. Updates to the record have been posted frequently in recent years, and as of May 2006 the IPv4 single stream record is a TCP session operating at 7.21 Gbps for 30 minutes over 30,000 km of fibre path.

It is certainly possible to have TCP perform for sustained intervals at very high speed, as the land speed records for TCP show, but something else is happening here, and a set of preconditions need to be met before attempting to set a new record:

- First, it is good—indeed essential—to have the network path all to yourself. Any form of packet drop is a major problem here, so the best way to ensure no packets are lost is to keep the network path all to yourself.
- Secondly, it is good—indeed essential—to have a fixed latency. If the objective of the exercise is to reach a steady-state data transmission, then any change in latency, particularly a reduction in latency, has the risk of a period of oversending, which in turn has a risk of packet loss. So keep the network as stable as possible.
- Thirdly, it is good—indeed essential—to have extremely low bit error rates from the underlying transmission media. Data corruption causes checksum failure, which causes packet drop.
- Lastly, it is essential to know in advance both the round-trip latency and the available bandwidth.

You can then multiply these two numbers together (RTT and bandwidth), divide by the packet size, round down, and be sure to configure the sending TCP session to have precisely this buffer size, and the receiver to have a slightly larger size. And then start up the session.

The intention here is for TCP to use slow start to the point where the sender runs out of buffer space, at which point it will continue to sit at this buffer speed for as long as the sender, receiver and network path all remain in a stable state. For the example configuration of a 10-Gbps system with 70 ms RTT, setting a buffer limit of 116,000 packets will cause the TCP session to operate at 9.94 Gbps. As long as the latency remains steady (no jitter), with no bit errors, and as long as there is no other cross traffic, in theory this sending rate can be sustained indefinitely, with a steady stream of data packets being matched by a steady stream of ACK packets.

Of course, this situation is artificially constrained. The real concerns here with the protocol are in the manner in which it shares a network path with other concurrent sessions as well as its ability to fill the available network capacity. In other words, what would be good to see is a high-speed, high-volume version of TCP that could coexist on a network with all other forms of traffic, and, perhaps more ambitiously, that this high-speed form of TCP could share the network fairly with other traffic sessions while at the same time making maximal use of the network. The problem with TCP in its current incarnation is that it takes way too long in its additive increase mode (congestion avoidance) to recover its sustainable operating speed when operating at high speed across transcontinental-size network paths. If we want very-high-speed TCP to be effective and efficient, then we need to look at changes to TCP for high-speed operation.

High-Speed TCP

There are two basic approaches to high-speed TCP: parallelism of existing TCP, or changes to TCP to allow faster acceleration rates in a single TCP stream.

Using parallel TCP streams as a means of increasing TCP performance is an approach that has existed for some time. The original HTTP specification, for example, allowed the use of parallel TCP sessions to download each component of a Webpage (although HTTP 1.1 reverted to a sequential download model because the overheads of session startup appeared to exceed the benefits of parallel TCP sessions in this case). Another approach to high-speed file transfer through parallelism is that of GRID FTP. The basic approach is to split up the communications payload into numerous discrete components, and send each of these components simultaneously. Each component of the transfer can be between the same two endpoints (such as GRID FTP), or can be spread across multiple endpoints (as with BitTorrent).

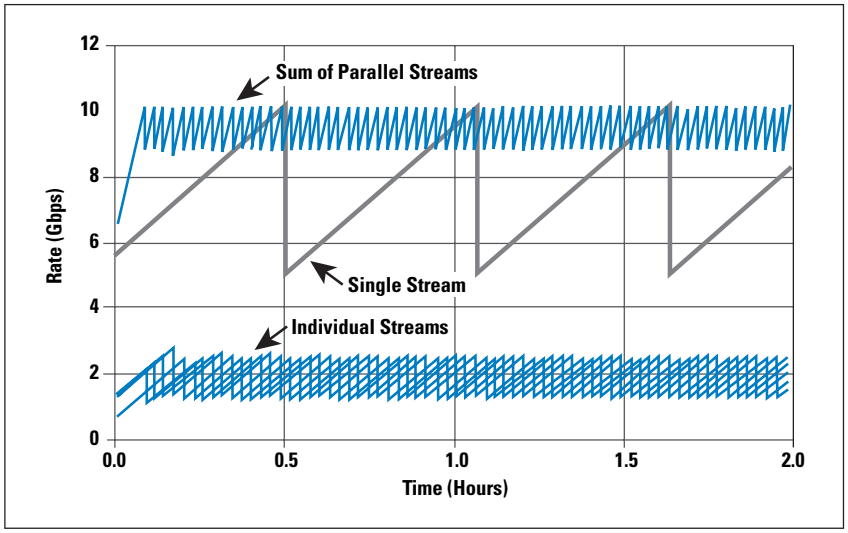
But for parallel TCP to operate correctly, we need to have already assembled all the data (or at a minimum know where all the data components are located). Where the data is being generated in real time (such as observatories or particle colliders) in massive quantities, there may be no choice but to treat the data set as a serial stream and use a high-speed transport protocol to dispatch it. In this case the task is to adjust the basic control algorithms for TCP to allow it to operate at high speed, but also to operate "fairly" on a mixed-traffic high-speed network.

Parallel TCP

Using parallelism as a key to higher speed is a common computing technique, and lies behind many supercomputer architectures. The same can apply to data transfer, where a data set is divided into numerous smaller chunks, and each component chunk is transmitted using its own TCP session. The underlying expectation here is that when using some number, *N*, of parallel TCP sessions, a single packet drop event will most probably cause the fastest of the *N* sessions to rate halve, because the fastest session will have more packets in flight in the network, and is therefore the most likely session to be impacted by a packet drop event. This session will then use congestion avoidance rate increase to recover, implying that the response to a single packet drop is reduction of the sending rate by at most $1/(2N)$. For example, using five parallel TCP sessions, the response to a single packet drop event is to reduce the total sending rate by $1/(2 \times 5)$, or 1/10, as compared to the response from a single TCP session, where a single packet drop event would reduce the sending rate by ½.

A simulated version of five parallel sessions in a 10 Gbps session is shown in Figure 3.

Figure 3: Parallel TCP Simulation: Single vs Parallel Streams



The essential characteristic of the aggregate flow is that under lossless conditions the data flow of N parallel sessions increases at a rate N times faster than a single session in congestion avoidance mode. Also the response to an isolated loss event is that of rate halving of a single flow, so that the total flow rate under ideal conditions is between R and $R \times (2N - 1)/2N$, or a long-term average throughput of $R \times (4N - 1)/4N$. For $N = 100$ our theoretical 10-Gbps connection could now operate at 9.9 Gbps.

Of course practice is different from theory, and a considerable amount of work has looked at the performance of parallel TCP under various conditions, in terms of both maximizing throughput and choosing the most efficient number of parallel active streams to use. Part of the problem is that although simple simulations, such as that used to generate Figure 4, tend to evenly distribute each of the parallel sessions to maximize the throughput, there is the more practical potential that the individual sessions self-synchronize. Because the parallel sessions have a similar range of window sizes, it is possible that at a given point in time a similar number of packets will be in the network path from each stream. If the packet drop event is a multiple packet drop event, such as a tail-drop queue, then it is entirely feasible that numerous parallel streams will experience packet loss simultaneously, and there is the consequential potential for the streams to fall into synchronization.

The two extremes, evenly distributed and tightly synchronized multiple streams, are indicated in Figure 4 The average throughput of parallel synchronized streams is the same as a single stream over extended periods in this simulation, and both are certainly far worse than an evenly distributed set of parallel streams.

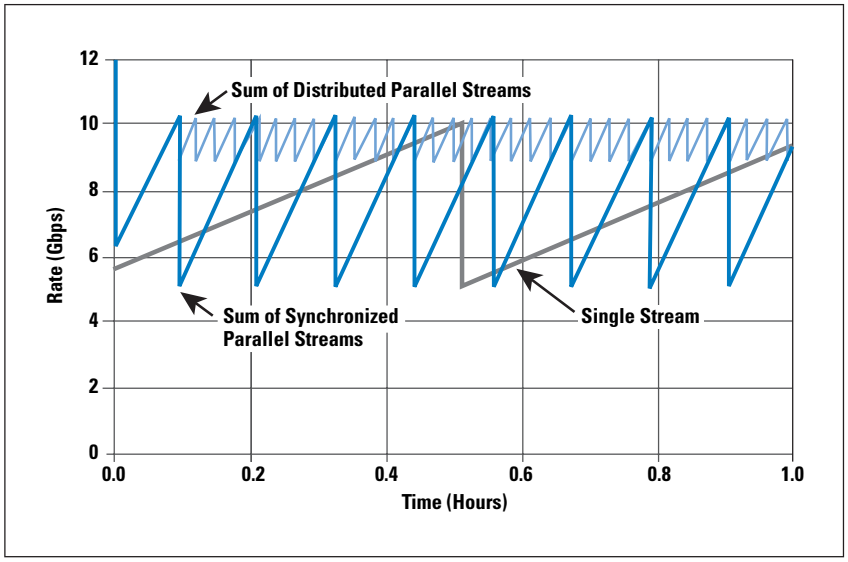


Figure 4: Comparison of Parallel TCP: Synchronized and Distributed Streams

One way to address this problem is to reunite these parallel streams into a single controlled stream that exhibits the same characteristics as evenly spread parallel streams. This approach, MulTCP, is considered in the next section.

If all this analysis of parallel TCP streams sounds a little academic and unrelated to networking today, it is useful to note that many *Internet Service Providers* (ISPs) currently see *BitTorrent* traffic as their highest-volume application. BitTorrent is a peer-to-peer protocol that undertakes transfer of datasets using a highly parallel transfer technique. Under BitTorrent the original dataset is split into blocks, each of which can be downloaded in parallel. The subtle twist here is that the individual sessions do not have the same source points, and the host may take feeds from many different sources simultaneously, as well as offering itself as a feed point for the already downloaded blocks. This behavior exploits the peer-to-peer nature of these networks to a very high extent, potentially not only exploiting parallel TCP sessions for speed gains, but also exploiting diverse network paths and diverse data sources to avoid single path congestion. Considering its effectiveness in terms of maximizing transfer speeds for high-volume datasets and its relative success in truly exploiting the potential of peer-to-peer networks—and of course the dramatic acceptance of BitTorrent and its extensive use—BitTorrent probably merits closer examination, but perhaps that is for another time and an article of its own.

Very High Speed Serial TCP

The other general form of approach is to reexamine the current TCP control algorithm to see if there are parameter or algorithm changes that could allow TCP to undertake a better form of rate adaptation to these high-capacity, long-delay network paths. The aim here is to achieve a good congestion response algorithm that does not amplify transient congestion conditions into sustained disaster areas, while at the same time being able to support high-speed data transfers, thereby making effective use of all available network capacity.

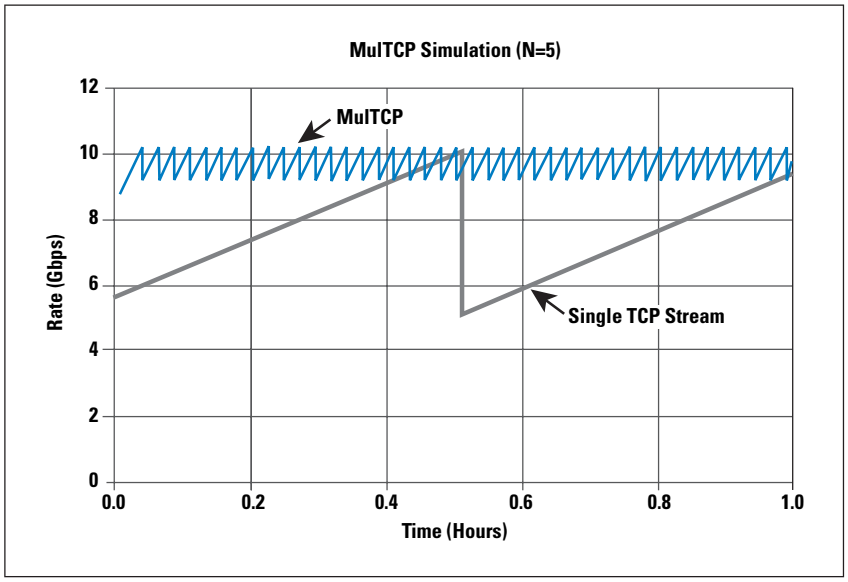
We also want TCP to behave sensibly in the face of other TCP sessions, so that it can share the network with other TCP sessions fairly.

MulTCP

The first of these approaches is *MulTCP*[1], which is a single TCP stream that behaves in a manner equivalent to *N* parallel TCP sessions, where the virtual sessions are evenly distributed in order to achieve the optimal outcome in terms of throughput. The essential changes to TCP are in congestion avoidance mode and the reaction of packet loss. In congestion avoidance mode MulTCP increases its congestion window by *N* segments per RTT, rather than the default of a single segment. Upon packet loss, MulTCP reduces its window by W/(2*N*), rather than the default of W/2. MulTCP uses a slightly different version of slow start, increasing its window by 3 segments per received ACK, rather than the default value of 2.

MulTCP represents a simple change to TCP that does not depart radically from the TCP congestion control algorithm. Of course when choosing an optimal value for *N*, some understanding of the network characteristics would help. If the value for *N* is too high, the MulTCP session has a tendency to claim an unfair amount of network capacity, but if the value is too low, it does not necessarily take full advantage of available network capacity. Figure 5 shows MulTCP compared to a simulation of an equivalent number of parallel TCP streams and a single TCP stream (*N* = 5 in this particular simulation).

Figure 5: MulTCP



Good as this is, there is the lingering impression that we can do better. It would be better not to have to configure the number of virtual parallel sessions; it would be better to support fair outcomes when competing with other concurrent TCP sessions over a range of bandwidths; and it would be better to have a wide range of scaling properties.

There is no shortage of options here for fine-tuning various aspects of TCP to meet some of these preferences, ranging from adaptations applied to the TCP rate control equation to approaches that view the loading onto the network as a power spectrum problem.

HighSpeed TCP

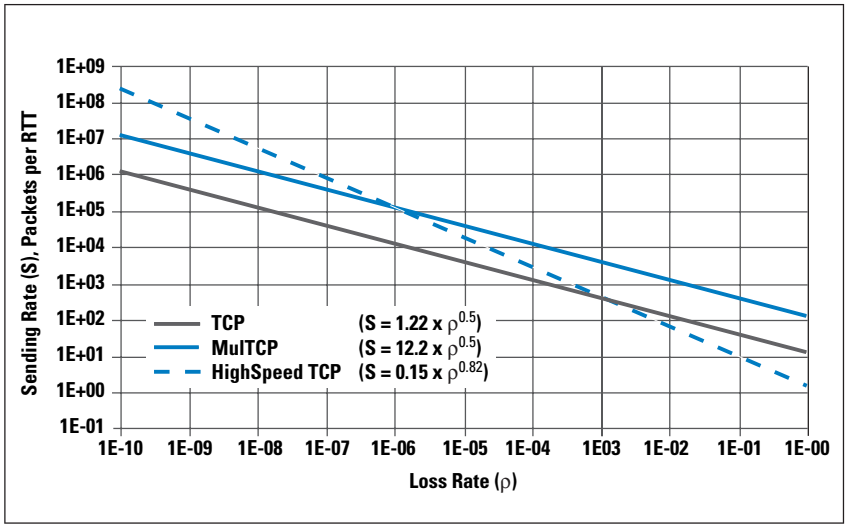
Another approach, described in [2], "HighSpeed TCP for Large Congestion Windows" looks at this from the perspective of the TCP rate equations, developed by Sally Floyd at ICIR.

When TCP operates in congestion avoidance mode at an average speed of W packets per RTT, then the number of packets per RTT varies between $(2/3)W$ and $(4/3)W$. Each cycle takes $(2/3)W$ RTT intervals, and the number of packets per cycle is therefore $(2/3)W^2$ packets. This result implies that the rate can be sustained at W packets per RTT as long as the packet loss rate is 1 packet loss per cycle, or a loss rate, ρ, where $\rho = 1/((2/3)W^2)$. Solving this equation for W gives the average packet rate per RTT of $W = \sqrt{(1.5)}/\sqrt{(\rho)}$. The general rate function for TCP, R, is therefore: $R = (\text{MSS/RTT}) \times (\sqrt{(1.5)}/\sqrt{(\rho)})$, where MSS is the TCP packet size.

Taking this same rate equation approach, what happens for N multiple streams? The ideal answer is that the parallel streams operate N times faster at the same loss rate, or, as a rate equation the number of packets per RTT, W_N, can be expressed as $W_N = N((\sqrt{(1.5)}/\sqrt{(\rho)})$, and each TCP cycle is compressed to an interval of $(2/3)$ (W_N^2/N^2).

But perhaps the desired response is not to shift the TCP rate response by a fixed factor of N—as is the intent with MulTCP—but to adaptively increase the sending rate through increasing values of N as the loss rate falls. The proposition made by HighSpeed TCP is to use a TCP response function that preserves the fixed relationship between the logarithm of the sending rate and the logarithm of the packet loss rate, but alters the slope of the function, such that TCP increases its congestion avoidance increment as the packet loss rate falls. This relationship is shown in Figure 6 where the log of the sending rate is compared to the log of the packet loss rate. MulTCP preserves the same relationship between the log of the sending rate and the log of the packet loss rate, but alters the offset, whereas changing the value of the exponent of the packet loss rate causes a different slope in the rate equation.

Figure 6: TCP Response Functions



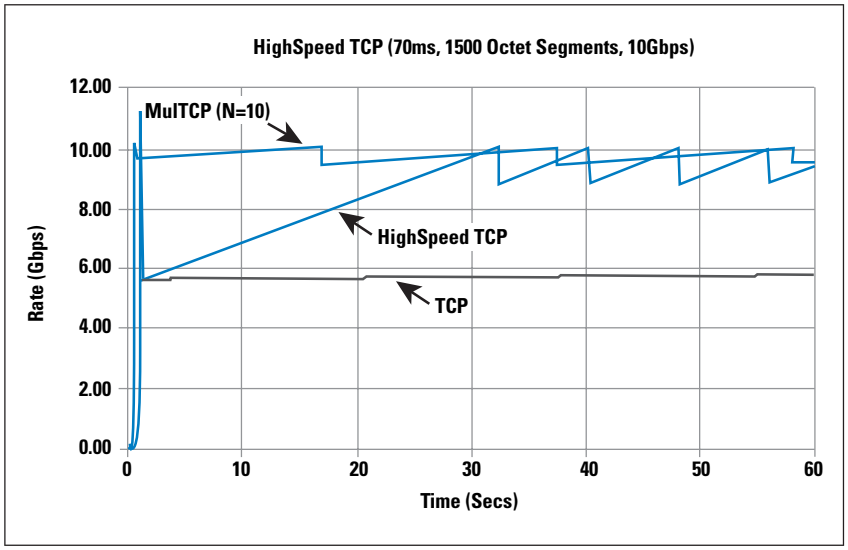
One way to look at the HighSpeed TCP proposal is that it operates in the same fashion as a turbocharger on an engine; the faster the engine is running, the higher the turbo-charged boost to the normal performance of the engine. Below a certain threshold value the TCP congestion avoidance function is unaltered, but when the packet loss rate falls below a certain threshold value then the higher speed congestion avoidance algorithm is invoked. The higher-speed rate equation proposed by HighSpeed TCP is based on achieving a transfer rate of 10 Gbps over a 100-ms latency path with a packet loss rate of 1 in 10 million packets. Working backward from these parameters gives us a rate equation for *W*, the number of packets per RTT interval of $W = 0.12/\rho^{0.835}$, approximately equivalent to a MulTCP session where the number of parallel sessions, *N*, is raised as the TCP rate increases.

This result can be translated into two critical parameters for a modified TCP: the number of segments to be added to the current window size for each lossless RTT time interval, and the number of segments to reduce the window size in response to a packet loss event. Conventional TCP uses values of 1 and (½)*W*, respectively. The HighSpeed TCP approach increases the congestion window by 1 segment for TCP transfer rates up to 10 Mbps, but then uses an increase of some 6 segments per RTT for 100 Mbps, 26 segments at 1 Gbps and 70 segments at 10 Gbps. In other words the faster the TCP rate that has already been achieved, then the greater the rate acceleration. Highspeed TCP also advocates a smaller multiplicative decrease in response to a single packet drop, so that at 10 Mbps the multiplier would be ½, at 100 Mbps the multiplier is 1/3, at 1 Gbps it is 1/5, and at 10 Gbps it is set to 1/10.

What does this process look like? Figure 7 shows a HighSpeed TCP simulation. What is not easy to discern is that during congestion avoidance HighSpeed TCP opens its sending window in increments of 53 through 64 segments each RTT interval, making the rate curve slightly upward during this window expansion phase.

HighSpeed TCP manages to recover from the initial rate halving from slow start in about 30 seconds, and operates at an 8-second cycle, as compared to the 38-minute cycle of a single TCP stream, or a 10-stream MulTCP session that operates at a 21-second cycle.

Figure 7: HighSpeed TCP Simulation



One other aspect of this work concerns the so-called slow start algorithm, which at these speeds is not really slow at all. The final RTT interval in our scenario has TCP attempting to send an additional 50 MB of data in just 70 ms, meaning an additional 33,333 packets are pushed into the network queues. Unless the network path is completely idle at this point, it is likely that hundreds—if not thousands—of these packets will be dropped in this step, pushing TCP back into a restart cycle. HighSpeed TCP has proposed a limited slow start to accompany HighSpeed TCP that limits the inflation of the sending window to a fixed upper rate per RTT to avoid this problem of slow start overwhelming the network and causing the TCP session to continually restart. Other changes for HighSpeed TCP are to extend the limit of three duplicate ACKs before retransmitting to a higher value, and a smoother recovery when a retransmitted packet is itself dropped.

Scalable TCP

Of course HighSpeed TCP is not the only offering in the high-performance TCP stakes.

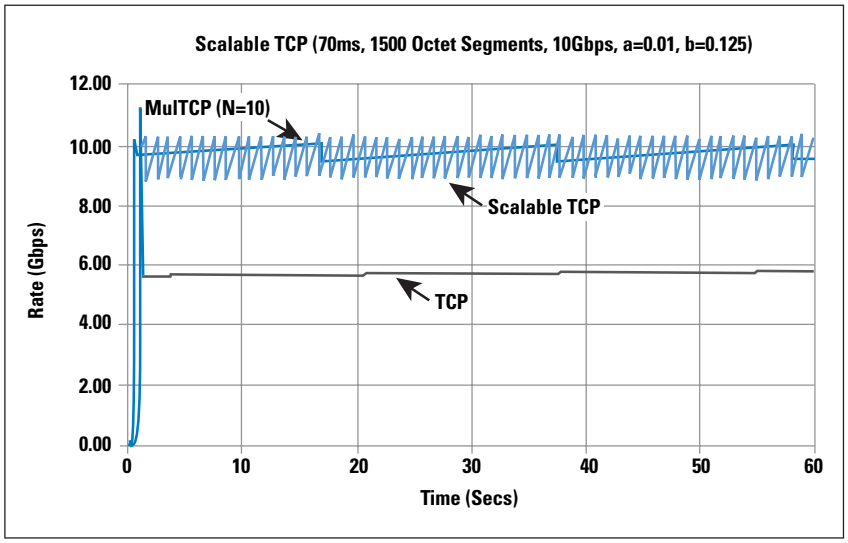
Scalable TCP[3], developed by Tom Kelly at Cambridge University, attempts to break the relationship between TCP window management and the RTT time interval. It does this by noting that in "conventional" TCP, the response to each ACK in congestion avoidance mode is to inflate the sender's congestion window size *(cwnd)* by *(1/cwnd)*, thereby ensuring that the window is inflated by 1 segment each RTT interval. Similarly the window halving on packet loss can be expressed as a reduction in size by *(cwnd/2)*. Scalable TCP replaces the additive function of the window size by the constant value *a*.

The multiplicative decrease is expressed as a fraction *b*, which is applied to the current congestion window size.

In Scalable TCP, for each ACK the congestion window is inflated by the constant value *a*, and upon packet loss the window is reduced by the fraction *b*. The relative performance of Scalable TCP as compared to conventional TCP and MulTCP is shown in Figure 8.

The essential characteristic of Scalable TCP is the use of a multiplicative increase in the congestion window, rather than a linear increase, effectively creating a higher frequency of oscillation of the TCP session, probing upward at a higher rate and more frequently than HighSpeed TCP or MulTCP. The frequency of oscillation of Scalable TCP is independent of the RTT interval, and the frequency can be expressed as $f = log(1 - b) / log(1 + a)$. In this respect, longer networks paths exhibit similar behavior to shorter paths at the bottleneck point. Scalable TCP also has a linear relationship between the log of the packet loss rate and the log of the sending rate, with a greater slope of HighSpeed TCP.

Figure 8: Scalable TCP



BIC and CUBIC

The common concern here is that TCP underperforms in those areas of application where there is a high bandwidth-delay product. The common problem observed here is that the additive window inflation algorithm used by TCP can be very inefficient in long-delay, high-speed environments. As can be seen in Figure 10, the ACK response for TCP is a congestion window inflation operation where the amount of inflation of the window is a function of the current window size and some additional scaling factor.

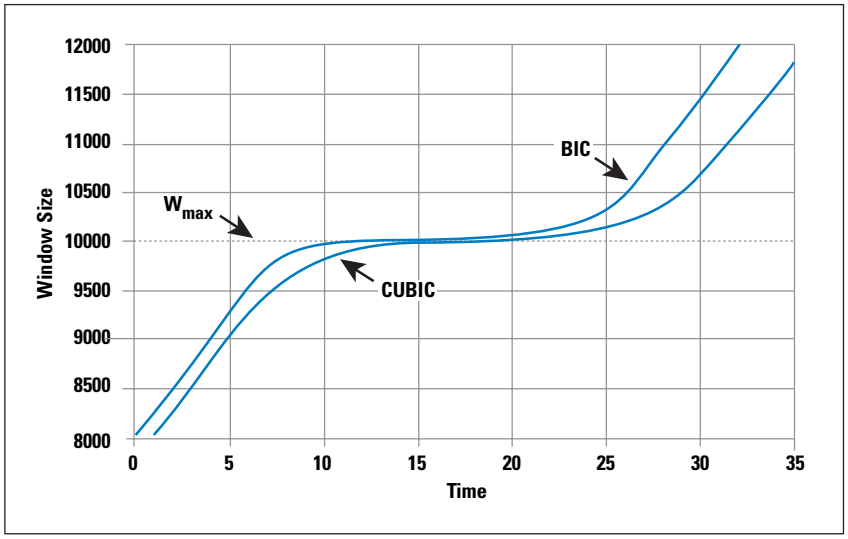
Binary Increase Congestion Control (BIC)[4] takes a different view, by assuming that TCP is actively searching for a packet sending rate that is on the threshold of triggering packet loss, and uses a binary chop search algorithm to achieve this efficiently.

When BIC performs a window reduction in response to packet drop, it remembers the previous maximum window size, as well as the current window setting. With each lossless RTT interval BIC attempts to inflate the congestion window by one half of the difference between the current window size and the previous maximum window size. In this way BIC quickly attempts to recover from the previous window reduction, and, as BIC approaches the old maximum value, it slows down its window inflation rate, halving its rate of window inflation each RTT. This process is not quite as drastic as it may sound, because BIC also uses a maximum inflation constant to limit the amount of rate change in any single RTT interval. The resultant behaviour is a hybrid of a linear and a non-linear response, where the initial window inflation after a window reduction is a linear increase, but as the window approaches the previous point where packet loss occurred the rate of window increase slows down. BIC uses the complementary approach to window inflation when the current window size passes the previous loss point. Initially further window inflation is small, and the size of the window inflation value doubles for each RTT, up to a limit value, beyond which the window inflation is once more linear.

BIC can be too aggressive in low RTT networks and in slower speed situations, leading to a refinement of BIC, namely CUBIC[5]. CUBIC uses a third-order polynomial function to govern the window inflation algorithm, rather than the exponential function used by BIC. The cubic function is a function of the elapsed time since the previous window reduction, rather than the implicit use by BIC of an RTT counter, so that CUBIC can produce fairer outcomes in a situation of multiple flows with different RTTs. CUBIC also limits the window adjustment in any single RTT interval to a maximum value, so the initial window adjustments after a reduction are linear. Here the new window size, W, is calculated as $W = C(t - K)^3 + W_{max}$, where C is a constant scaling factor, t is the elapsed time since the last window reduction event, W_{max} is the size of the window prior to the most recent reduction and K is a calculated value: $K = (W_{max} \beta / C)1/3$. This function is more stable when the window size approaches the previous window size W_{max}. The use of a time interval rather than an RTT counter in the window size adjustment is intended to make CUBIC more sensitive to concurrent TCP sessions, particularly in short RTT environments.

Figure 9 shows the relative adjustments for BIC and CUBIC, using a single time base. The essential difference between the two algorithms is evident in that the CUBIC algorithm attempts to reduce the amount of change in the window size when near the value where packet drop was previously encountered.

Figure 9: Window Adjustment for BIC and CUBIC



Westwood

The "steady state" mode of TCP operation is one that is characterized by the "sawtooth" pattern of rate oscillation. The additive increase is the means of exploring for viable sending rates while not causing transient congestion events by accelerating the sending rate too quickly. The multiplicative decrease is the means by which TCP reacts to a packet loss event that is interpreted as being symptomatic of network congestion along the sending path.

BIC and CUBIC concentrate on the rate increase function, attempting to provide for greater stability for TCP sessions as they converge to a long-term available sending rate. The other perspective is to examine the multiplicative decrease function, to see if there is further information that a TCP session can use to modify this rate decrease function.

The approach taken by Westwood[6], and a subsequent refinement, Westwood+[7], is to concentrate on the halving by TCP of its congestion window in response to packet loss (as signaled by three duplicate ACK packets). The conventional TCP algorithm of halving the congestion window can be refined by the observation that the stream of return ACK packets actually provides an indication of the current bottleneck capacity of the network path, as well as an ongoing refinement of the minimum RTT of the network path. The Westwood algorithm maintains a bandwidth estimate by tracking the TCP acknowledgement value and the inter-arrival time between ACK packets in order to estimate the current network path bottleneck bandwidth. This technique is similar to the "Packet Pair" approach, and that used in the TCP Vegas. In the case of the Westwood approach the bandwidth estimate is based on the receiving ACK rate, and is used to set the congestion window, rather than the TCP send window. The Westwood sender keeps track of the minimum RTT interval, as well as a bandwidth estimate based on the return ACK stream. In response to a packet loss event, Westwood does not halve the congestion window, but instead sets it to the bandwidth estimate times the minimum RTT value.

If the current RTT equals the minimum RTT, implying that there are no queue delays over the entire network path, then the sending rate is set to the bandwidth of the network path. If the current RTT is greater than the minimum RTT, the sending rate is set to a value that is lower than the bandwidth estimate, and allows for additive increase to once again probe for the threshold sending rate when packet loss occurs.

The major concern here is the potential variation in inter-ACK timing, and although Westwood uses every available data and ACK pairing to refine the current bandwidth estimate, the approach also uses a low pass filter to ensure that the bandwidth estimate remains relatively stable over time. The practical result here is that the receiver may be performing some form of ACK distortion, such as a delayed ACK response, and the network path contains jitter components in both the forward and reverse direction, so that ACK sequences can arrive back at the sender with a high variance of inter-ACK arrival times. Westwood+ further refines this technique to account for a false high reading of the bandwidth estimate due to ACK compression, using a minimum measurement interval of the greater of the RTT or 50 ms.

The intention here is to ensure that TCP does not over-correct when it reduces its congestion window, so that the problems relating to the slow inflation rate of the window are less critical for overall TCP performance. The critical part of this work lies in the filtering technique that takes a noisy sequence of measurement samples and applies an anti-aliasing filter followed by a low-pass discrete-time filter to the data stream in order to generate a reasonably accurate available bandwidth estimate. This estimate is coupled with the minimum RTT measurement to provide a lower bound for the TCP congestion window setting following detection of packet loss and subsequent fast retransmit repair of the data stream. If the packet loss is caused by network congestion the new setting will be lower than the threshold bandwidth (lower by the ratio RTT_{min} / $RTT_{current}$), so that the new sending rate will also allow the queued backlog of traffic along the path to clear. If the packet loss is caused by media corruption, the RTT value will be closer to the minimum RTT value, in which case the TCP session-rate backoff is smaller, allowing for a faster recovery of the previous data rate.

Although this approach has direct application in environments where the probability of bit-level corruption is intermittently high, such as often encountered with wireless systems, it also has some application to the long-delay, high-speed TCP environment. The rate backoff of TCP Westwood is one that is based on the RTT_{min} / $RTT_{current}$ ratio, rather than rate halving in conventional TCP, or a constant ratio, such as used in MulTCP, allowing the TCP session to oscillate its sending rate closer to the achievable bandwidth rather than performing a relatively high-impact rate backoff in response to every packet loss event.

H-TCP

The observation made by the proponents of H-TCP[9] is that better TCP outcomes on high-speed networks is achieved by modifying TCP behavior to make the time interval between congestion events smaller. The signal that TCP has taken up its available bandwidth is a congestion event, and by increasing the frequency of these events TCP will track this resource metric with greater accuracy. To achieve this tracking, the H-TCP proponents argue that both the window increase and decrease functions may be altered, but in deciding whether to alter these functions, and in what way, they argue that a critical factor lies in the level of sensitivity to other concurrent network flows, and the ability to converge to stable resource allocations to various concurrent flows.

> "While such modifications might appear straightforward, it has been shown that they often negatively impact the behaviour of networks of TCP flows. High-speed TCP and BIC-TCP can exhibit extremely slow convergence following network disturbances such as the start-up of new flows; Scalable-TCP is a multiplicative-increase multiplicative-decrease strategy and as such it is known that it may fail to converge to fairness in drop-tail networks."
>
> Work-in-progress: **`draft-leith-tcp-htcp-01.txt`**

H-TCP argues for minimal changes to the window control functions, observing that in terms of fairness a flow with a large congestion window should, in absolute terms, reduce the size of their window by a larger amount that smaller-sized flows, as a means of readily establishing a dynamic equilibrium between established TCP flows and new flows entering the same network path.

H-TCP proposes a timer-based response function to window inflation, where for an initial period, the existing value of one segment per RTT is maintained, but after this period the inflation function is a function of the time since the last congestion event, using an order-2 polynomial function where the window increment in each RTT interval, $\alpha = (\frac{1}{2}T2 + 10T + 1)$, where T is the elapsed time since the last packet loss event. This equation is further modified by the current window reduction factor β where $\alpha' = 2 \times (1 - \beta) \times \alpha$.

The window reduction multiplicative factor, β, is based on the variance of the RTT interval , and β is set to RTT_{min} / RTT_{max} for the previous congestion interval, unless the RTT has a variance of more than 20 percent, in which case the value of ½ is used.

H-TCP appears to represent a further step along the evolutionary path for TCP, taking the adaptive window inflation function of HighSpeed TCP, using an elapsed timer as a control parameter as was done in Scalable TCP, and using the RTT ratio as the basis for the moderation of the window reduction value from Westwood.

FAST

FAST[10] is another approach to high-speed TCP. FAST is probably best viewed in context in terms of the per packet response of the various high speed TCP approaches, as indicated in the following Control and Response table:

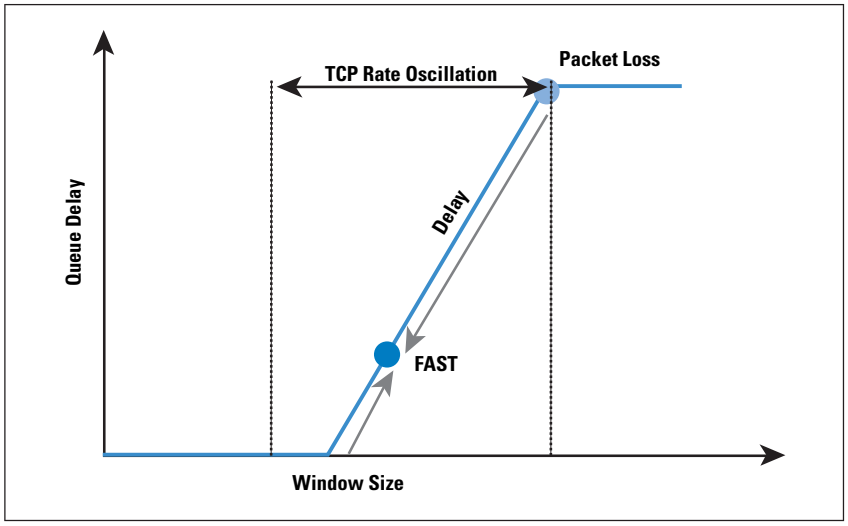
Type	Control Method	Trigger	Response
TCP	AIMD(1,0.5)	ACK response Loss response	W = W + 1/W W = W – W × 0.5
MulTCP	AIMD(N,1/2N)	ACK response Loss response	W = W + N/W W = W – W × 1/2N
HighSpeed TCP	AIMD(a(w), b(w))	ACK response Loss response	W = W + a(W)/W W = W – W × b(W)
Scalable TCP	MIMD(1/100, 1/8)	ACK response Loss response	W = W + 1/100 W = W – W × 1/8
FAST	RTT Variation	RTT	W = W × (base RTT/RTT) + α

All these approaches share a common structure of window adjustment, where the sender's window is adjusted according to a control function and a flow gain. TCP, MulTCP, HighSpeed TCP, Scalable TCP, BIC, CUBIC, Westwood, and H-TCP all operate according to a congestion measure that is based on ACK clocking and a packet loss trigger. What is happening in these models is that a bottleneck point on the network path has reached a level of saturation such that the bottleneck queue is full and packet loss is occurring. It is noted that the build up of the queue prior to packet loss would have caused a deterioration of the RTT.

This fact leads to the observation made by FAST, that another form of congestion signalling is one that is based on RTT variance, or cumulative queuing delay variance. FAST is based on this latter form of congestion signalling.

FAST attempts to stabilize the packet flow at a rate that also stabilizes queue delay, by basing its window adjustment, and therefore its sending rate, such that the RTT interval is stabilized. The window response function is based on adjusting the window size by the proportionate amount that the current RTT varies from the average RTT measurement. If the current RTT is lower than the average, then window size is increased, and if the current RTT is higher then window size is decreased. The amount of window adjustment is based on the proportionate difference between the two values, leading to the observation that FAST exponentially converges to a base RTT flow state. By comparison, conventional TCP has no converged state, but instead oscillates between the rate at which packet loss occurs and some lower rate (Figure 10).

Figure 10: TCP Response Function vs. FAST



FAST maintains an exponential weighted average RTT measurement and adjusts its window in proportion to the amount by which the current RTT measurement differs from the weighted average RTT measurement. It is harder to provide a graph of a simulation of FAST as compared to the other TCP methods, and the more instructive material has been gathered from various experiments using FAST.

XCP — End-to-End and Network Signalling

It is possible to also call in the assistance of the routers on the path and call on them to mark packets with signaling information relating to current congestion levels. This approach was first explored with the concept of ECN, or *Explicit Congestion Notification*, and has been generalized into a transport flow control protocol, called XCP,[11] where feedback relating to network load is based on explicit signals provided by routers relating to their relative sustainable load levels. Interestingly this digresses from the original design approach of TCP, where the TCP signaling is set up as effectively a heartbeat signal being exchanged by the end systems, and the TCP flow control process is based upon interpretation of the distortions of this heartbeat signal by the network.

XCP appears to be leading into a design approach where the network switching elements play an active role in end-to-end flow control, by effectively signalling to the end systems the current available capacity along the network path. This setup allows the end systems to respond rapidly to available capacity by increasing the packet rate to the point where the routers along the path signal that no further capacity is available, or to back off the sending rate when the routers along the path signal transient congestion conditions.

Whether such an approach of using explicit router-to-end host signals leads to more efficient very high-speed transport protocols remains to be determined, however.

Where Next?

The basic question here is whether we have reached some form of fundamental limitation of the TCP window-based congestion control protocol, or whether it is a case that the window-based control system remains robust at these speeds and distances, but that the manner of control signalling will evolve to adapt to an ever-widening range of speed extremes in this environment.

Rate-based pacing, as used in FAST can certainly help with the problem of the problem of guessing what are "safe" window inflation and reduction increments, and it is an open question as to whether it is even necessary to use a window inflation and deflation algorithm or whether it would be more effective to head in other directions, such as rate control, RTT stability control or adding additional network-generated information into the high-speed control loop. Explicit router-based signaling, such as described in XCP, allows for quite precise controls over the TCP session, although what is lost there is the adaptive ability to deploy the control system over any existing IP network.

However, across all these approaches, the basic TCP objectives remain the same: what we want is a transport protocol that can use the available network capacity as efficiently as possible—and as quickly as possible—minimizing the number of retransmissions and maximizing the effective data throughput.

We also want a protocol that can adapt to other users of the network, and attempt to fairly balance its use with competing claims for network resources.

The various approaches that have been studied to date all represent engineering compromises in one form or another. In attempting to optimize the instantaneous transfer rate the congestion control algorithm may not be responsive to other concurrent transport sessions along the same path. Or in attempting to optimize fairness with other concurrent sessions, the control algorithm may be unresponsive to available network path capacity. The control algorithm may be very unresponsive to dynamic changes in the RTT that may occur during the session because of routing changes in the network path. Which particular metrics of TCP performance are critical in a heterogeneous networking environment is a topic where we have yet to see a clear consensus emerging from the various research efforts.

However, we have learned a few things about TCP that form part of this consideration of where to take TCP in this very-high-speed world:

- The first lesson is that TCP has been so effective in terms of overall network efficiency and mutual fairness because everyone uses much the same form of TCP, with very similar response characteristics. If we all elected to use radically different control functions in each of our TCP implementations then it appears likely that we would have a poorly performing chaotic network subject to extended conditions of complete overload and inefficient network use.

- The second lesson is that a transport protocol does not need to solve media level or application problems. The most general form of transport protocol should not rely on characteristics of specific media, but should use specific responses from the lower layers of the protocol stack in order to function correctly as a transport system.
- The third lesson from TCP is that a transport protocol can become remarkably persistent and be used in contexts that were simply not considered in the original protocol design, so any design should be careful to allow generous margins of use conditions.
- The final lesson is one of fair robustness under competition. Does the protocol negotiate a fair share of the underlying network resource in the face of competing resource claims from concurrent transport flows?

Of all these lessons, the first appears to be the most valuable and probably the most difficult to put into practice. The Internet works as well as it does today largely because we all use the much same transport control protocol. If we want to consider some changes to this control protocol to support higher-speed flows over extended latency, then it would be perhaps reasonable to see if there is a single control structure and a single protocol that we can all use.

So deciding on a single approach for high-speed flows in the high-speed Internet is perhaps the most critical part of this entire agenda of activity. It is one thing to have a collection of differently controlled packet flows each operating at megabits-per-second flow rates on a multi-gigabit network, but it is quite a frightening prospect to have all kinds of different forms of flows each operating at gigabits per second on the same multigigabit network. If we cannot make some progress in reaching a common view of a single high-speed TCP control algorithm then it may indeed be the case that none of these approaches will operate efficiently in a highly diverse high-speed network environment.

Acknowledgment

I must acknowledge the patient efforts of Larry Dunn in reading through numerous iterations of this article, correcting the text and questioning some of my wilder assertions. Thanks Larry.

However, whatever errors may remain are, undoubtedly, all mine.

Further Reading

There is a wealth of reading on this topic, and here any decent search engine can assist. However if you are interested in this topic and want a starting reference that describes it in a very careful and structured manner, then I can recommend the following two sources as a good way to start exploring this topic to gain an overview of the current state of the art in this area:

- "HighSpeed TCP for Large Congestion Windows," S. Floyd, RFC 3649, December 2003.

  Floyd's treatment of this topic is precise, encompassing, and wonderfully presented. If only all RFCs were of this quality.

- Proceedings of the Workshops on Protocols for Fast Long-Distance Networks.

  These workshops have been held in:

  2003: **http://datatag.web.cern.ch/datatag/pfldnet2003/**

  2004: **http://www-didc.lbl.gov/PFLDnet2004/program.htm**

  2005: **http://www.ens-lyon.fr/LIP/RESO/pfldnet2005/**

References

[1] "Differentiated End-to-End Internet Services Using a Weighted Proportional Fair Sharing TCP," J. Crowroft and P. Oechcslin, ACM SIGCOMM *Computer Communication Review,* Volume 28, No. 3, pp. 53–69, July 1998.

[2] "HighSpeed TCP for Large Congestion Windows," S. Floyd, RFC 3649, December 2003.

[3] "Scalable TCP: Improving Performance in High-Speed Wide Area Networks," T. Kelly, ACM SIGCOMM *Computer Communication Review,* Volume 33, No. 2, pp. 83–91, April 2003.

[4] "Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks," L. Xu, K. Harfoush, and I. Rhee, *Proceedings of IEEE INFOCOMM 2004,* March 2004.

[5] "CUBIC: A New TCP-Friendly High-Speed TCP Variant," I. Rhee, L. Xu, **http://www.csc.ncsu.edu/faculty/rhee/export/bitcp/cubic-paper.pdf**, February 2005.

[6] "TCP Westwood: Congestion Window Control Using Bandwidth Estimation," M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, and S. Mascolo, *Proceedings of IEEE Globecom 2001,* Volume 3, pp. 1698–1702, November 2001.

[7] "Linux 2.4 Implementation of Westwood+ TCP with Rate-Halving: A Performance Evaluation over the Internet," A. Dell'Aera, L. A. Greco, and S. Mascolo, Tech. Rep. No. 08/03/S, Politecnico di Bari, **http://deecal03.poliba.it/mascolo/tcp%20westwood/Tech_Rep_08_03_S.pdf**

[8] "End-to-end Internet packet dynamics," V. Paxson, *Proceedings of ACM SIGCOMM 97,* pp. 139–152, 1997.

[9] "H-TCP: TCP Congestion Control for High Bandwidth-Delay Product Paths," D. Leith, R. Shorten, Work in Progress, June 2005. Internet Draft: **draft-leith-tcp-htcp-00.txt**

[10] "FAST TCP: Motivation, Architecture, Algorithms, Performance," C. Jin, X. Wei, and S. H. Low, *Proceedings of IEEE INFOCOM 2004,* March 2004.

[11] "Congestion Control for High Bandwidth-Delay Product Networks," D. Katabi, M. Handley, and C. Rohrs, ACM SIGCOMM *Computer Communication Review,* Volume 32, No. 4, pp. 89–102, October 2002.

[12] "TCP Performance," Geoff Huston, *The Internet Protocol Journal*, Volume 3, No. 2, June 2000.

[13] "The Future for TCP," Geoff Huston, *The Internet Protocol Journal*, Volume 3, No. 3, September 2000.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for almost two decades, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector, and has served time with Telstra, where he was the Chief Scientist in the company's Internet area. Geoff is currently the Internet Research Scientist at the Asia Pacific Network Information Centre (APNIC). He has been a member of the Internet Architecture Board, and currently co-chairs three Working Groups in the IETF. He is author of several Internet-related books. E-mail: **gih@apnic.net**

How Instant Messaging Is Transforming the Enterprise Network

by David Strom

Instant Messaging (IM) has come of age and is close to becoming one of those protocols that offers something for everyone. Once the province of chatty teens looking to replace phone conversations with electronic ones, IM is now a corporate mainstay and part of a new breed of applications that are built around "presence detection," the ability to determine when someone—or something—is online and available to communicate.

Indeed, IM is rapidly spreading across the corporate world and becoming an able replacement for overflowing voicemail and e-mail inboxes that are clogged with spam and buried in irrelevant and non-time-sensitive postings. If you must get through to a busy corporate executive, IM is becoming the fastest and most effective method of communicating. Move over CrackBerry.

IM offers several benefits today, having taken some lessons learned by other Internet protocols of the past. First, it has a solid user and developer base. Second, it has a relatively simple building-block structure like the best of Internet protocols, with well-defined clients and servers. Third, interoperability efforts are beginning to pay off among the leading independent and private IM systems. Fourth, open-source rules are making inroads in all the right places. Fifth, Microsoft is a friend (for once) of IM and helping matters—rather than playing its usual monopolist role in this space, the company is actually encouraging future developments and interoperability. Finally, a new collection of advanced applications is taking hold that will take advantage of the existing Internet and IM infrastructure and create some very sophisticated IM applications.

Let's examine more closely where IM originated, where it is going, and what the specific implications are for each of these developments and for networking professionals. As a warning, this article by its very nature takes some positions on products and vendors. These opinions are solely those of the author, and they represent nothing wider or more inclusive.

User Base

The IM servers are operated by either public network or private entities. The major difference between the two is that the public systems operate across the Internet and can be accessed by any users who download the appropriate client software and create their own identity. Message traffic is usually transmitted in plaintext and without any encryption whatsoever.

The private IM systems are usually maintained by a corporate IT department and operate behind firewalls; they offer message encryption, message retention, and archiving; prepopulated buddy lists that are integrated into the corporate authentication and directory servers; and better security and privacy that are specific to a particular set of corporate users. These private systems are not available to the public and are designed strictly for employee communications or communications among particular trading partners of the corporation

The four most popular public IM systems are currently all in corporate hands: Microsoft, Yahoo, eBay/Skype, and AOL. Actually, we should make that five systems because AOL owns two separate networks, *AOL Instant Messenger* (AIM) and *I seek you* (ICQ). Introduced in November 1996, ICQ was actually the first general-purpose IM system combining presence or a list of contacts with the ability to send messages. Other popular systems include the open-source Jabber and Tencent QQ, the latter very popular in China. Estimates vary widely as to the total number of nonduplicated users—because many people have multiple accounts and use multiple systems—but it is safe to say that more than 150 million users are active across all these systems at any moment. The most recent estimates of active users are as follows:[1]

IM System	Estimate of Active Users
AIM	53 million active users
ICQ	15 million active users
Skype	10 million active users
MSN Messenger	29 million active users
Yahoo Messenger	21 million active users
Jabber	13.5 million enterprise users
Tencent QQ	10 million active users

Why IM Is So Popular for Businesses

But these numbers are more about individuals using IM. They hide the real story over the past several years, the rise of IM as a solid enterprise communications tool. Corporate IM usage has skyrocked the last several years, and one survey has found IM users in more than 50 percent of American corporations[2]. As mentioned earlier, there are public and private IM systems. The vast majority of the private IM systems are for institutional use for communications inside a company or among several suppliers, customers, and other trading partners.

The largest players in the private IM space are Microsoft Office Live Communications Server and IBM/Lotus' Sametime, although Jabber Corporation (not to be confused with the Jabber Software Foundation) is also gaining a strong following. We will discuss more about the role of open source in a moment, but first let's examine the reasons why IM has become so popular among so many business users.

First, workers have become more mobile and more difficult to track down. As secretarial support disappears and voicemail becomes more the norm, you want to know when people are actually at their desk—or laptop—these days. Staffs are more far-flung, and the global village becomes a lot smaller when you use IM to "talk" to someone halfway across the planet and get an immediate response. Finding someone who is available requires more than just making a phone call or exchanging e-mail messages. IM automatically tells you who is available—and who is not—at any given hour of the day.

Second, e-mail is no longer the productivity tool it once was because pipes are clogged with spam, viruses, and phishing attacks. Getting a quick response—that is, within minutes—through e-mail now seems so quaint, so "last year."

Third, IM enables better collaboration and a tighter sense of community. With IM, you can educate an entire team, give the team feedback in real time, develop relationships, and cement the team together. It is a nice antidote and countermeasure to connect all these home-based and remote workers.

Fourth, the next generation of IM is not just about text chats; it also offers solid integration with voice and video. Voice and video calling is now part of Microsoft, Yahoo, Apple, and AOL IM software as well as part of the Skype network, which pioneered the feature. These audio and video extensions are becoming more popular with the private Lotus and Microsoft systems as well.

Finally, the real-time features of IM and its ability to track someone down no matter where they are located are attractive to customers, partners, and suppliers that need a guaranteed method of communication. IM is becoming the critical technology ingredient for corporations that are looking for faster response times, tying their customers closer together, and enabling teleworkers to communicate across the globe.

Components

Following are some definitions and explanations for those unfamiliar with the world of IM. Every IM network is composed of clients, servers, and protocols to connect them.

Each IM client has three major pieces:

A buddy list or roster of friends with whom you wish to communicate—The list is organized by groups that you specify, such as "friends," "work colleagues," "family," and so forth. The list indicates who is online, who is available to talk to, and who is offline or blocked by the user from communicating. Users organize their buddies in different ways and have complete control over the categories, naming conventions, and the like.

A separate window that shows the text chats in process—Users type in this window and view the responses of their correspondents.

Any additional features for video and audio chats and for file transfers between users

The last item bears some further discussion. All major IM products are moving beyond their roots of simple text chats toward more integrated and sophisticated communications, including real-time voice and video calls. Indeed, the mixture of *Voice over IP* (VoIP) and IM is a potent and popular one, accounting for the rapid uptake in Skype's adoption around the world. To use Skype as an example (although Yahoo has begun offering similar phone calling features in its IM client, and the others are soon to follow), users can make phone calls to the land-line phone numbers for a few pennies per minute—even calls to numbers in other countries. This is part of its attraction, along with voice mailboxes that are attached to a particular IM username.

The IM server maintains the directory of user accounts and keeps track of who is online, and in most cases routes messages among users. The major difference between an IM server and a *Simple Mail Transfer Protocol* (SMTP) e-mail server is that the IM server operates in real time, sending messages back and forth between two users as they finish typing a line of text. The servers also pass information in real time as to the availability of various users in the directory, when they come online and change their "status" message.

Users can typically set their availability in one of many different modes:

Online and ready to receive messages

Away from the computer, in which case correspondents receive a message saying so (or whatever the user wishes to be displayed)

Unavailable or offline

Blocked from anyone's view for privacy reasons

This status message can be changed at the user's discretion and is one of the main attractions for teens and other hypercommunicators. You can actually track what people are doing (or at least, saying that they are doing), by monitoring their status messages. (I am at the beach, I am taking a nap, I am at lunch, I am having coffee, and so forth.) For my teenaged daughter, this is one way she documents her life and one way that her friends can keep track of her—having a cell phone is not enough! There are numerous third-party add-ins to enhance your away message with clever graphics, hyperlinks to various Websites, and other effluvia as well.

The combination of instant access and persistent status indicator is at the core of why IM is such a powerful application. In a single window on your computer, you have a list of all your correspondents and can quickly determine who is online and who is not.

The blocking ability for some systems works universally, meaning that your presence is cloaked for everyone, as well as for specific users that you do not wish to communicate with or know your particular status, such as ex-spouses or ex-colleagues.

In most IM networks, you can be signed on from only one computer at any given moment. If you attempt to sign on from a second machine, you get an error message or your first computer is automatically logged out of the system. This is one way for the network to keep track of where you are located, because you can be in only one place at any given time.

Each server uses the TCP/IP Internet infrastructure and communicates with its clients over an assigned port number across the Internet. These ports can be blocked or proxied to different numbers, depending on the network administrator's policies toward IM traffic. Typical port numbers follow:

IM System	Port Numbers
ICQ	4000
AIM	5190–3
XMPP	5222–3
MSNP (Microsoft)	1863
YMSG (Yahoo)	5050
Skype	80, 443, and others

Notice an interesting thing about Skype's protocol: there is no single assigned port number. Users can set one of the ports in its configuration settings, but Skype uses a series of ports to communicate.[3] This setup suggests several concerns, which we address next.

The Dark Side

Although these are all compelling reasons for the rise of IM across the corporate network, all is not constructive with IM. This section discusses problems specifically germane to Skype and problems with all IM products in general.

When the Skype client is installed on a computer, it picks a random port to communicate with other Skype computers, using what is believed to be a form of *Request for Comments* (RFC) 3489[4]. This process is similar to many network-based games and peer-to-peer file-sharing products—no surprise because the developers of Skype worked on the Kazaa music file-sharing software. Because of its programming model, Skype is adept at traversing *Network Address Translation* (NAT) routers and can usually find a communications path to the outside world. Skype also encrypts all its message traffic, and this fact coupled with random port usage and its peer-to-peer programming model makes it look very similar to some malicious code that is unleashed across your network.

This is part of its charm and its challenge: network administrators who want to block Skype usage usually have a very difficult time figuring out how to do so[5], and may have to resort to third-party blocking products or clever configurations. One of the papers listed in [3] shows a way to block Skype using the popular open-source Squid caching proxy: not only do you have to prevent outbound *User Datagram Protocol* (UDP) connections over port 443, but you also must prevent connections to numeric IP addresses.

Although Skype has its own problems because of the way it is designed, there are several significant drawbacks to widespread adoption and deployment of any IM application. IM is not immune to infections, and just as its popularity is on the increase, so are ways to send malicious payloads and attacks too. What makes matters worse with IM versus say, e-mail, is its very instant nature: an infection can easily spread across a network in a matter of seconds, given that users are logged in, have long lists of users, and tend to think that any message coming from their respondents is more trusted than the average e-mail. In addition, Internet chat has long been a mechanism for controlling large-scale bot-nets of zombie computers, whose owners are unaware of such usage. Numerous virus authors have used exploits in Internet Relay Chat, for example, to control their villains across the Internet.

To avoid these problems, many corporations have either designed their own or are using one of several commercial IM protection products to screen incoming messages for particular patterns and methods of attack. The IM protection products work just like antivirus products work with e-mail messages: they download pattern files on a regular basis from a central server, and perform deep packet inspection across a perimeter to determine what is malicious and what is not.

Interoperability

Each public IM system is an island unto itself: users on one cannot easily communicate with users of another, unless one of two things happens:

A user runs one of the multisystem client programs that allows them to sign in to multiple systems concurrently. Still, using these types of products means that just the user can communicate with his or her "buddies" across systems. Many mostly free products that enable this are available[6].

A private IM operator can combine more than one protocol inside the IM server application. This approach means that clients need not know or care about other IM protocols, such as using Microsoft's Live Communications Server 2005[7].

But variables are changing on the interoperability scene to make life better for IM users. First, efforts are under way among the major operators to form better relationships with each other:

In October 2005, Yahoo and Microsoft announced plans to introduce interoperability between MSN and Yahoo Messenger by mid-2006, using *Session Initiation Protocols* (SIPs). In December 2005, AOL and Google announced a strategic partnership deal where Google Talk users can talk with AIM and ICQ users provided they have an identity at AOL.

Second, both Microsoft and Apple have made efforts to include multi-protocol IM clients as part of their desktop operating systems. Apple's iChat in its latest Mac OS 10.4 Tiger, as an example, now supports AIM, Google Talk, and Jabber. Microsoft has announced plans to support other networks in its next release of Windows Vista, expected later this year.

Finally, the private IM systems of Microsoft and Lotus both support multiple IM protocols, and are widening their support for others, making them more useful for corporations.

Still, with all this activity, the IM interoperability scene is pretty poor: think where e-mail was in the early 1990s with custom-crafted gateways and the like so that an MCIMail user could send messages to a CompuServe user.

Setting up two systems to talk to each other is neither simple nor obvious, and each pair of systems must be done separately. So to add Google Talk to Trillian, a user would need to provide the server host name (`talk.google.com`) and port number (5222). (By the way, GoogleTalk has the most helpful instructions on how to set up a variety of third-party applications to connect to its servers.)

But that is not all—even if a user follows these instructions to set up cross-system connections, most systems can exchange only plaintext messages. Video and voice chats between disparate systems are not generally supported, although Apple's iChat has done the best job so far in this arena. And even if users take the multiple-client approach, the structure of their buddy lists is not always maintained and sometimes is presented in a single group of buddies, rather than separated into the groups that were specified when initially setting up the IM account.

The other concern for cross-systems interoperability is a lack of support for privacy or online status. All of the IM systems have the ability to create blacklists, or lists of users that cannot view your online status. These blacklists are not necessarily preserved when running the multiple client systems.

The Rise of Open Source

There is hope on the interoperability scene, however, and that hope is spelled *open source*. The Jabber group of programmers is growing, and the community is aggressively establishing a more pluralistic IM society. These steps revolve around software using the protocol called the *Extensible Messaging and Presence Protocol* (XMPP), the IETF's formalization of the core protocols created by the Jabber open-source community in 1999, and contained in four RFCs[8, 9, 10, and 11].

Jeremie Miller developed the original Jabber server in 1998. Now the project has reached critical mass. Notable is the wide number of different server and client formulations that support XMPP. Jabber.com sells a commercial license, along with a combination of *General Public License* (GPL)-based licensed servers and other commercial versions. The project has supported the efforts of dozens of client implementations[12]. Last year, support reached a new milestone with Google Talk and more recently the Gizmo Project using these protocols.

Numerous efforts are under way with these clients to extend basic IM functions into new areas, including providing more sophisticated and secure communications, the ability to have multiple identities presented (`david@strom.com` for work colleagues, `dstrom@gmail.com` for personal communications) from the same IM client, and support for more interoperable communications between Jabber and private IM systems.

At the heart of XMPP is the *Extensible Markup Language* (XML) constructs and basic protocols. The core "transport" layer for XMPP is an XML streaming protocol that makes it possible to exchange fragments of XML between any two network endpoints. Authentication and channel encryption happen at the XML streaming layer using other IETF-standard protocols for *Simple Authentication and Security Layer*[13] and *Transport Layer Security*[14].

Servers can connect to each other for interdomain communications, using the form of address for each user as `<user@domain>`—similar to SMTP e-mail, and in many cases, the IM address is the same as one's Internet e-mail address to simplify things.

What is notable about using XMPP is that RFC 3921 also makes it possible to separate the messaging and presence functions if desired (although most deployments offer both). This feature is helpful when building applications-to-applications messaging that does not involve users typing text messages to each other, such as a server sending a network operator an alert when it detects a problem.

The Jabber Software Foundation develops extensions to XMPP through a standards process centered on *Jabber Enhancement Proposals* (JEPs), similar to the RFC process[15]. Currently, more than 30 active proposals have been developed, extending IM into bookmarks, delayed messaging, and other areas.

What Microsoft Is Doing

Microsoft is heavily involved in the IM scene in three important areas. The company operates one of the larger public IM networks, it includes an IM client as part of its Windows operating system, and it sells a private IM server that has some powerful interoperability features called *Live Communications Server* (LCS). What does this mean for the IM community? All good things. Microsoft's MSN and Skype are the more popular IM services outside of North America, and having Skype now a part of eBay is making Microsoft add competitive features such as voice and video chats to its public IM service. Microsoft has actually led the way on IM interoperability with LCS, a fact that can only motivate its competitors to include more pluralist IM offerings of their own. Finally, building in more support for IM in future versions of Windows will help popularize these applications even further.

It was not always this way. Earlier versions of Windows included something called Windows Messenger that was woefully underfeatured and had many bugs. But like so many early Microsoft efforts, technology has improved over time, and now the built-in software that comes with Windows is actually quite competitive with the public IM clients from AOL, Yahoo, and Skype.

Certainly, having Microsoft on one side and open-source efforts on the other is a nice way to encourage development and innovation in the IM arena, and we should expect more here in the future.

Building IM Applications

For most of this article we have addressed the one-to-one aspect of IM. However, IM is evolving into a much more important role, and that is one-to-many communications, and communications between applications instead of actual people. Many vendors have begun selling products in this space, and it is more interesting for several reasons:

First, IM is replacing other means for applications communications. It used to be the case that many network management applications used the *Simple Network Management Protocol* (SNMP) or SMTP protocols to send out their alerts. Now, many applications are using IM messages and taking advantage of the real-time nature of the protocol.

Second, the origins of IM go back to group chat sessions, so group collaboration tools make sense for new IM applications.

Third, even the closed public IM vendors have begun to open their programming interfaces, making it is easier for corporations to build new and sophisticated applications that are presence-aware, in some cases between two computer programs to communicate their status. AOL this year is one such example of opening its IM *application programming interface* (API) kimono, and of course Jabber has always been an open system that has helped lead more of these innovations.

One illustration is with the automotive giant Reynolds and Reynolds, which is using Jabber servers to monitor its own software status at the numerous automotive dealerships around the world. The IT department at Reynolds can quickly see if the company's software is down and take steps to get it working again.

Accredited Home Lenders is using IM to provide its loan brokers a secure and reliable means of communicating in real time with loan specialists to resolve problems with loan applications. And Ecreation built a virtual disk jockey for a Dutch radio station that also broadcasts over the Internet, allowing the station to take requests from listeners around the world through Microsoft's IM network.

Even traders have embraced IM. NetEnergy has been using IM for the past three years, and now negotiates trades between buyers and sellers of oil contracts using IM, decreasing errors and enabling faster communications.

Finally, IM figures prominently helping deaf and hard-of-hearing users communicate. In the era before IM, deaf users required a telephone relay operator to type the message to them and speak to the hearing callers. Go America has built a gateway to IM for its `i711.com` Website, so that deaf users can send messages directly to the operator.

Summary

We have tried to paint a comprehensive a picture of what IM is and where it is going. Certainly, the amount of messaging traffic using the various IM protocols is impressive, and will continue to grow as these new applications are created and as more people discover the advantages of using IM. In several instances IM has replaced voicemail for most internal communications, particularly at high-tech companies and places where real-time communications is important. Although IM is not without its problems, there are ways to protect networks from infection and abuse.

For Further Reading

[1] Nielsen//NetRatings, August 2005 study.

[2] Osterman Research survey:
`http://www.ostermanresearch.com/results/surveyresults_0905.htm`

[3] More details about the underlying Skype protocols, mechanisms for blocking its use, and other helpful tips and tricks for network administrators can be found at this page maintained by Salman A. Baset:
`http://www1.cs.columbia.edu/~salman/skype/index.html`

[4] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy, "STUN—Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)," RFC 3489, March 2003.

[5] A dissection of the Skype protocol along with suggestions about how to block its use can be found in this paper by P. Biondi and F. Desclaux: "Silver Needle in the Skype."
http://www.blackhat.com/presentations/bh-europe-06/bh-eu-06-biondi/bh-eu-06-biondi-up.pdf

[6] Adium and iChat for the Mac, Gaim for Windows and Linux, Trillian Pro for Windows, WebMessenger for Windows Mobile/ Palm, and others.

[7] Microsoft's Live Communications Server 2005 includes its Public IM connector for an additional charge. Lotus' Sametime has had AIM connectivity for several years, and will support other IM networks later this year.

[8] P. Saint-Andre, ed., "Extensible Messaging and Presence Protocol (XMPP): Core," RFC 3920, October 2004.

[9] P. Saint-Andre, ed., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence," RFC 3921, October 2004.

[10] P. Saint-Andre, "Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM)," RFC 3922, October 2004.

[11] P. Saint-Andre, "End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP)," RFC 3923, October 2004.

[12] A list of software clients that support Jabber protocols can be found at:
http://www.jabber.org/software/clients.shtml

[13] J. Myers, "Simple Authentication and Security Layer (SASL)," RFC 2222, October 1997.

[14] T. Dierks and C. Allen, "The TLS Protocol Version 1.0," RFC 2246, January 1999.

[15] Jabber Enhancement proposals are listed at:
http://www.jabber.org/jeps/

DAVID STROM has been writing about Internet protocols and applications for nearly 20 years. Founding editor-in-chief for *Network Computing* magazine, he was most recently the editor-in-chief for **tomshardware.com** and related Websites. Strom has written two books on Internet e-mail (with the doyenne of POP, Marshall T. Rose) and home networking and thousands of magazine articles for most of the leading trade magazines in the IT, computing, and networking fields. He can be reached by e-mail at **david@strom.com**, or by IM: **davidstrom** (AIM and Skype) or **dstrom** (Yahoo, Google Talk, and MSN).

Letters to the Editor

Dear Editor,

In Russ White's "Working with IP Addresses" article (IPJ Volume 9, Number 1), he presents an example subnetting problem ("The Hardest Subnetting Problem") together with a worked solution. While useful as a reinforcement exercise for the rest of the article, care should be exercised before using the steps in the solution "as-is" in a real-world network configuration.

The main problem is that by packing subnets tightly together as shown, growth is restricted in order to guarantee that no address space is wasted. Worse, growth of host numbers on all but the smallest subnet requires renumbering of the subnet or all the smaller subnets allocated after it.

For example, the /26 subnet with 58 hosts will not accommodate more than another four hosts, less than 10-percent growth, without being renumbered.

Since renumbering a network is a nontrivial task even with the tools at our disposal, it is desirable to make it as infrequent as possible.[1]

Allowing for growth will likely but not necessarily waste some address space, but it is preferable to frequent renumbering. It turns out that this example has alternative arrangements of subnets that would permit growth of some subnets without the need to renumber and would lessen the amount of renumbering when it is required.

Using realistic estimates of future hosts rather than current numbers isa simple measure to decrease the frequency of renumbering required. This would also make it obvious that the entire allocation is close to exhaustion and can be exhausted by the need to accommodate as little as six hosts on two subnets that are near full capacity.

Constraints on the supply of IPv4 address space limits how much growth can be accommodated and requires taking a shorter-term rather than longer-term view of growth. For private RFC 1918[2] IP allocations (such as the one used in the example), this applies in only very large organisations, allowing a long-term view to be accommodated.

Unfortunately, the future is hard to predict with any degree of accuracy. In most cases needs for subnet allocation become gradually known over time rather than all at once. The consequences of incorrect estimation can be minimised by using an allocation scheme that allows for as much growth as possible in existing subnets while leaving as much room as possible for future allocations.

This scenario can be achieved by distributing the subnets evenly, weighted by size, across the available address space. The larger the subnet, the more room that needs to be left between it and other large networks. This is particularly important for subnets that are near to capacity. At least the sum of the sizes of neighbouring networks should be allowed. Space close to a network should be reserved for it to grow into, and the remaining space between can be allocated to smaller networks in a recursive fashion. Any allocations in the areas of likely growth should be reclaimable, and preferably these networks should be sparsely populated in order to limit the impact of renumbering on these networks. Working with a diagram of the address space, for example, a linear graph or a binary tree of the address space is a helpful aid.

A more systematic way of distributing the subnets evenly is to use *mirror-image* (MI) counting for allocating subnet numbers. This process is described in RFC 1219[3], but note that some aspects of subnet addressing have altered since this RFC was written (see RFC 1878[4]), so the description of mirror-image counting there and procedure text exclude subnet numbers that are now valid.

Using mirror-image counting is like normal counting starting from zero, except that the binary digits of the number are reversed. These numbers can be allocated as subnet numbers, starting from the most significant bit. Contrary to the example in RFC 1219, leading zeros (including the solitary zero in zero itself) should always be removed before the number is reversed.

Simplifying greatly, new subnets are allocated by incrementing the subnet number until a number is reached where a subnet of the required size can be accommodated or the subnet prefix becomes so long no subnets of the required size remain. If the prefix matches a common but shorter prefix, the subnet may be able to be allocated if we can lengthen the mask of the matching subnet prefix, freeing space from a previous allocation by reducing its maximum possible size. If the longest mask is always used when allocating subnets it is sufficient to just to skip matching prefixes. Note that the null prefix is common with all subsequent prefixes until its subnet mask is made smaller, extending the prefix.

The mask chosen is preferably the longest for the required subnet size—but can be as short as the length of the subnet prefix, because it can be adjusted later: made shorter if the subnetwork grows beyond its mask (if no later allocation has been made) or longer if a subnet sharing its prefix is allocated or increases size. The host number ignoring the subnet part must be allocated from 1.

As the number is incremented it grows from right to left, progressively enumerating subnets in smaller sizes. Since subnet numbers grow from right to left and host numbers from left to right, collision is delayed between the two. Allocating subnets in descending order of size is preferable in this procedure because it tends to reduce fragmentation of the address space.

The following table shows an example allocation using the sorted number of hosts in the example:

MI Number	Subnet Prefix	Network Size	Network Number	Prefix	Last Host Number	Max Host Number
(null)	00	64	0	/26	58	62
1	10	64	128	/26	177	190
01	010	32	64	/27	93	94
11	1100	16	192	/28	206	206
001 matches subnet prefix 00						
101 matches subnet prefix 10						
011	01100	8	96	/29	99	102

Note that the /28 and the /29 can grow simply by changing their netmask. A better allocation is possible if the third and fourth hosts in the sorted list are interchanged. In this case the three smallest networks would be able to grow without renumbering. Shortening a netmask is a much simpler operation than renumbering.

Of course in the real world, needs for subnet allocation do not conveniently arrive sorted in ascending order. If it happened that one of the two largest subnets was the fifth requiring allocation, fragmentation of the address space would require renumbering one of the three smallest networks to recover an address block of the necessary size.

Another point that may be worth mentioning is that most modern hosts and routers allow for multiple subnets to share the same physical subnet, allowing two smaller subnets to cover a range of addresses that would otherwise receive a single larger allocation. For example, a 40-host subnet can be allocated a /27 and a /28 rather than a /26.

—Andrew Friedman, Sydney, Australia
rbnsw-ipj@yahoo.com.au

Ed: Readers may wish to also peruse RFC 3531[5].

[1] P. Ferguson and H. Berkowitz, "Network Renumbering Overview: Why Would I Want It and What Is It Anyway?" RFC 2071, January 1997.

[2] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address Allocation for Private Internets," RFC 1918, February 1996.

[3] P. F. Tsuchiya, "On the Assignment of Subnet Numbers," RFC 1219, April 1991.

[4] T. Pummill and B. Manning, "Variable Length Subnet Table for IPv4," RFC 1878, December 1995.

[5] M. Blanchet, "A Flexible Method for Managing the Assignment of Bits of an IPv6 Address Block,"RFC 3531, April 2003.

The author responds:

Andrew is correct in stating that it is often better to try to account for future growth when assigning address space. There are many viable ways to allow for growth when allocating address spaces; hopefully, this topic will be covered more fully in a future article. I used the method in the article to illustrate how to employ the technique for working with IP addresses, rather than as an absolute best practice for allocating addresses.

—Russ White, Cisco Systems
riw@cisco.com

Dear Editor,

Russ White's article titled "Working with IP Addresses" was a nice refresher on how complicated working with IPv4 addresses has become. It should remind us all how we have gotten used to dealing with the operational expense of IPv4 address scarcity. The story about putting a frog in a pot of cold water comes to mind.

In any case, at the end of the article in the section titled "Working with IPv6 Addresses," I think the author tries too hard to fit the IPv6 address structure into the model for IPv4. Actually, it is a lot simpler.

The IPv6 address structure and textual representation was designed to avoid most of the complexities encountered in IPv4. The big differences follow:

- Addresses are represented in groups of hexadecimal digits instead of decimal digits. Hexadecimal avoids the need to convert the decimal digits to octal to find subnet boundaries. In hexadecimal there are four bits per character. This makes it easy to find the subnet boundary in an address; in many cases it is at a character boundary.
- Subnet prefix lengths are listed directly in decimal. There are no decimal subnet masks. This eliminates the need to convert decimal addresses to octets, convert the subnet masks to octets, apply the mask, and convert the result back to decimal—or to use the table and division methods described in the article.

The combination of these changes makes it much easier to work with IPv6 addresses. They are, of course, longer. The length has a few advantages besides a much larger Internet.

A byproduct of the larger address space is that most of the common subnet boundaries fall on hexadecimal digit boundaries; for example, using the example address in the article:

```
2002:FF10:9876:DD0A:9090:4896:AC56:0E01
```

The most common subnet boundary is 64 bits. The address and prefix is represented as:

```
2002:FF10:9876:DD0A:9090:4896:AC56:0E01/64
```

The subnet itself then follows:

```
2002:FF10:9876:DD0A::/64
```

The current common prefix allocated to a site is a /48. The site prefix is then:

```
2002:FF10:9876::/48
```

The current default allocation to an ISP is a /32. The ISP prefix is then:

```
2002:FF10::/32
```

These common prefix lengths can be derived directly without any need for decimal-to-octal conversions, table lookups, divisions, etc.

One of the other benefits of the larger addresses and a byproduct of IPv6 autoconfiguration is that the low-order 64 bits of an IPv6 address are reserved for the host address (called Interface Identifier in IPv6 terminology). This means that "The Hardest Subnetting Problem" described in the article is avoided completely. You can have as many hosts on a specific segment as you want in IPv6. There is no need to do this kind of calculation. This makes an initial network design trivial and, more importantly, makes later changes very easy. There is no need to redesign a subnet architecture because a few hosts need to be added to a subnet.

—Bob Hinden, Nokia
bob.hinden@nokia.com

The author responds:

Bob brings up many interesting points about IPv6, and the use of the IPv6 address space. While most IPv6 address spaces have prefix lengths that break on even octet boundaries today, we can't always count on this, for all time, so it is always good to have techniques to work with situations where the prefix length is not on an octet boundary when they do occur. As for the last problem, it is true that in all cases the subnet is the set of octets excluding the last 64 bits. But if we move the problem up one level, and ask: "What is the most efficient way to allocate out an existing /48 so customer A can get 10 subnets, customer B can get 20 subnets, etc. ?" we can see the same problem could occur at the next higher level.

—Russ White, Cisco Systems
riw@cisco.com

Corrections

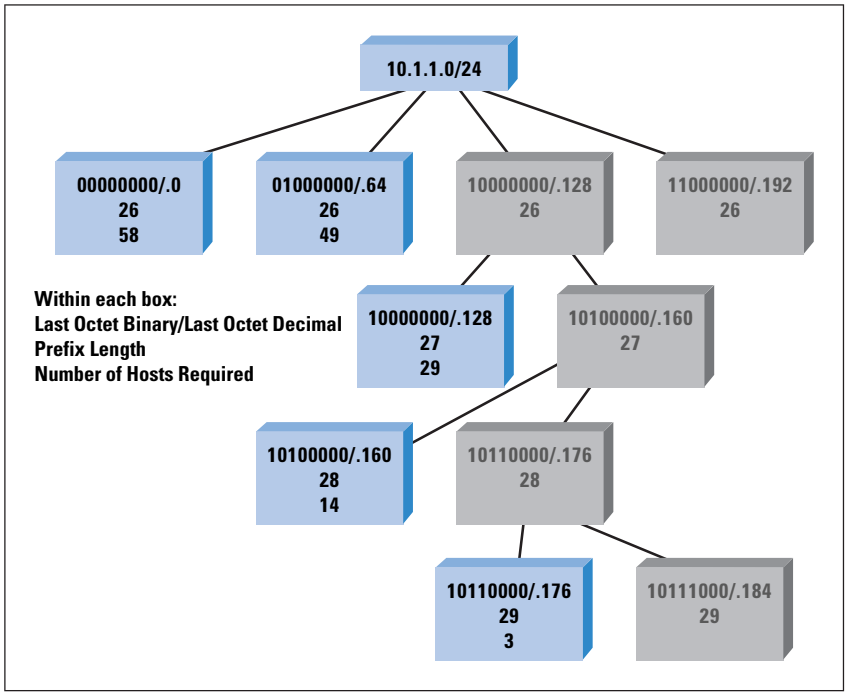
A few of our eagle-eyed readers have pointed us to some errors in IPJ, Volume 9, Number 1. The text below Figure 6 on page 29 and continuing at the top of page 30 should read as follows:

The figure shows four hosts with the addresses `10.1.0.1`, `10.1.0.2`, `10.1.0.3`, and `10.1.0.4`. Router A advertises `10.1.0.0/24`, meaning: "Any host within the address range `10.1.0.0` through `10.1.0.255` is reachable through me." Note that not all the hosts within this range exist, and that is okay—if a host within that range of addresses is reachable, it is reachable through Router A. In IP, the address that A is advertising is called a *network address*, and you can conveniently think of it as an address for the wire to which the hosts and router are attached, rather than a specific device.

For many people, the confusing part comes next. Router B is advertising `10.1.1.0/24`, which is another network address. Router C can combine—or *aggregate*—these two advertisements into a single advertisement. Although we have just removed the correspondence between the wire and the network address, we have not changed the fundamental meaning of the advertisement itself. In other words, Router C is saying: "Any host within the range of addresses from `10.1.0.0` through `10.1.1.255` is reachable through me." There is no wire with this address space, but devices beyond Router C do not know this, so it does not matter.

Also, Figure 8 on page 32 is reproduced here in its corrected form:

Figure 8: Subnet Chart



Book Review

Wireless Networking

Wireless Networking in the Developing World: A practical guide to planning and building low-cost telecommunications infrastructure, by Rob Flickenger et al., ISBN 1-4116-7837-0, 234 pages, Limehouse Book Sprint Team, January 2006. `http://wndw.net`

To quote from the book's Website:

"This book was created by a team of individuals who each, in their own field, are actively participating in the ever-expanding Internet by pushing its reach farther than ever before. Over a period of a few months, we have produced a complete book that documents our efforts to build wireless networks in the developing world."

Even though I don't live and work in what is commonly regarded as part of the developing world, I found this to be a unique and informative book, as its practical descriptions of wireless networking have application in many environments.

Given the widespread availability of the raw materials of computers, open-source software, Wi-Fi equipment, various pieces of recycled kitchenware, scrap metal, and plastic, and a wealth of online information resources, it is possible to construct inexpensive high-speed wireless network systems almost anywhere these days. However, perhaps the most visible missing component of the overall picture, but also the most valuable, is a practical path through this wealth of information on how to construct wireless networks, and a path that is based on the recent experiences of others who have constructed cost-effective and practical wireless networks in communities in the developing world. This book sets out to meet that goal.

Organization

The book starts with a description of radio physics covering the basics of the topic. It builds upon this a description of the typical radio design trade-offs between information capacity and radio penetration, and describes the commonly encountered factors of absorption, reflection, diffraction, and interference. I found the practical approach to Fresnel zone calculation and the description of the relationship between distance and antenna height so well done that I was tempted to embark on the design of a neighborhood Wi-Fi straightaway!

The chapter on network design is somewhat of a hybrid section, covering a mix of physical layout of a wireless network and TCP/IP considerations. There were the usual summaries of IP address structure and an introduction to routing.

Study of the deployment of the *Optimized Link State Routing* (OLSR) protocol is, however, more detailed. This is a link state routing protocol that is open-source, supportable by Linux-based access points, and accommodates link quality metrics into the routing protocol metric. I found the consideration of the link budget in this section a useful practical description of the considerations that are unique to the wireless world, and the worked examples are excellent, together with some useful references to online tools. This chapter is relatively dense, and many topics are covered in a relatively short space. I suspect that an interested reader would want to drill down further before feeling confident enough to manage a service network, but some carefully chosen references to further reading are there, so that the reader can follow up this introductory material with more specialized references.

The section on antennas and transmission lines was also well-structured. I had heard of using cylindrical cans as Wi-Fi antennas, but knew little of the detail of how to actually do it. This book not only explains their design, but provides a step-by-step illustrated guide to their construction. It also provides a good description of what is involved in outdoor installation of wireless equipment. The consideration of commercial solutions as compared to the do-it-yourself approach was carefully presented, as was the section devoted to security considerations.

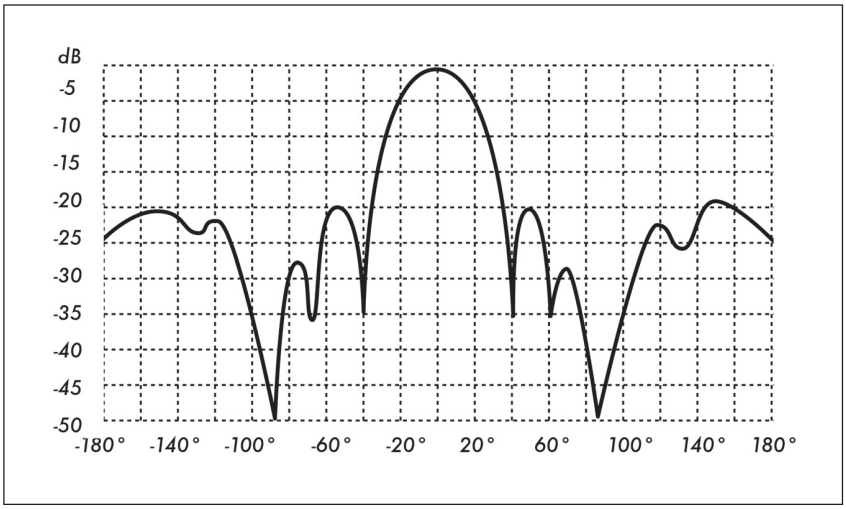
Aside from the technical considerations, the book also has some very interesting case studies of wireless networking projects, and was careful to include both success and failure stories. The issues in the developing world about combining technical capability with practical business solutions for communities that can be financially self-sustaining are indeed challenging, as the case studies show. They provide not only useful information about related experiences in setting up such network services, but also show how such projects can be assessed in a constructive manner.

Thoughtfully Written

Having spent some time working in this area myself as part of the ISOC *Developing Countries Workshop* training team, I have developed an appreciation of what constitutes truly useful and valuable training material, and this book is perhaps the best example I've seen yet. It is practical, helpful, technically accurate, and relatively complete in terms of coverage of material. Where the book does not dive into fine detail it provides useful references for further reading. The book is thoughtfully written in a simple non-nonsense style and does not hide behind technical jargon. Above all, it is material that can instill confidence that these networks can readily be built and operated by people like you and me.

I certainly would not call myself an expert after reading this book, but the next time a radio technician arrives in the office and starts talking about radiation patterns, front-to-back ratios, and the relative merits of omnis and yagis, at least I'll have an idea of what he is talking about. Even better, I might even be able to show him my own modest efforts in do-it-yourself Wi-Fi networking by then!

Rectangular plot of a Yagi Radiation Pattern from Chapter 4 of the book



Publishing Model

This is not a conventional technical book in the sense that it does not come with a conventional technical book price tag. The book is published in a manner as to be readily available in the developing world, so an online publication model has been used here. The PDF is freely available under a *Creative Commons* Attribution-Share-Alike 2.5 license at **http://wndw.net**, and they have managed to squeeze all 254 pages into an impressively small 1.92-MB file. You can find related resources and ways that you can assist in this project at **http://wndw.net**.

—Geoff Huston, APNIC
gih@apnic.net

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

Internet Governance

The *World Summit on the Information Society* (WSIS) was held in two phases. The first phase took place in Geneva in December 2003, and the second phase took place in Tunis in November 2005. The so-called "WSIS Outcome Documents" are now available at:

`http://www.itu.int/wsis/promotional/outcome.pdf`

The follow-on to WSIS is called the *Internet Governance Forum* (IGF). The forum will hold its first meeting in Athens, Greece October 30th to November 2nd, 2006. For more information visit:

`http://www.intgovforum.org/`

The *Internet Society* (ISOC) played an active part in the WSIS process. You will find background information here:

`http://www.isoc.org/isoc/conferences/wsis/index.shtml`

DNS Root Name Servers Explained

Daniel Karrenberg of RIPE NCC has written two "Member Briefings" on the subject of DNS root servers that can be found on the ISOC Website:

`http://www.isoc.org/briefings/019/`
`http://www.isoc.org/briefings/020/`

Internationalized Domain Names

Internationalized Domain Names (IDNs) are, according to the ICANN Website, "...domain names represented by local language characters. Such domain names could contain letters or characters from non-ASCII scripts (for example, Arabic or Chinese). Many efforts are ongoing in the Internet community to make domain names available in character sets other than ASCII." ICANN has established an information area on its Website with links to more information about IDNs. See:

`http://icann.org/topics/idn/`

The ISP Column

Geoff Huston is well known to readers of this journal. He also hosts *The ISP Column* that can be found here:

`http://www.isoc.org/pubs/isp/index.shtml`

The Internet Protocol Journal
Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

September 2006 Volume 9, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

One of the most successful networking technologies of recent years has been IEEE 802.11 or, as it is commonly known, "Wi-Fi." Wireless networks have seen widespread deployment within organizations as well as in public "hotspots" all over the world. As a frequent traveler, I am very pleased with this development. It has been a long time since I had to resort to a modem and phone line in order to access e-mail or use the Web. Wireless networks have truly changed the way we use the Internet. Our first article, by T. Sridhar, explores the emerging use of *Wireless LAN Switches* in wireless access networks.

IPv6 is a technology that perhaps should have been widely deployed by now, but wide deployment has not happened yet, for numerous reasons. This journal has covered many aspects of IPv6. This time, Iljitsch van Beijnum looks at some of the details you need to be aware of when considering a move to IPv6. The article is adapted from his book *Running IPv6,* which was reviewed in our December 2005 issue.

In previous editions of IPJ we have pointed you to other sources of information, such as *The IETF Journal,* Geoff Huston's *ISP Column,* and other documents available from the Internet Society Website at **http://www.isoc.org**. This time I want to make you aware of an article that originally appeared in *Apster*, the newsletter of the *Asia Pacific Network Information Centre* (APNIC), one of the five *Regional Internet Registries* (RIRs). The article is entitled "IP Addressing in China and the Myth of Address Shortage," and you will find the URL for it in our "Fragments" section. If you want to further explore the work of the RIRs, you can start by visiting the *Number Resource Organization* (NRO) at **http://nro.net**.

You may have read that both of our sister publications, *Packet* and *IQ Magazine,* are publishing their final issues this September. Naturally, this has led to some of our readers asking what is in store for IPJ. We want to reassure you that we intend to continue publishing IPJ in both its paper and online forms. Plans are also under way to enhance our Website to provide you with more tools and resources. If you have suggestions for the Website, please send us a note at **ipj@cisco.com**.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Wireless LAN Switches — Functions and Deployment

by T. Sridhar, Flextronics

Deployment of *Wireless LAN* (WLAN) switches is increasing in enterprise networks. These devices, which can be standalone switches or integrated into a blade on an enterprise class switch, are useful for the management and control of WLAN access points. Although their deployment is a relatively new phenomenon, such control and configuration functions have existed before in WLAN controller devices.

WLAN switches connect to the WLAN *access points* (APs) through wired connections (through a switch port). They also connect to the enterprise network through their other switch ports. The switches are the "gateway" to the wired enterprise—all frames from WLAN clients have to pass through the WLAN switches to the enterprise network.

To understand the motivation for WLAN switches and their operation in the network, it is useful to view the WLAN network architecture and the functions of the access points. We can view the WLAN switch as the control function and the APs as the wireless termination function.

This article presents the function of WLAN switches and controllers by detailing WLAN network architectures along with functions of the AP and controller. It also presents the various functions on the controller to AP interface. Subsequently, it outlines variables related to Layer 2/3 mobility in the centralized architecture and concludes by presenting some common myths and reality about these architectures.

This article uses the term *Wireless Termination Point* (WTP) to refer generically to APs and the term *Access Controller* (AC) to refer generically to the WLAN control function (whether implemented on a WLAN switch or standalone controller).

WLAN Network Architectures

Three types of WLAN network architectures are commonly deployed:

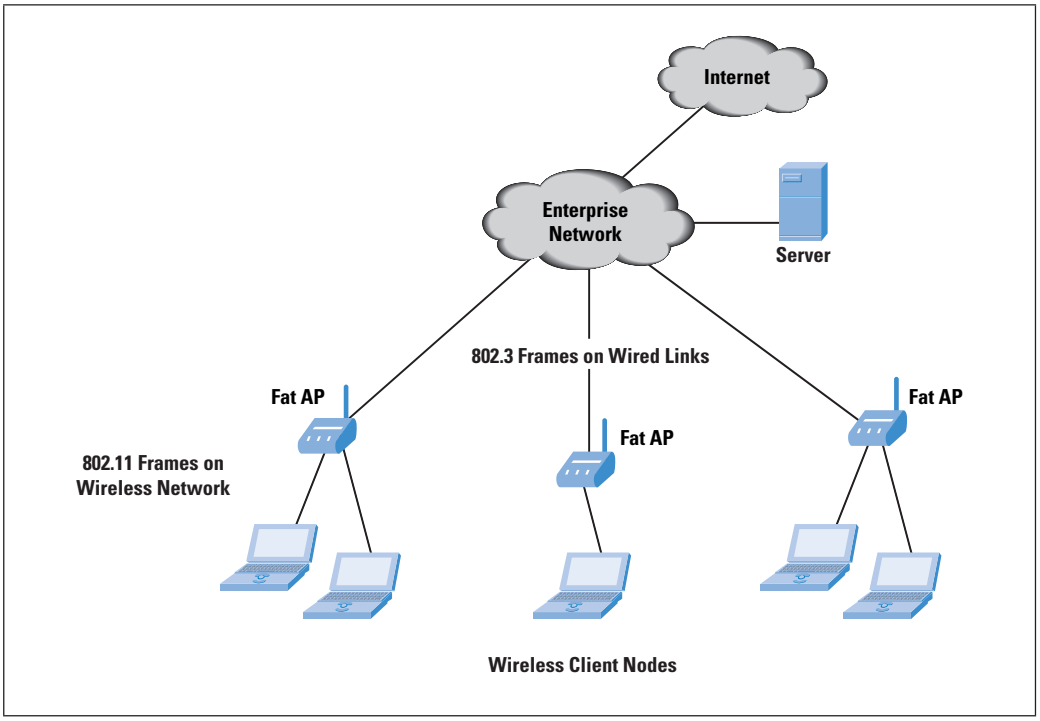
1. Autonomous Architecture
2. Centralized Architecture
3. Distributed Architecture

The following sections describe these architectures in greater detail.

Autonomous Architecture

In the autonomous architecture, the WTPs completely implement and terminate the 802.11 function so that frames on the wired LAN are 802.3 frames. Each WTP can be independently managed as a separate network entity on the network. The access point in such a network is often called a "Fat AP" (see Figure 1).

Figure 1: FAT APs in Autonomous WLAN Network Architecture

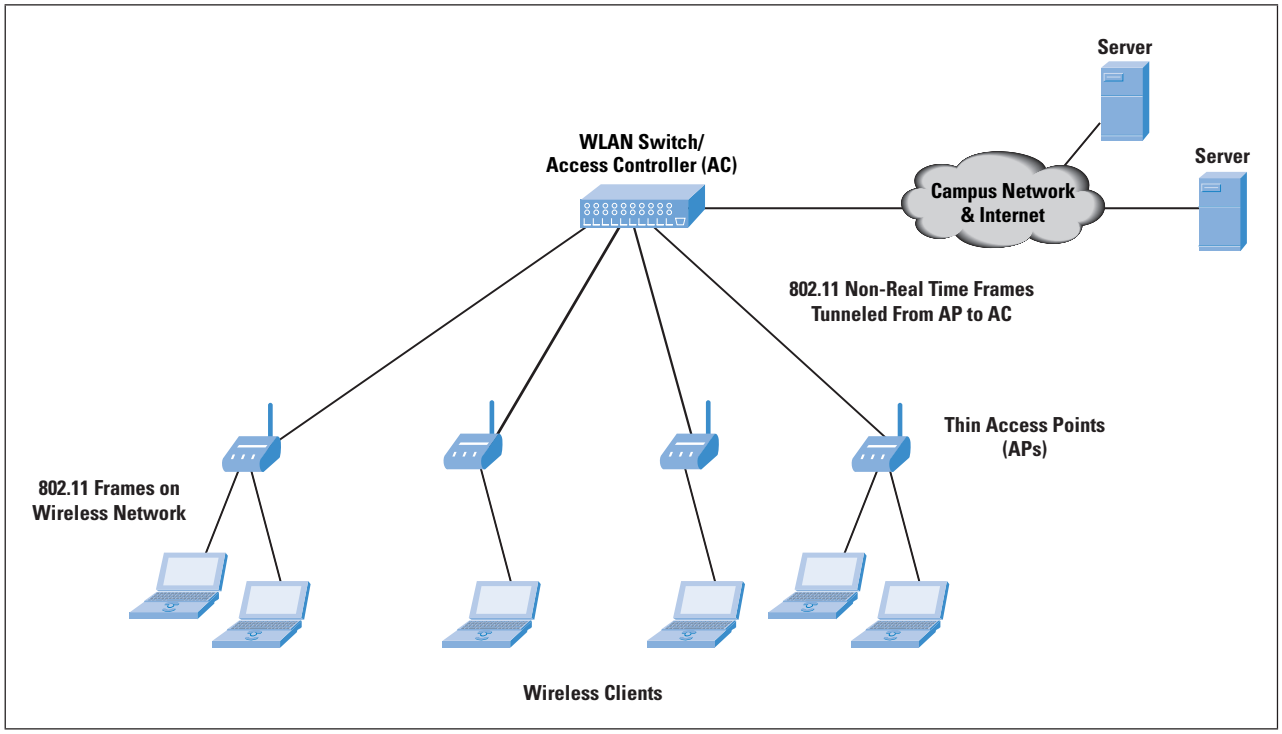


During the initial stages of WLAN deployment, most APs were autonomous APs, and manageable as independent entities in the network. During the past few years, centralized architectures (discussed next) with ACs and WTPs have gained popularity. The primary advantage of the centralized architecture is that it provides network administrators with a structured and hierarchical mode of control for multiple WTPs in the enterprise.

Centralized Architecture

The centralized architecture is a hierarchical architecture that involves a WLAN controller that is responsible for configuration, control, and management of several WTPs. The WLAN controller is also known as the *Access Controller* (AC). The 802.11 function is split between the WTP and the AC. Because the WTPs in this model have a reduced function as compared to the autonomous architecture, they are also known as "Thin APs." Some of the functions on the APs are variable, as discussed in the following section (see Figure 2).

Figure 2: Thin APs in Centralized WLAN Network Architecture



Distributed Architecture

In the distributed architecture, the various WTPs can form distributed networks with other WTPs through wired or wireless connections. A mesh network of WTPs is one example of such an architecture. The WTPs in the mesh can be linked with 802.11 links or wired 802.3 links. This architecture is often used in municipal networks and other deployments where an "outdoor" component is involved. This article does not address the distributed architecture.

WTP Functions – Fat, Thin, and Fit APs

To understand the autonomous and centralized architecture, it is useful to look at the functions performed by the APs. We start with the Fat APs, which form the core of the autonomous architecture, followed by the Thin APs, which were specified as part of the WLAN switch- or controller-based centralized architecture. The article will then outline the functions of a new variant called the "Fit AP," an optimized version of the AP for centralized architectures.

Fat Access Points

Figure 1 shows an example of an autonomous network with a fat access point. The AP is an addressable node in the network with its own IP address on its interfaces. It can forward traffic between the wired and wireless interfaces. It can also have more than one wired interface and can forward traffic between the wired interfaces—similar to a Layer 2 or Layer 3 switch. Connectivity to the wired enterprise can be through a Layer 2 or Layer 3 network.

It is important to understand that there is no "backhauling" of traffic from the Fat AP to another device through tunnels. This aspect is important and is addressed when discussing the other AP types. In addition, Fat APs can provide "router-like" functions such as the *Dynamic Host Configuration Protocol* (DHCP) server capabilities.

Management of the AP is done through a protocol such as the *Simple Network Management Protocol* (SNMP) or the *Hypertext Transfer Protocol* (HTTP) for Web-based management and a *Command-Line Interface* (CLI). To manage multiple APs, the network manager has to connect to each AP through one of these management schemes. Each AP shows up on the network map as a separate node. Any aggregation of the nodes for management and control has to be done at the *Network Management System* (NMS) level, which involves development of an NMS application.

Fat APs also have enhanced capabilities such as *Access Control Lists* (ACLs), which permit filtering of traffic for specific WLAN clients. Another significant capability of these devices is configuration and enforcement of *Quality of Service* (QoS)-related functions. For example, traffic from specific mobile stations might need to have a higher priority than others. Or, you might need to insert and enforce IEEE 802.1p priority or *Differentiated Services Code Point* (DSCP) for traffic from mobile stations. In summary, these APs act like a switch or router in that they provide many of the functions of such devices.

The downside of such APs is complexity. Fat APs tend to be built on powerful hardware and require complex software. These devices are expensive to install and maintain because of the complexity. Nevertheless, the devices have uses in smaller network installations.

Some Fat AP installations still use a controller at the back end for control and management functions. These controllers lead to a slightly scaled-down version of the Fat AP, called, not surprisingly, a Fit AP, discussed later.

Thin Access Points

As their name indicates, Thin APs are intended to reduce the complexity of APs. An important motivation for this reduction is the location of APs. In several enterprises, APs are plenum-mounted (and thus in hard-to-reach areas) so that they can provide optimum radio connectivity for end stations. In environments like warehouses, this is even more evident. For such reasons, network managers prefer to install APs just once and not have to perform complex maintenance on them.

Thin APs are often known as "intelligent antennas," in that their primary function is to receive and transmit wireless traffic. They backhaul the wireless frames to a controller where the frames are processed before being switched to the wired LAN (see Figure 2).

The APs use a (typically secure) tunnel to backhaul the wireless traffic to the controller. In their most basic form, Thin APs do not even perform WLAN encryption such as *Wired Equivalence Privacy* (WEP) or *WiFi Protected Access* (WPA/WPA2). This encryption is done at the controller—the APs just transmit or receive the encrypted wireless frames, thereby keeping the APs simple and avoiding the necessity to upgrade their hardware or software.

The introduction of WPA2 necessitated encryption on the controller. Although WPA was hardware-compatible with WEP and required only a firmware upgrade, WPA2 was not backward-compatible. Instead of replacing APs across the enterprise, network managers could just backhaul the wireless traffic to the controller where the WPA2 decryption was done, and the frames were sent on the wired LAN.

The protocol between the AP and the controller for carrying the control and data traffic was proprietary. Also, there is no capability to manage the AP as a single entity on the Layer 2/3 network—it can be managed only through the controller, to which the NMS can communicate through HTTP, SNMP, or CLI/Telnet. A controller can manage and control multiple APs, implying that the controller should be based on powerful hardware and often be able to perform switching and routing functions. Another important requirement is that the connectivity and tunnel between the AP and the AC should ensure low delay for packets between those two entities.

With Thin APs, QoS enforcement and ACL-based filtering are handled at the controller—not a problem because all the frames from the AP have to pass through the controller anyway. Centralized control functions for ACLs and QoS are not new—they were implemented in networks with Fat APs too. Such installations have controllers that act as the gateway for managing traffic from APs to the wired network. However, the controller function takes on a new dimension with Thin APs, especially with respect to the data plane and forwarding functions. The controller function subsequently was integrated into Ethernet switches that connected the wireless and wired LANs—the motivation for the family of devices known as WLAN switches.

The Wireless MAC architecture in this scenario is known as the *Remote MAC* architecture. The entire set of 802.11 MAC functions is offloaded to the WLAN controller, including the delay-sensitive MAC functions.

Fit Access Points

Fit APs are gaining in popularity in that they try to take advantage of the best of both worlds—that is, the Fat APs and the Thin APs. A Fit AP provides the wireless encryption while using the AC for the actual key exchange. This approach is used for newer APs that use the latest wireless chipsets supporting WPA2. The management and policy functions reside on the controller that connects to multiple APs through tunnels.

Also, Fit APs provide additional functions such as DHCP relay for the station to obtain an IP address through DHCP. In addition, Fit APs can perform functions such as VLAN tagging based on the *Service Set Identifier* (SSID) that the client uses to associate with the AP (when the AP supports multiple SSIDs).

Two types of MAC implementations are possible with Fit APs, known as the *Local MAC* and the *Split MAC* architectures. Local MAC is where all the wireless MAC functions are performed at the AP. The complete 802.11 MAC functions, including management and control frame processing, are resident on the APs. These functions include time-sensitive functions (also known as *Real Time MAC* functions).

The Split MAC architecture divides the implementation of the MAC functions between the AP and the controller. The real-time MAC functions include functions such as beacon generation, probe transmission and response, control frame processing (for example *Request to Send* and *Clear to Send*—RTS and CTS), retransmission, and so on. The non-real time functions include authentication and deauthentication; association and reassociation; bridging between Ethernet and Wireless LAN; fragmentation; and so on.

Vendors differ in the type of functions that are split between the AP and the controller, and in some cases, even about what constitutes real time. One common implementation of a Fit AP involves local MAC at the AP and control and management functions at the AP.

Access Controller and Control Functions

The next critical component of the Centralized WLAN Architecture is the *Access Controller* (AC). For the following discussion, we consider the controller function to be implemented on a WLAN switch and call the function an AC. We also use the term "WTP" to refer to APs (fat, thin, or fit).

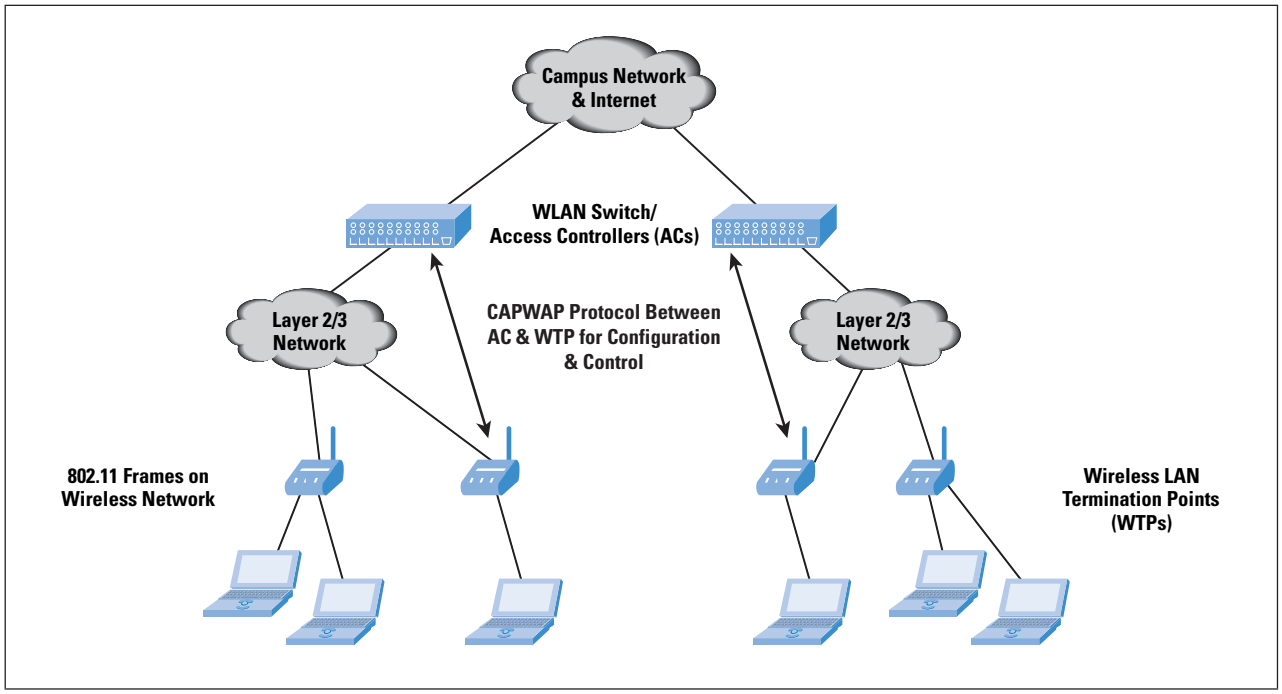
The *Control and Provisioning of Wireless Access Points* (CAPWAP) Working Group in the IETF is working on defining the interface and protocol between an AP and its controlled WTP. This section uses the CAPWAP framework to detail the interface between the AC and the WTP. [3,4,5]

Figure 3 shows an enterprise network with multiple ACs and WTPs. The WTPs can be connected to the ACs through a Layer 2 (switched) or Layer 3 (routed) network. The interface between the WTP and the AC is responsible for the following:

- Discovery and selection of an AC by WTP
- Firmware download to the WTP by the AC—upon startup and upon triggering by the WTP
- Capabilities negotiation between the WTP and the AC

- Mutual authentication between the WTP and the AC
- Configuration, status, and statistics exchange between the WTP and the AC
- QoS mapping across the wired and wireless segments

Figure 3: Centralized WLAN Architecture with Multiple ACs, WTPs and CAPWAP Protocol Context



In addition, although CAPWAP does not explicitly define all the details, the AC performs functions such as *Radio Resource Management* (RRM) and rogue AP detection based on configuration and monitoring of the various access points in its domain of control. The extent of these functions varies according to the vendor implementation. Another important function provided by ACs is mobility management. The following sections provide more detail about these functions, with specific reference to CAPWAP. Note that the CAPWAP protocol, which is based on the Cisco *Lightweight Access Point Protocol* (LWAPP), is still under development in the IETF, as of the writing this article (March 2006).

Discovery and AC Selection

A WTP discovers an AC to connect to through discovery request messages, to which one or more ACs can respond (depending on the network topology). Communication between the AC and the WTP is through the *User Datagram Protocol* (UDP). The WTP determines which AC to connect to and then tries to establish a secure session with the AC. Subsequent CAWAP packets are sent over the secure session.

Subsequently a configuration exchange takes place between the AC and WTP. This exchange includes:

- IEEE SSID
- Security parameters (for WEP, WPA, and WPA2)
- Data rate that is to be advertised (11 or 54 Mbps)
- Radio channels to be used

CAPWAP Functions

CAPWAP control messages include the following message types:

- Discovery
- WTP configuration—used to push a specific configuration to the WTP and also to retrieve statistics from a WTP; statistics includes information such as:
  - Number of fragmented frames, multicast frames transmitted and received
  - Number of transmit retries, excessive retries (failed count)
  - Number of successfully transmitted and failed *Requests to Sends* (RTS)
  - Number of errored frames: duplicate frames, failed acks, decryption errors, *frame-check-sequence* (FCS) error count, etc.

Configuration includes information such as beacon period, maximum transmit power level, *Orthogonal Frequency Division Multiplexing* (OFDM) control, antenna control, supported rates, QoS, encryption, and so on.

- *Mobile session management*—to push specific mobile policies to the WTP

  ACs can add policy information about specific mobile devices that can include security parameters that the WTP should apply for that mobile device. It can indicate whether the WTP should forward or discard traffic for that mobile device.

- *Firmware management*—used to push a specific firmware image to the WTP

AC and WTP Interaction

The WTP provides information such as hardware, software, or boot version; maximum number of radios; radios in use; encryption capabilities; type of radio (802.11b/g/a/n); type of MAC (local, split, or both); tunneling modes; and frame type between AC and WTP (for example, local bridging or native bridging—that is, encapsulating all user payloads as native wireless frames).

The AC information includes hardware or software version, number of mobile stations currently associated with the AC, number of WTPs currently attached to the AC, maximum numbers for each of these, security parameters (authentication credentials) between AC and WTP, control IPv4 or IPv6 address, and so on.

Because the WTPs fall under the category of "Fit APs," they can also be configured with an IP address from the AC. Another parameter that can be configured is ACLs at the MAC address level.

Rebooting (reset) of the WTP can be done by the AC at any time. Independently, the WTP can request a new image through an *Image Data Request*, which is followed by an *Image Data Response* and the image data itself.

Events are sent by the WTP when it determines that it has important information to send to the AC. Such information can include data transfer messages that can be used to deliver debug information from the WTP to the AC.

Radio Resource Management

Radio resource management is a generic term used to describe the control and configuration of radios on the AP. The type of control includes reducing and increasing the strength automatically or on user input—for example, if two WTPs controlled by an AC are interfering with each other, the AC can send a signal to one of the APs to reduce its strength. It can also do this based on user configuration.

Several WTPs are designed to also be used as "Air Monitors;" that is, they can monitor channels when not transmitting. Opinion is still divided on whether this mode of using WTPs is efficient—some vendors use dedicated air monitors instead of having their WTPs do double duty. With dedicated air monitors, it is much easier to scan and monitor all channels without having to worry about degrading the service for client stations.

Air monitors can forward information about other access points to the AC. The AC can determine if the information is for a valid WTP (that is, one that is supposed to be on the network and has, in fact, registered with the AC) or for a "rogue" access point. If it is for a rogue access point, the AC can perform multiple steps to prevent clients from attaching to this AP—for example, it can instruct the air monitor to "jam" this rogue AP by increasing the transmit power on the same channel.

Mobility Management

Mobility management can take two forms—Layer 2 and Layer 3 mobility. Consider a client moving from one WTP to another, a scenario that can happen when a user with a laptop moves between two conference rooms within the same building. The client station reassociates with the new WTP, after which authentication is performed. Note that the association with the previous AP is "broken" before the association with the new AP is "made;" thus handoff in WLANs is known as "break before make." Although this approach can lead to potential traffic disruption (and retransmissions), it is chosen over "make before break" (used in cellular telecommunications) to keep the client radio simple and less expensive.

One way to envision Layer 2 and Layer 3 mobility is to treat Layer 2 mobility as movement between APs under the control of the same AC (that is, Layer 3 network), whereas Layer 3 mobility is movement between APs under the control of different ACs.

Layer 2 Mobility

Layer 2 mobility means that when the station moves from one WTP to another, there is no impact on the IP addressability, effectively meaning that all the APs are on the same Layer 2 network and implying that they are connected to the same AC (see Figure 4). To prevent loss of data destined to the Layer 2 client, the WLAN switch must now forward client data to the new WTP. After the client association, the new WTP sends out an Ethernet frame to the AC with the client's MAC address as the source address. The switch now associates the client's MAC address to the port on which the new WTP is connected.

Although this process works well with Layer 2 (switched network) connectivity between the APs and the AC, it requires a slightly different approach when tunnels are used between them. The AC moves the mapping of the client to a different tunnel (that is, a virtual port) when it receives the MAC frame from the new WTP.

Another concern to be addressed with Layer 2 handoff is the buffering of data at the WTP. In normal circumstances, the switch or AC is not aware of the handoff until it hears from the new WTP. However, with enhanced statistics available at the WTP, it can determine that the specific client has moved away from the old WTP and stop forwarding data to the old WTP. These statistics can include maximum retry attempts on the *Carrier Sense Multiple Access/Collision Avoidance* (CSMA/CA) MAC layer protocol on the wireless link. The switch does not need to buffer the data because it is not clear when the handoff to the new WTP will occur. This approach helps avoid wasteful traffic on the link between the old WTP and the AC.

Some vendors have approached this problem differently with Fat APs. There, the APs might buffer the traffic until they see a frame from the switch indicating that the client is now on a different switch port. These APs then send the buffered traffic to the switch, which forwards that to the new WTP. Because our intent is to lower the complexity of the WTPs, this approach is not a preferred one in the Centralized AC + WTP architecture.

Another important feature of Layer 2 roaming is preauthentication that needs to be done on the new WTP. Through 802.11i, clients can preauthenticate with neighboring WTPs so that roaming to a different WTP does not involve the lengthy authentication process of *Pairwise Master Keys* (PMKs) being sent to the new WTP. (The *Pairwise Transient Keys* (PTKs) still need to be derived.)

When the AC maintains the PMK for a specific client (through interaction with a RADIUS server), this process is automatic—that is, the AC can send the client-specific PMK to the new WTP. The encryption of 802.11 frames is still done by the old and new WTPs with the new PTKs.

Layer 3 Mobility

Layer 3 mobility involves the client retaining the same IP address while moving across multiple APs. This often happens when the client has published its IP address to multiple nodes. Such a scenario is likely in peer-to-peer communications and when the mobile station needs to act as a server for some function. It is desirable that the correspondent nodes communicating with the mobile node not have to change their configuration whenever the mobile node moves to a new Layer 3 network.

This problem of Layer 3 mobility is solved by *Mobile IP*[6]. We do not discuss the details of Mobile IP here except to indicate that it has three distinct components. The *Home Agent* (HA) on the client's home network is responsible for the address of the client. All packets destined to the client's (invariant) IP address are sent to the Home Agent. If the client is on the home network, the HA forwards the packets directly to the client. If it is on a foreign or visited network, the HA forwards the packets to a *Foreign Agent* (FA) that is on the visited network.

To do this, it has to set up a tunnel to the FA—which is usually a *Generic Routing Encapsulation* (GRE) or IP-in-IP tunnel.

After stripping out the original packet from the tunnel, the FA is responsible for forwarding the packet to the client. This description is a simplification—numerous other steps are involved here. The important factor in a wireless LAN scenario for Layer 3 client mobility is where the Mobile IP endpoint resides. Some client stations include a software stack for a MIP client.

This *Client MIP* (CMIP) software:

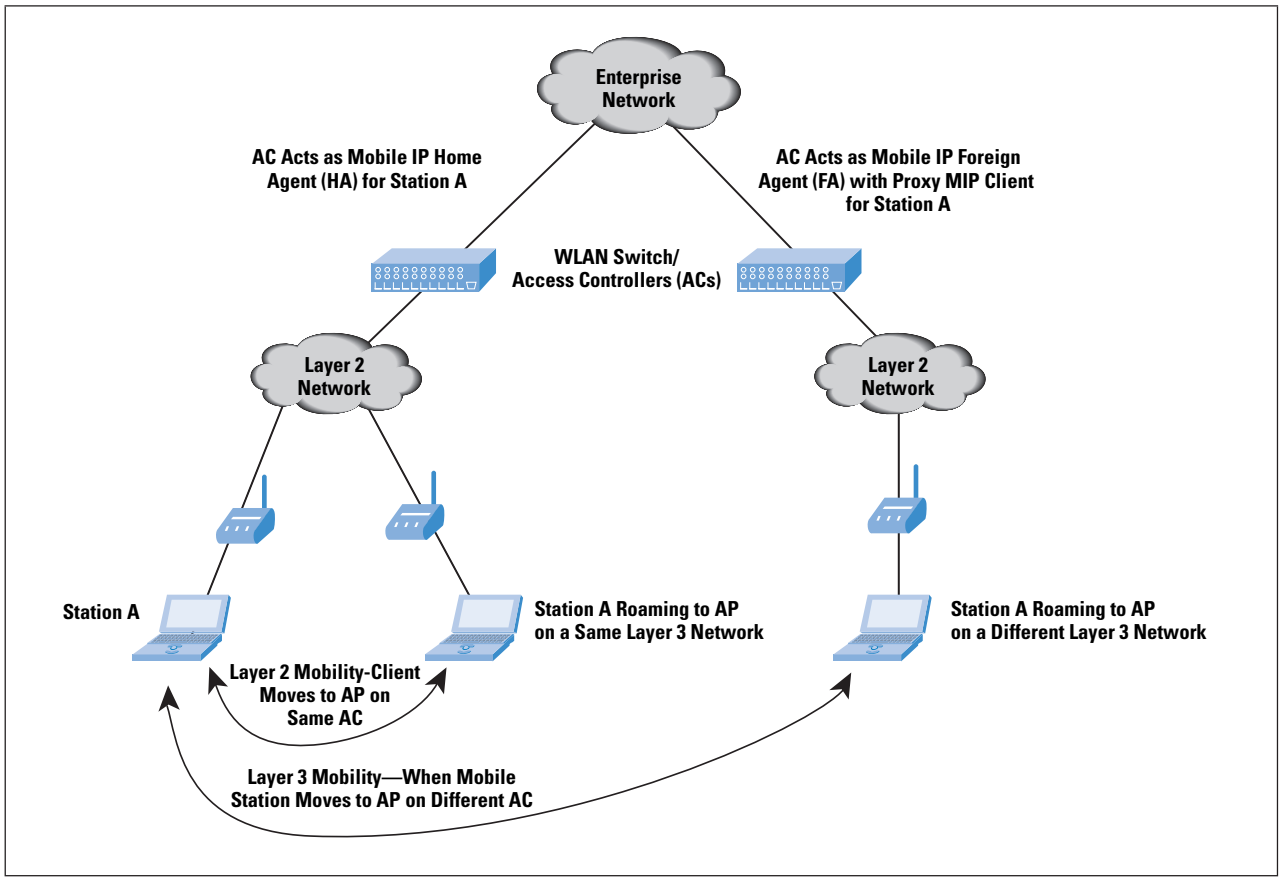
- Strips out the MIP header in the packet
- Inserts a new header to spoof the client's higher-layer applications into believing that the packets were destined for the client's IP address on the foreign network

The CMIP approach was the recommended approach for implementing MIP. However, it has the disadvantage of having to add a MIP client to every mobile station in the network—a setup that can become cumbersome when there are a large number of mobile stations.

The Centralized AC + WTP architecture offers a way of alleviating this problem. Some AC/WLAN switch vendors have implemented the MIP function on the AC so that the client never needs to be changed. Some implementations call this a *Proxy MIP* function.

The AC acts as an FA to terminate the tunnel from the HA and also performs the translation of the packets to the client's address on the visited network when forwarding packets to the client. When the client sends Layer 3 packets out, it sends them through the AC, which, in turn, modifies the headers for the source IP address and tunnels the packets to the HA. This process is called "reverse tunneling" (see Figure 4).

Figure 4: Layer 2 and Layer 3 Mobility in Centraliized WLAN Network Architecture



When you consider a large enterprise network topology with multiple ACs and APs, you can envision the MIP tunnels to be established between the various ACs. (That is, they act as Foreign Agents for one set of users and as Home Agents for another set.) From a scalability perspective, it is important that the ACs have the necessary horsepower and switching capability (switching between tunnels from the APs to the ACs to the tunnels between the ACs).

WLAN Switches and Centralized Architectures – Common Myths

Previous sections considered various aspects of the Centralized AC + WTP architecture and some of the implementation factors. This section outlines some common myths about these architectures and implementations. The intent is to examine this still-evolving area to facilitate clarity.

1. *Myth 1: ACs need to perform switching functions—hence the name WLAN switches.*

   There is no such requirement. In fact, the earliest ACs were appliances (and in some cases, PCs running Linux). The control function is the important part of the implementation—the switching is often included to accelerate the forwarding of traffic to and from the APs.

2. *Myth 2: Rogue WTP detection is a standard function of ACs.*

   This is a desired function in several implementations but is not necessarily "standard." One reason is that this is an area of differentiation among vendors (for example, the algorithms they use to classify a WTP as a rogue WTP). Another reason is that the ACs have to rely on APs or air monitors, and this reliance varies according to implementation.

3. *Myth 3: The delineation between Fat, Thin, and Fit APs is clearly defined.*

   There are several types of implementations of AP (and AC) functions, so this myth is not necessarily true. For a sample of the taxonomy (snapshot) of WTP and AC implementations, see RFC 4118[4].

4. *Myth 4: Layer 2 and Layer 3 mobility are standard in AC + WTP architectures.*

   This is not really true. The Proxy MIP implementation for Layer 3 mobility is a step in this direction, but most AC vendors rely on proprietary mechanisms for AC-AC communication and Layer 3 mobility.

5. *Myth 5: Security functions such as firewall, intrusion detection, and so on are not a function of ACs.*

   Some vendors have debunked this argument and implemented such functions in their AC. This is an area for vendor differentiation.

Summary

This article has provided the functions and deployment of WLAN switches by detailing the architectures that rely on a centralized controller managing a set of wireless termination points. It outlined some major aspects of the CAPWAP control functions and the concerns related to Layer 2 and Layer 3 mobility while implementing an AC + WTP architecture. Although protocol standardization is being done in the IETF for this emerging area, there is still sufficient scope for vendor differentiation.

References

[1] "IEEE 802.11i and Wireless Security," David Halasz, **www.embedded.com**, August 25, 2004.

[2] Rich Seifert, *The Switch Book: The Complete Guide to LAN Switching Technology,* ISBN 0471345865, Wiley, 2000.

[3] B. O'Hara, et al., "Configuration and Provisioning for Wireless Access Points (CAPWAP): Problem Statement," RFC 3990, February 2005.

[4] L. Yang, et al., "Architecture Taxonomy for Control and Provisioning of Wireless Access Points (CAPWAP)," RFC 4118, June 2005.

[5] P. Calhoun, Editor, "CAPWAP Protocol Specification," (work in progress), Internet Draft, **draft-ietf-capwap-protocol-specification-00**, February 24, 2006.

[6] C. Perkins, Editor,"IP Mobility Support for IPv4," RFC 3344, August 2002.

[7] Edgar Danielyan, "IEEE 802.11," *The Internet Protocol Journal,* Volume 5, No. 1, March 2005.

[8] Gregory R. Scholz, "Securing Wireless Networks," *The Internet Protocol Journal,* Volume 5, No. 3, September 2002.

T. SRIDHAR is Vice President of Technology at Flextronics in San Jose, California. He received his BE in Electronics and Communications Engineering from the College of Engineering, Guindy, Anna University, Madras, India, and his Master of Science in Electrical and Computer Engineering from the University of Texas at Austin. He can be reached at **T.Sridhar@flextronics.com**

IPv6 Internals

by Iljitsch van Beijnum

This article discusses some of the protocol details you should be aware of when planning a transition from IPv4 to IPv6. Although it is not intended as a complete step-by-step guide, this article explains the differences between IPv4 and IPv6 as they relate to actually operating a network. Vendor- and operating system-specific details can be found in the book from which this text was adapted, and further information is available in the references.

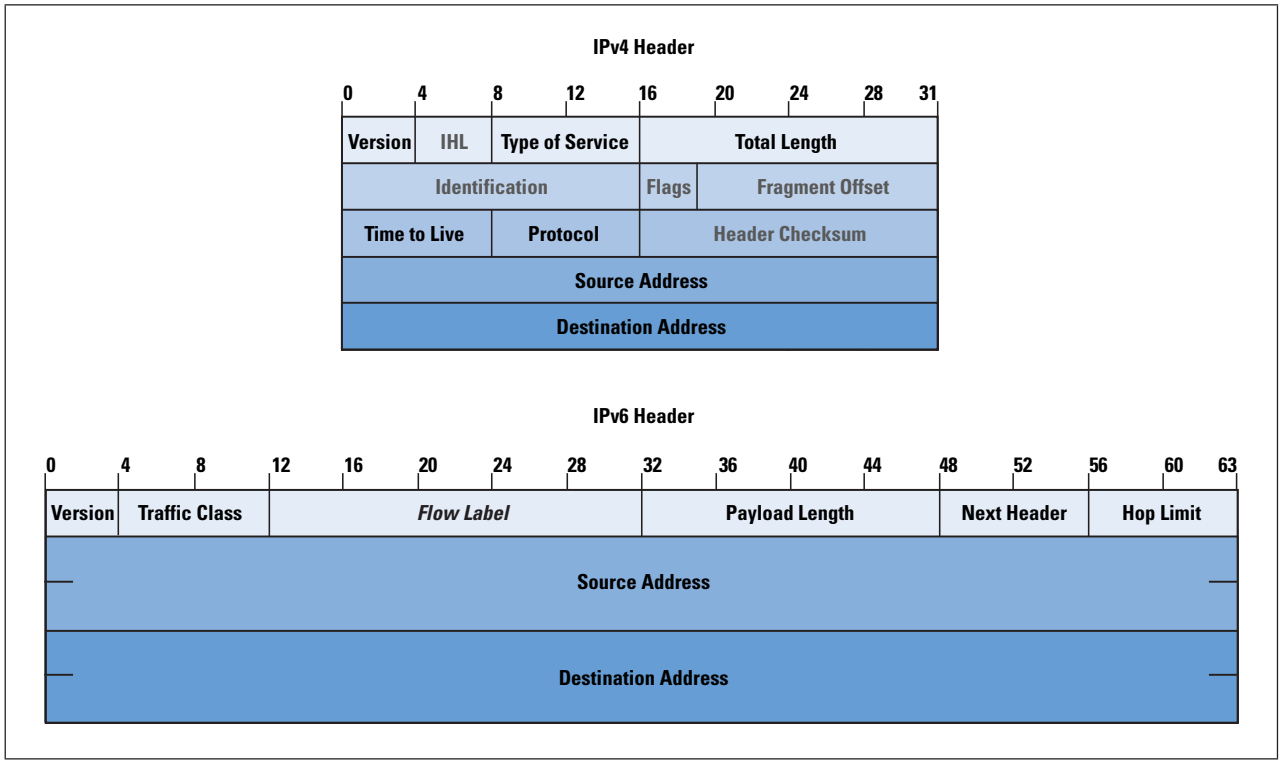
The easiest way to observe—in action—the mechanisms discussed in this article is to set up an IPv6 router on the local subnet and enable IPv6 on the operating system of your choice, if it is not enabled by default. If "native" IPv6 connectivity is not possible, you can set up automatic IPv6 tunneling or use a manually configured IPv6-in-IPv4 tunnel. Getting portable IPv6 address space from a *Regional Internet Registry* (RIR)[1] is a topic worthy of its own article, but *6to4*[2] creates 65,536 IPv6 subnets from a single IPv4 address, and service providers that provide IPv6 connectivity—either natively or over manually configured tunnels—are usually quite generous with IPv6 address space. However, you need to renumber when changing *Internet Service Providers* (ISPs), or when changing IPv4 addresses with 6to4. Most router vendors currently support IPv6 routing, but all widely used general-purpose operating systems can also route IPv6.

When you have IPv6 connectivity, the browser that comes with your system should be able to work over IPv6 (visit **`http://www.kame.net/`**), and there are v6 versions of *ping* and *traceroute* (called *ping6* and *traceroute6*) to determine IPv6 connectivity. More and more applications work over IPv6, but many still do not.

Differences Between IPv4 and IPv6

All knowledge about IPv6 begins with studying the IPv6 header format and the ways in which it is different from the IPv4 header format. Even though at the time the IPv6 specifications were written 64-bit CPUs were rare, the IPv6 designers elected to optimize the IPv6 header for 64-bit processing. For this reason, I have drawn the IPv6 header 64 bits wide in Figure 1, a little different from the way it is usually depicted. Because 64-bit CPUs can read one 64-bit-wide memory word at a time, it is helpful that fields that are 64 bits (or a multiple of 64 bits) wide start at an even 64-bit boundary. Because every 64-bit boundary is also a 32-bit boundary, 32-bit CPUs aren't affected negatively by 64-bit optimization. The IPv4 header is presented in the usual form that highlights its 32-bit background.

Figure 1: The IPv4 and IPv6 headers



The fields in the IPv4 header that are not present in the IPv6 header have gray text; the field that is present in IPv6 but not in IPv4 is shown in italic. The changes from IPv4 to IPv6 follow:

- *Version* now always contains 6 rather than 4.
- The *Internet Header Length* (IHL) field that indicates the length of the IPv4 header is no longer needed because the IPv6 header is always 40 bytes long.
- *Type of Service* is now *Traffic Class*. The original semantics of the IPv4 Type of Service field have been superseded by the *diffserv* semantics per RFC 2474[3]. However, in IPv4, both interpretations of the field are in use (although most routers either cannot or are not configured to look at the field anyway). The IPv6 RFCs do not mandate a specific way to use the Traffic Class field, but generally the RFC 2474 diffserv interpretation is assumed.
- The *Flow Label* is new in IPv6. The idea is that packets belonging to the same stream, session, or flow share a common flow label value, making the session easily recognizable without having to look "deep" into the packet. Recognizing a stream or session is often useful in *Quality of Service* mechanisms. Although few implementations actually look at the flow label, most systems do set different flow labels for packets belonging to different TCP sessions. A zero value in this field means that setting a flow label per session is either not supported or not desired.

- The *Total Length* is the length of the IPv4 packet including the header, but in IPv6, the *Payload Length* does not include the 40-byte IPv6 header, thereby saving the host or router receiving a packet from having to check whether the packet is large enough to hold the IP header in the first place—making for a small efficiency gain. Despite the name, the Payload Length field includes the length of any additional headers, not just the length of the user data.
- The *Identification, Flags,* and *Fragment Offset* fields are used when IPv4 packets must be fragmented. Fragmentation in IPv6 works very differently (explained later), so these fields are relegated to a header of their own.
- *Time to Live* (TTL) is now called *Hop Limit*. This field is initialized with a suitable value at the origin of a packet and decremented by each router along the way. When the field reaches zero, the packet is destroyed. This way, packets cannot circle the network forever when there are loops. Per RFC 791[4], the IPv4 TTL field should be decremented by the number of seconds that a packet is buffered in a router, but keeping track of how long packets are buffered is too difficult to implement, regardless of buffering time. The new name is a better description of what actually happens.
- The *Protocol* field in IPv4 is replaced by *Next Header* in IPv6. In both cases, the field indicates the type of header that follows the IPv4 or IPv6 header. In most cases, the value of this field would be 6 for TCP or 17 for the *User Datagram Protocol* (UDP). Because the IPv6 header has a fixed length, any options such as source routing or fragmentation must be implemented as additional headers that sit between the IPv6 header and the higher-layer protocol such as TCP, forming a "protocol chain."
- The IPv4 *Header Checksum* was removed in IPv6.
- The *Source Address* and *Destination Address* serve the same function in IPv6 as in IPv4, except that they are now four times as long at 128 bits.

All IPv6 hosts and routers are required to support a maximum packet size of at least 1280 bytes. For lower-layer protocols that cannot support a *Maximum Transmission Unit* (MTU) of 1280 bytes, the relevant "IPv6 over …" standard must have a mechanism to break up and reassemble IPv6 packets so that the minimum of 1280 bytes can be accommodated. In IPv4, the official minimum size is 68 bytes—too low to be workable.

Checksums

In IPv4, the IP header is protected by a header checksum, and higher-layer protocols generally also have a checksum. The checksum algorithm for the IPv4 header, *Internet Control Message Protocol* (ICMP), ICMPv6, TCP, and UDP is the same one's complement addition, except that in IPv4, UDP packets may forego checksumming and simply set the checksum field to zero. In IPv6, this practice is no longer allowed: UDP packets must have a valid checksum.

The TCP, UDP, and ICMPv6 checksums are computed over a "pseudoheader" and the TCP, UDP, or ICMPv6 header, and user data, respectively. The pseudoheader consists of the source and destination addresses, the upper-layer packet length, and the protocol number. Including this information in the checksum calculation ensures that TCP, UDP, or ICMPv6 do not process packets that were delivered incorrectly, for instance, because of a bit error in the IP header.

IPv6 no longer has a header checksum to protect the IP header, meaning that when a packet header is corrupted by transmission errors, the packet is very likely to be delivered incorrectly. However, higher-layer protocols should be able to detect these problems, so they are not fatal. Also, lower layers almost always employ a *Cyclic Redundancy Check* (CRC) to detect errors.

Extension Headers

To allow special processing along the way, IPv4 allows extension of the IP header with one or more options. These options are rarely used today, both because they do not really solve common problems and because packets with options cannot be processed in the "fast path," and many routers and firewalls block some or all options. Not unlike the checkout counters at a grocery store, many routers have several "paths" that packets may follow: a fast one, implemented in hardware or highly optimized software, that supports only the most common operations (no checks), and one or more slower paths that use more advanced but slower software code that supports less common operations such as looking at IP options. However, many modern routers have only a fast path, so using additional features does not lead to a performance penalty.

Because the header is of fixed length in IPv6, options cannot be tagged onto the IP header as in IPv4. Instead, they are put in a header of their own that sits between the IPv6 header and the TCP or UDP (or other higher-level protocol) header. The most common extension headers follow:

- *Hop-by-Hop Options:* See the section that follows.
- *Routing:* This header is similar to the *Source Route* option in IPv4.
- *Fragment:* This header is used for fragmentation; see later in this article.
- *Authentication:* This header authenticates the user data and most header fields.
- *Encapsulating Security Payload* (ESP): This header encrypts or authenticates user data.
- *Destination Options:* See the section that follows.

The Hop-by-Hop Options and Destination Options headers are container headers: they have room for multiple suboptions. The Hop-by-Hop Options are processed by all routers along the way. All other options are normally ignored by routers and processed only by the destination. Obviously firewalls, or routers configured to perform filtering, may also look at these options. The Hop-by-Hop Options, Routing, Fragment, and Destination Options extension headers are defined in RFC 2460[5]. The Authentication and ESP extension headers are part of *IP Security* (IPsec).

Note that there is no standard extension header format, meaning that when a host encounters a header that it does not recognize in the protocol chain, the only thing it can do is discard the packet. Worse, firewalls and routers configured to filter IPv6 have the same problem: as soon as they encounter an unknown extension header, they must decide to allow or disallow the packet, even though another header deeper inside the packet would possibly trigger the opposite behavior. In other words, an IPv6 packet with a TCP payload that would normally be allowed through could be blocked if there is an unknown extension header between the IPv6 and TCP headers.

ICMPv6

The IPv6 version of the ICMP generally serves the same purposes as its IPv4 counterpart, but there are some changes. In IPv4, when a router or the destination host cannot process the packet properly, it sends back an ICMP error message along with the original IP header and the first 8 bytes of the higher-layer header. For UDP and TCP, this is enough for the source of the original host to see which TCP session or UDP association generated the offending packet. Because IPv6 supports an arbitrary number of extension headers between the IPv6 header and the higher-layer header, ICMPv6 returns as much of the original packet as will fit in the minimum MTU size of 1280 bytes. In addition to error messages, which are recognizable by an ICMP type of 127 or lower, there are also informational messages, with a type of 128 or higher. Because informational messages are not the result of an error, they do not include an original packet or part thereof. The most common ICMPv6 message types follow:

1: Destination unreachable
2: Packet too big
3: Time exceeded
4: Parameter problem
128: Echo request
129: Echo reply
130: Multicast listener query
131: Multicast listener report
132: Multicast listener done
133: Router solicitation
134: Router advertisement
135: Neighbor solicitation
136: Neighbor advertisement
137: Redirect message

ICMP and ICMPv6 messages also include a "code" that indicates the exact nature of the ICMP message within a certain type. As with ICMP, ICMPv6 calculates a checksum over the control message, but unlike ICMP, the ICMPv6 checksum calculation also includes a pseudoheader. Another departure from IPv4 is the fact that hosts and routers are required to limit the number of ICMPv6 messages they send. So if a router receives 100 packets per second toward an unreachable destination, it is not supposed to send back ICMPv6 packets at the same rate of 100 per second. The ICMPv6 redirect message works slightly different from the ICMP redirect message in IPv4. Like its IPv4 counterpart, the ICMPv6 redirect can be used by a router to inform a host that it should use a different router to reach the destination in question. But routers can also use the IPv6 Redirect to tell a host that the destination is reachable on the local subnet. Thus two hosts that have addresses in different prefixes can communicate directly after receiving redirects from a router.

Neighbor Discovery

When a system wants to send an IPv6 packet to another system connected to the same subnet or link, it needs to know what MAC address (or "link address" in the new IPv6 terminology) it should address the packet to, unless the interface in question is a point-to-point interface. Neighbor discovery allows systems to discover each other's MAC addresses, similar to *Address Resolution Protocol* (ARP) on Ethernet with IPv4.

Each IPv6 system joins the "solicited node" multicast group that corresponds to each of its addresses. Because the solicited node group address consists of the prefix `ff02:0:0:0:0:1:ff00::/104` followed by the bottom 24 bits of the address in question, addresses in different prefixes based on the same interface identifier (including the link-local address) all map to the same solicited node address.

Whenever a system needs to find out the link address for another system residing on the same link, it sends a neighbor solicitation to the solicited node address to which the IPv6 address of the remote system maps. The source host includes its own MAC address in the neighbor solicitation, so the neighbor knows where to send the reply.

Neighbor Unreachability Detection

RFC 2461[6] specifies a procedure for neighbor unreachability detection. IPv6 hosts and routers actively track whether their neighbors are reachable by periodically sending neighbor discovery messages directly to the neighbor. If the neighbor answers, it is reachable; if it does not, there must be some kind of problem, and the system discards the neighbor's MAC address and tries a regular multicast neighbor discovery procedure, allowing IPv6 systems to detect dead neighbors and neighbors that change their MAC address. But it is most useful to detect dead routers. On a subnet with more than one router, a host can simply install a default route toward another router when the router that it has been using becomes unreachable.

If a router loses its IPv6 address and no longer runs IPv6, Windows XP, Linux, MacOS, and FreeBSD all switch over to another router without incident. However, turning off the active router has much more severe effects: at the very least, ongoing downloads stall for a while, and in some cases, the session breaks. I have no explanation for this difference in behavior.

Stateless Address Autoconfiguration

Hosts and routers always configure link-local addresses on every interface on which IPv6 is enabled. The link-local address is nearly always derived from the interface MAC address, but to guarantee uniqueness, it is necessary to perform *Duplicate Address Detection,* which is discussed later.

When a host has a link-local address, it can obtain one or more global IPv6 addresses by using RFC 2462[7, 12], *Stateless Address Autoconfiguration*. IPv6 routers send out *Router Advertisement* (RA) packets (ICMPv6 type 134) periodically and in response to router solicitations. The information in RAs includes:

- An 8-bit *cur hop limit* field that tells hosts what value to use in the Hop Limit field of outgoing IPv6 packets
- The *managed address configuration* (M) flag—This flag is not well-defined, but the basic idea is that when it is set, hosts use a stateful mechanism (presumably *Dynamic Host Configuration Protocol Version 6* [DHCPv6]) to configure their addresses, and when the flag is not set, they use stateless address autoconfiguration.
- The *other stateful configuration* (O) flag—This flag is similar to the M flag, but indicates that the host should use a stateful mechanism to discover nonaddress configuration information.
- A 16-bit *router lifetime* value in seconds—This value tells hosts how long the default route that was created as the result of this RA should remain valid.
- The 32-bit *reachable time* value in milliseconds—This value indicates how long a neighbor should be considered reachable after receiving a "reachability confirmation," which is generally a neighbor advertisement but could be any packet.
- The 32-bit *retrans timer* value in milliseconds—The retrans timer tells hosts how long they should wait before retransmitting neighbor solicitation messages when there is no answer.

When fields that determine a value are set to zero, this means the value is not specified in the RA, so hosts must discover that value through other means. In addition to the preceding, router advertisements may also contain one or more options, such as:

- *Source link-layer address,* the router MAC address
- *MTU,* the maximum packet size that should be used on this subnet
- *Prefix information,* which specifies the prefixes used on the subnet and their properties

The prefix information option, in turn, has its own list of attributes:

- The *address prefix* itself and its length—For stateless address autoconfiguration to work, the prefix must be 64 bits long.
- The *on-link* flag—This flag tells hosts that the prefix is "on-link," so systems with addresses within this prefix are reachable on the subnet in question without help from a router.
- The *autonomous address configuration* flag—This flag tells hosts that they can create an address for themselves by combining this prefix with an interface identifier.
- A 32-bit *valid lifetime* in seconds—This value indicates how long the prefix should be considered on-link and how long autoconfigured addresses using the prefix can be used.
- A 32-bit *preferred lifetime* in seconds—This flag tells hosts how long autoconfigured addresses using this prefix are preferred.

Duplicate Address Detection

To avoid the situation where two IPv6 systems use the same address, systems perform *Duplicate Address Detection* for (nearly) all new IPv6 addresses before they are used. Duplicate address detection is done for global unicast addresses—and not just for those created using stateless address autoconfiguration, but also for link-local addresses. For obvious reasons, there is no duplicate detection for anycast addresses, because the whole point of anycast is that multiple systems have the same address.

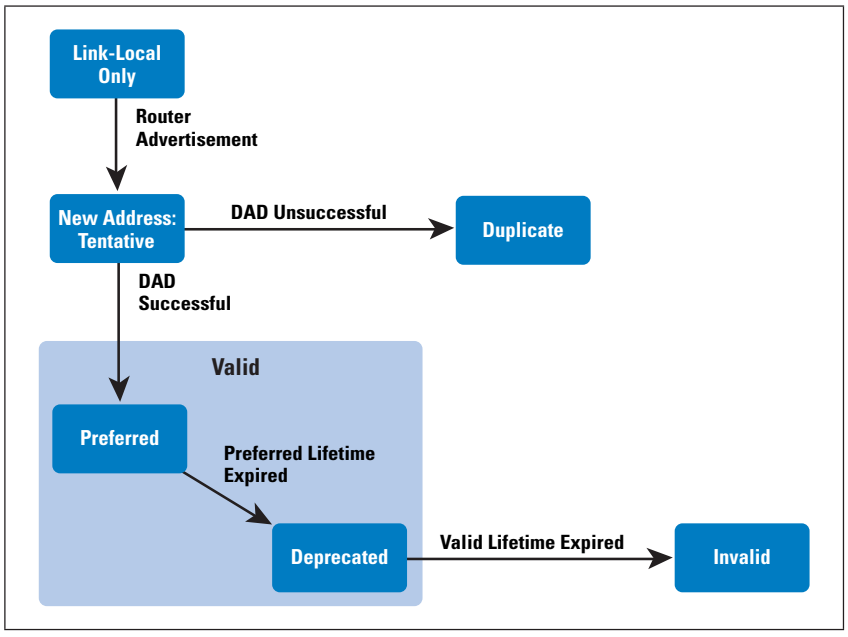


Figure 2: The Lifecycle of an IPv6 Address

As depicted in Figure 2, a host starts with only a link-local address. Duplicate address detection is also done for the link-local address, but this is not shown in the figure.

When a host receives a router advertisement that contains one or more prefixes with the autonomous address configuration flag set, the host creates addresses with interface identifiers derived from the IEEE *64-bit Extended Unique Identifier* (EUI-64) and possibly also a randomly generated one, if the host uses RFC 3041[8] address privacy. The host marks the resulting addresses as "tentative" and proceeds to execute the duplicate address detection procedure by joining the solicited node multicast group for the address in question and sending out one or more neighbor solicitation messages for the address. (If the number of duplicate address detection retries is configured to be zero, no duplicate detection is performed.) Only when there is no answer is the address used. If there is a conflict, the system is supposed to log the error and wait for manual intervention.

Address Lifetime

After successfully maneuvering past the duplicate address detection hurdle, addresses configured through stateless address autoconfiguration can be used until the "preferred lifetime" from the router advertisement message expires. In most cases, the lifetime does not expire because new RAs refresh the timers. But if there are no more RAs, eventually the preferred lifetime elapses and the address becomes "deprecated." New sessions should not use deprecated addresses but should choose "preferred" (nondeprecated) addresses, if available. However, existing sessions will continue to use the deprecated address. Eventually, the "valid lifetime" also runs out, and the deprecated address is removed from the interface, breaking any sessions that are still using the address.

Renumbering

Having different preferred and valid timers for the router advertisement itself and also for any prefixes contained in it makes it possible to do two things: renumber easily and cause more problems. It is even possible to do both at the same time. With stateless autoconfiguration, renumbering is easy: you simply give the router an address in the new prefix and set the preferred lifetime for the old prefix to zero, making hosts create one or more new addresses and deprecate any existing ones in the old prefix as soon as they receive the resulting router advertisement. After that, all new communication should start using the new address immediately. Existing TCP sessions and UDP associations continue to use the same address as before. After some time, all communication that started before the change should have stopped so that the old addresses can be removed safely.

This process is slightly more complex than it seems at first glance: as a precaution against attackers, hosts are not supposed to trust a valid lifetime of less than 7200. So make sure that the hosts have received at least one RA after setting the valid lifetime to 7200, and then set both the lifetimes to zero and remove the autonomous address configuration flag for the prefix. Two hours later, all hosts should have removed the addresses in this prefix, so you can remove the prefix from the router.

Beware that when you renumber because you are switching from one ISP to another, it is unavoidable that at some point, packets with source addresses in address space from ISP A end up at ISP B, or the other way around. If ISP B employs antispoofing or ingress filtering, it will not allow these packets through, so reduced connectivity will result. You can ask one ISP to remove the filters temporarily and then send out all your outgoing traffic over that ISP (or one that did not filter in the first place). However, do not expect too much cooperation from your ISP unless you are a valued customer.

Address Prefix and Router Lifetime Mismatch

Earlier, I mentioned the potential for causing more problems because router advertisements and the prefixes they contain have independent lifetimes. This scenario allows for four permutations:

- The RA lifetime is valid, and the prefix lifetime is valid: IPv6 works.
- The RA lifetime is invalid, and the prefix lifetime is invalid: IPv6 is disabled.
- The RA lifetime is valid, but the prefix lifetime is invalid: The system has an IPv6 default route but no global IPv6 address.
- The RA lifetime is invalid, but the prefix lifetime is valid: The system has a global IPv6 address but no IPv6 default route.

When a host has no global addresses but does have an IPv6 default route (case 3), it cannot reach the rest of the IPv6 Internet. Unfortunately, FreeBSD and MacOS hosts do not know that: they try anyway, with long delays as a result. Only after trying all the remote destination IPv6 addresses and timing out, the system falls back on IPv4 (for applications that try more than one address). Linux, on the other hand, does not install the IPv6 default route or ignores it when no global IPv6 addresses are present, so the timeout is immediate.

Windows XP does install the default route but magically manages to avoid lengthy timeouts anyway. On the other hand, Windows XP suffers timeouts when it has an IPv6 address but no default route (case 4) because Windows implements the on-link assumption: it first performs neighbor discovery on the local subnet for any IPv6 addresses. Only after neighbor discovery times out does Windows revert to IPv4. FreeBSD and MacOS, however, do not implement the on-link assumption, so they immediately notice that the IPv6 destination address is unreachable and revert to IPv4—if an IPv4 address is available and the application cycles through all addresses. With Linux, the default route does not seem to expire even though the timers eventually reach zero and lower. But addresses do expire and are removed when the lifetime for the associated prefix times out.

Address Selection

Choice is good, but it comes with problems of its own. The explicit support for multiple addresses in IPv6 requires the system or applications to choose which address to use for a given communication session. The coexistence of IPv4 and IPv6 in the same host makes this situation even more pressing. RFC 3484[9] provides guidelines in this area—it lists no fewer than 10 rules for choosing a destination address and 8 rules for selecting a source address. Most of these rules are fairly obvious, such as preferring a nondeprecated address over a deprecated one and not using a link-local source address to communicate with a destination that has a global address. It gets more interesting with the "policy table." On systems that support this mechanism, such as Windows XP and FreeBSD 5.4, the administrator can instruct the system to prefer certain address ranges over others.

Path MTU Discovery and Fragmentation

Because routers cannot fragment IPv6 packets, *Path MTU Discovery* (PMTUD) is mandatory in all cases where links with MTUs larger than 1280 bytes are used for IPv6, so it is imperative that routers generate ICMPv6 packet-too-big messages and that these messages make it back to the source of the offending packet. Filtering out these ICMPv6 messages makes it impossible to communicate reliably.

If you decide that you must filter ICMPv6 packet-too-big messages, you *must* use an MTU equal to the IPv6 mandatory minimum of 1280 bytes across your network so there is no need for PMTUD.

Upon reception of a packet-too-big message, TCP reduces its packet size to accommodate the smaller MTU on the path in question. However, protocols that run over UDP often cannot arbitrarily reduce their packet size. In IPv4, UDP packets are generally sent without the "don't fragment" bit set, so routers fragment them if necessary. In IPv6, this setup is not possible; if the packet is too large, the source host has to fragment it. The source host does this by first splitting the packet into unfragmentable and fragmentable parts. The IPv6 header and any headers that must be processed by routers along the way make up the unfragmentable part; the payload data and any headers that have to be processed only on the destination host are the fragmentable part. The fragmentable part is then split into as many parts as required to fit in the path MTU, and each part is transmitted as a packet containing the unfragmentable part, a fragment header, and one of the fragments of the fragmentable part.

The fragment header is 8 bytes, and except for a "next header" field and two reserved fields, it contains the same fragment offset, more fragments, and identification fields as the IPv4 header. The identification field is now 32 bits long and is used to indicate which fragments belong to the same original packet. All fragments except the last one have the "more-fragments" bit set and are multiples of 8 bytes.

After receiving the first fragment (which is not necessarily the first fragment of the original packet), a host waits up to 60 seconds for all other fragments to come in and, if they do, reassembles the original packet by combining all the fragments with the same source and destination addresses and identification field into a single packet. If one or more fragments is lost, the packet cannot be reassembled, so the entire packet is lost.

Note that IPv6 fragmentation has the same problem as IPv4 fragmentation: the TCP or UDP port numbers are available only in the first fragment, making it hard for firewalls and the like to filter fragmented packets. Common solutions are to reassemble the packet prior to filtering or to discard all fragments.

DHCPv6

DHCPv6 (RFC 3315[10]) is the IPv6 version of the DHCP. Because IPv6 has stateless address autoconfiguration, DHCP occupies a very different part of the landscape in IPv6 compared to IPv4. Although the details are different in the by-now-expected places (address length, use of multicasts, some streamlining), the DHCPv6 protocol itself is quite similar to the IPv4 version of DHCP. The more important differences are the way in which the protocol is used. DHCPv6 has three purposes:

- *Address configuration:* Giving out addresses to individual hosts
- *Nonaddress configuration:* Giving out other configuration information, such as DNS resolver addresses and domain search lists
- *Prefix delegation:* Giving out entire prefixes to routers (RFC 3633[11])

A DHCPv6 client interested in an address or other configuration information sends out a *solicit* message indicating its needs to the link-local scope multicast address **`ff02::1:2`**, port 547. (Server-to-client messages are addressed to port 546.) DHCPv6 servers that receive the *solicit* message either directly or forwarded by a relay and can accommodate the request respond with an *advertise* message. The client considers the offers in the various *advertise* messages and directs a *request* message to the server of its choice. The server then replies with a *reply* message, confirming the address or configuration information. Alternatively, if the client wants to receive only configuration information and no addresses or prefixes, it can send a *request-information* message, and the server immediately sends back a reply message, so only half the messages are exchanged and the whole process completes much faster. The client can also use the "rapid commit" option to indicate that it wants to use the expedited procedure for address or prefix assignment if it is fairly certain that it will take up the offer from the first DHCPv6 server that responds.

As expected, IPv6 addresses assigned with DHCPv6 come with a preferred and a valid lifetime. Sometime before this timer expires, the client sends a *renew* message, asking the server if it can continue to use the address. When it has no more use for the address, the client sends a *release* message. Less common situations have other messages.

To allow servers to recognize clients, each device that implements DHCPv6 has *DHCP Unique Identifier* (DUID). In IPv4, DHCP clients use a MAC address or user-supplied string as a Client Identifier. In DHCPv6 this client identifier is always the DUID. Devices can create their DUID in various ways, as long as the DUID is unique and not subject to change, if at all possible.

DHCPv6 supports an authentication mechanism that allows clients and servers to interact in a secure way, so third parties cannot inject false DHCP messages or modify legitimate ones. However, this mechanism must be preconfigured manually on all servers and clients, partially negating the advantages of DHCP over manual configuration.

An interesting use of DHCPv6 is *Prefix Delegation* (PD). With DHCPv6 PD, routers request a prefix that they then use to number one or more of their interfaces, supporting stateless address autoconfiguration for hosts connected to that interface. By creatively borrowing the DHCP timers and reusing them in router advertisements, a whole site can be renumbered by changing a single setting in a DHCPv6 configuration on a DHCPv6 server or a router functioning as a DHCPv6 PD server.

Ed.: This article is adapted from chapter 8 of *Running IPv6* by Iljitsch van Beijnum, published by Apress in 2005, ISBN 1590595270. The article differs from the chapter in that it has been edited for size and the vendor-specific examples have been removed. Used with permission. For information about the book, see:
http://www.apress.com/book/bookDisplay.html?bID=10026

References

[1] Karrenberg D., Ross G., Wilson P., and Nobile L., "Development of the Regional Internet Registry System," *The Internet Protocol Journal*, Volume 4, No. 4, December 2001.

[2] Carpenter, B., Fink, B., and Moore, K., "Connecting IPv6 Routing Domains Over the IPv4 Internet," *The Internet Protocol Journal*, Volume 3, No. 1, March 2000.

[3] Nichols, K., Blake, S., Baker, F., and Black, D., "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474, December 1998.

[4] Postel, J., "Internet Protocol," RFC 791, September 1981.

[5] Deering, S. and Hinden, R., "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1998.

[6] Narten, T., Nordmark, E., and Simpson, W., "Neighbor Discovery for IP Version 6 (IPv6)," RFC 2461, December 1998.

[7] Narten, T. and Thomson, S., "IPv6 Stateless Address Autoconfiguration," RFC 2462, December 1998.

[8] Narten, T. and Draves, R., "Privacy Extensions for Stateless Address Autoconfiguration in IPv6," RFC 3041, January 2001.

[9] Draves, R., "Default Address Selection for Internet Protocol Version 6 (IPv6)," RFC 3484, February 2003.

[10] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and Carney, M., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3315, July 2003.

[11] Troan, O. and Droms, R., "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) Version 6," RFC 3633, December 2003.

[12] François Donzé, "IPv6 Address Autoconfiguration," *The Internet Protocol Journal,* Volume 7, No. 2, June 2004.

ILJITSCH VAN BEIJNUM holds a Bachelor of Information and Communication Technology degree from the Haagse Hogeschool in The Hague, Netherlands. In 1995, he found himself in the emerging Internet Service Provider business. There he learned about system administration, IP networking, and especially routing. After first starting a small ISP with four others and working as a senior network engineer for UUNET Netherlands, he became a freelance consultant in 2000. Not long after that, he started contributing to the IETF Multihoming in IPv6 working group. He wrote the book *BGP: Building Reliable Networks with the Border Gateway Protocol,* ISBN 0-596-00254-8, published by O'Reilly in 2002, and *Running IPv6,* ISBN 1590595270, published by Apress in 2005. E-mail: **iljitsch@muada.com**

Book Reviews

Electronic Brains

Electronic Brains, Stories from the Dawn of the Computer Age, by Mike Hally, ISBN 0-309-09630-8, Joseph Henry Press, 2005.

Electronic Brains is a personal account from the early days of computing that describes the childhood of a technology that is little more than 50 years old. The book originated as a BBC radio programme, still accessible at **`http://www.bbc.co.uk/radio4/science/electronicbrains.shtml`**. Mike Hally traveled over the globe looking for the first "computors" and the stories from the dawn of a new age. This book contains the results of the investigation, giving a first-hand testimony of hard work, passion, and amazing developments that shaped the second half of the last century.

Organization

Chapter 1, "From ABC to ENIAC," presents the development of what is commonly accepted as the first computer, the ENIAC, a computer that replaced calculating machines and people making the operations in ballistic trajectories analysis by hand. Credit is given to John Atanassof and Clifford Berry, the developers of ABC, possibly the first operational computer in the world.

Development of the UNIVAC, the computer famed by predicting the result of the 1952 U.S. presidential election, is presented in Chapter 2. Designed by Eckert and Mauchly, the developers of ENIAC, UNIVACs were commercial computers used for processing census data and so well marketed that the term "UNIVAC" was used as a synonym for "computer."

Chapter 3 looks at the development of the Rand 409, maybe the first mass-produced computer. The 409 was a medium-sized computer, with a price tag of US$100,000 that compared favorably against UNIVAC's $1 million, achieving a sell rate of one per week.

"Computing in Great Britain" is the focus of Chapter 4, where credit is given to Maurice Wilkes and Alan Turing. A worthy detail that gives a glimpse of the technical difficulties overcome is the description of memory based on mercury delay-lines, where binary data was stored using sound pulses on tubes filled with mercury engineered in such way that the delay from transmitter to receiver allowed the electronics to do the calculation before the data in memory was needed at the receiver side.

Perhaps the strangest computer development is set forth in Chapter 5. The *Lyons Electronics Office* (LEO) was a computer developed by a large catering company to expedite its clerical operations. LEO was possibly the first commercial computer in the world, so successful that the catering company began to produce and sell it to other corporations.

Chapter 6 describes the efforts by USSR scientists to develop computing technology. More than one development was made; it is not clear which was the first soviet computer, and the developments were secret—in some cases very specialized, such as a computer with ternary logic instead of the currently used binary logic (ENIAC used decimal logic).

Chapter 7 focuses on computing developments in Australia, work that did not last because the funds were scarce and sometimes the budget was assigned to other sciences, such as radiophysics. Here we can see that computers were used for purposes totally different than their uses in cold-war countries; for example, they were used to answer crossword puzzles—strange if we consider that the disk had a capacity of 3 KB.

A strange computer, formally known as *Hydraulic Economics Computer,* is described in Chapter 8. It was not a typical computer—it was a system developed to show the interrelation between macroeconomic variables using colored water, pumps, and valves. Universities, central banks, and Ford bought the computer, and four of them survive in different parts of the world. The emergence of IBM is the subject of Chapter 9, which presents IBM as a late adopter of computing technology that eventually became the leader of the computer age. We learn that the first computer produced by IBM was the IBM 701; after that came the IBM 1401 and then the IBM 360—the system that consolidated IBM as the ruler in the computing world.

Summary

From the ABC to the well-known ENIAC and UNIVAC, *Electronic Brains* is a testimony to the people who worked day and night to accomplish something that few others understood. Motivated mainly by passion and with little to no economic support, team spirit is a common factor in all the computer developments: "...it was like a brotherhood! We would help each other in case someone got stuck on a particular activity. I would have gone anywhere with those guys. I've never had such unified job environment. We knew we were pushing back the frontiers."

Electronic Brains is an enjoyable book that I recommend to any person with interest in computers and technology. Computer historians could scoff at the rather simple analysis of technical details, but this is not a technical book. The value of *Electronic Brains* is the first-hand account of early undertakings and the multiple-country investigation that is presented. With many anecdotes, this book will serve as a witness to the pioneers of a new era, the computing era.

—*Claudio Gutiérrez*
claudio.gutierrez.m@gmail.com

Business 2010

Business 2010—Mapping the Commercial Landscape, by Ian Pearson and Michael Lyons, ISBN 1-84439-105-1, Published by Spiro Press, **http://www.spiropress.com/**

This interesting book explores how trends in technology, economic factors, social changes, and evolving attitudes to technology will reshape the business landscape by the year 2010. The book describes its subject matter in terms that are understandable and interesting to both technical and nontechnical audiences. It is valuable to technologists because it expands their perception of the future beyond that which is available through traditional sources such as vendor roadmap sessions by linking closely commercial, technical, and social trends.

Organisation

The book is divided into three main sections. The first looks at the major influences on future business: technological progress, changing attitudes, social forces, and economics. The implications of these factors are then examined, and finally the application of the analysis to business strategy is examined. These ideas are then pulled together in a succinct and easily understood conclusion.

Pearson and Lyons focus on the effect of particular techniques. Some of these, such as self-organising systems and the mimicking of natural phenomena ("biomimetrics"), are fairly unconventional, but others, such as increased miniaturisation, wireless devices, low-cost computing and networking, the semantic Web, and artificial intelligence will be more familiar. The Internet and its potential effect on financial transactions and taxation features heavily. The authors note that attitudes to technology are changing and adoption cycles are reducing, describing the impact that technology has had on the physical labour market and the likely future impact on knowledge workers. The authors consider the economic implications of the exploitation of information, looking at the relative cost of creation and reproduction when compared with more traditional goods and services.

The next three chapters look at the implications of this analysis, starting by looking at numerous trade-offs and counter-balancing forces, such as the effect of the "browser wars" and the relationships between customers and producers. The importance of customer and worker information to a commercial organisation and the problems arising from its exploitation are described. The discussion then considers how the knowledge economy changes the importance of physical assets and commercial relationships, followed by an examination of the political and organisational implications of technology.

Finally the authors look at the business effects, starting with the ease of transferring information between systems. They note that corporate intranets make both the devolving of authority through outsourcing and the imposition of increased command and control through micromanagement easier.

Pearson and Lyons suggest that new technology alters the value chains that influence businesses, leading to more temporary business relationships, their replacement by "value-nets," and the rise of the virtual company. This section concludes by looking at globalisation—how goods and services are paid for and some of the implications for taxation.

The authors ask the question—how can business adapt? They start their analysis by examining the interactions between the physical and mental worlds and cyberspace, noting that a strategic analysis works only if the forces acting on a business do not change too rapidly. As change becomes more rapid, there will be no time to develop business cases, because first-mover advantage will be the only advantage a business can have. Pearson and Lyons conclude that the critical factors in allowing cyber-economy to grow are ease of navigation and the effective use of branding. They conclude by examining who will be the winners and losers in business in the year 2010—and why.

Synopsis

This book is succinct and well-written, covering a complex but interesting field in just under 200 pages. The authors paint a convincing description of future business trends, exploring the technical, commercial, economic, and political pressures that will influence them. Their cause, effect, and potential response treatment leads the reader through the subject in a way that is both interesting and instructive. The authors are not afraid to be controversial and at times they take the reader into some very unfamiliar territory, adding extra spice to the book.

While other books are available that look at the future from a more technologically orientated perspective, this book is one of the few that manages to couple the developments in the commercial and technical worlds, thereby giving a more comprehensive viewpoint. In an age when technologists are increasingly being asked to take more of a commercial view, this can only be a good thing. The approach taken has much in common with that taken by Alvin Toffler in his books *Future Shock* and *The Third Wave*. An updated treatment like this is to be welcomed.

The Authors

Ian Pearson works for British Telecom (BT) as its chief futurologist; he is a well-known speaker on future technology trends and has published extensively in this field. Michael Lyons also works for BT and has more than 30 years of research experience in the telecoms industry. He has recently been working in the fields of decision support systems and long-term research issues, leading a research team in BT's Research and Venturing department. Pearson is described as an "unfettered thinker" and Lyons as a "pragmatic modeller," characteristics which give the book its balanced view.

—Edward Smith, BT, UK
edward.a.smith@btinternet.com

Fragments

ICANN Ratifies Global Policy for Allocation of IPv6 Address Space

On September 7, 2006, the ICANN Board ratified the *Global Policy for Allocation of IPv6 Address Space*. This policy provides for the allocation of IPv6 address space from ICANN to the *Regional Internet Registries* (RIRs).

On July 13, 2006, the Secretary of the *Address Supporting Organization* (ASO) *Address Council* (AC) forwarded to ICANN the proposed global policy for allocation of IPv6 address space. This proposed global policy had been submitted to the ASO AC by the Executive Council of the *Number Resource Organization* (NRO) on June 6, 2006, and adopted by the ASO AC on July 12, 2006. Each RIR community individually discussed the policy and approved its adoption via their own policy development processes. The IPv6 Allocation Policy document is available from the ASO Website:

http://aso.icann.org/docs/aso-global-ipv6.pdf

See also:

http://www.icann.org/announcements/announcement-11sep06.htm

http://www.nro.net

IP addressing in China and the Myth of Address Shortage

In recent years, various sources have repeated a myth that the IPv4 address pool is close to exhaustion. Many of these stories also falsely claim that there are fewer IPv4 addresses allocated to China than to some individual US universities. The *Asia Pacific Network Information Centre* (APNIC) is committed to countering this myth and has published an article in its newsletter *Apster* on this topic. The article is available here:

http://www.apnic.net/news/hot-topics/internet-gov/ip-china.html

Calendar of Internet-related Events

The *Internet Society* (ISOC) maintains an online list of meetings and conferences, see:

http://geneva.isoc.org/events/

Don't forget to tell us if you move!

We receive quite a lot of IPJ return mail marked as "undeliverable." If you change your address please let us know by either using the IPJ subscription tool or sending an e-mail with the new information to **ipj@cisco.com**. Your cooperation is much appreciated.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

The Internet Protocol Journal
Ole J. Jacobsen, Editor and Publisher

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

December 2006 Volume 9, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

Internet security and stability are topics we keep returning to in this journal. So far we have mainly focused on technologies that protect systems from unauthorized access and ensure that data in transit over wired or wireless networks cannot be intercepted. We have discussed security-enhanced versions of many of the Internet core protocols, including the *Border Gateway Protocol* (BGP), *Simple Network Management Protocol* (SNMP), and the *Domain Name System* (DNS). You can find all these articles by visiting our Website and referring to our index files. All back issues continue to be available in both HTML and PDF formats. In this issue, Wesley Eddy explains a vulnerability in the *Transmission Control Protocol* (TCP) in which a sender can overwhelm a receiver by sending a large number of SYN protocol exchanges. This form of *Denial of Service* attack, known as *SYN Flooding*, was first reported in 1996, and researchers have developed several solutions to combat the problem.

Speaking of Internet stability, at 12:26 GMT on December 26, 2006, an earthquake of magnitude 6.7 struck off Taiwan's southern coast. Six submarine cables were damaged, resulting in widespread disruption of Internet service in parts of Asia. We hope to bring you more details and analysis of this event in a future issue of IPJ. The topic will also be discussed at the next *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT), which will take place in Bali, Indonesia, February 21 through March 2, 2007. For details see: **http://www.apricot2007.net**

The design and operation of systems that use Internet protocols for communication in conjunction with advanced applications—such as an e-commerce system—require the use of a certain amount of "middleware." This software, largely hidden from the end user, has been the subject of a great deal of development and standardization work for several decades. An important component of today's Web systems is the *Extensible Markup Language* (XML). Silvano Da Ros explains how XML networking can be used as a critical building block for network application interoperability.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Defenses Against TCP SYN Flooding Attacks

by Wesley M. Eddy, Verizon Federal Network Systems

This article discusses a specific *Denial of Service* (DoS) attack known as *TCP SYN Flooding*. The attack exploits an implementation characteristic of the *Transmission Control Protocol* (TCP), and can be used to make server processes incapable of answering a legitimate client application's requests for new TCP connections. Any service that binds to and listens on a TCP socket is potentially vulnerable to TCP SYN flooding attacks. Because this includes popular server applications for e-mail, Web, and file storage services, understanding and knowing how to protect against these attacks is a critical part of practical network engineering.

The attack has been well-known for a decade, and variations of it are still seen. Although effective techniques exist to combat SYN flooding, no single standard remedy for TCP implementations has emerged. Varied solutions can be found among current operating systems and equipment, with differing implications for both the applications and networks under defense. This article describes the attack and why it works, and follows with an overview and assessment of the current tactics that are used in both end hosts and network devices to combat SYN flooding attacks.

Basic Vulnerability

The SYN flooding attack became well-known in 1996, when the magazines *2600* and *Phrack* published descriptions of the attack along with source code to perform it[1]. This information was quickly used in attacks on an Internet service provider's (ISP's) mail and Telnet servers, causing outages that were widely publicized in *The Washington Post* and *The Wall Street Journal* (among other venues). CERT quickly released an advisory on the attack technique[2].

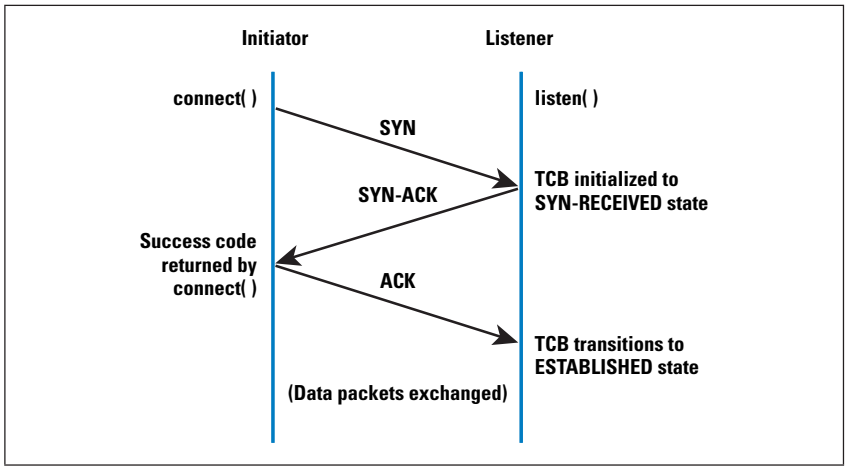


Figure 1: Normal TCP 3-Way Handshake

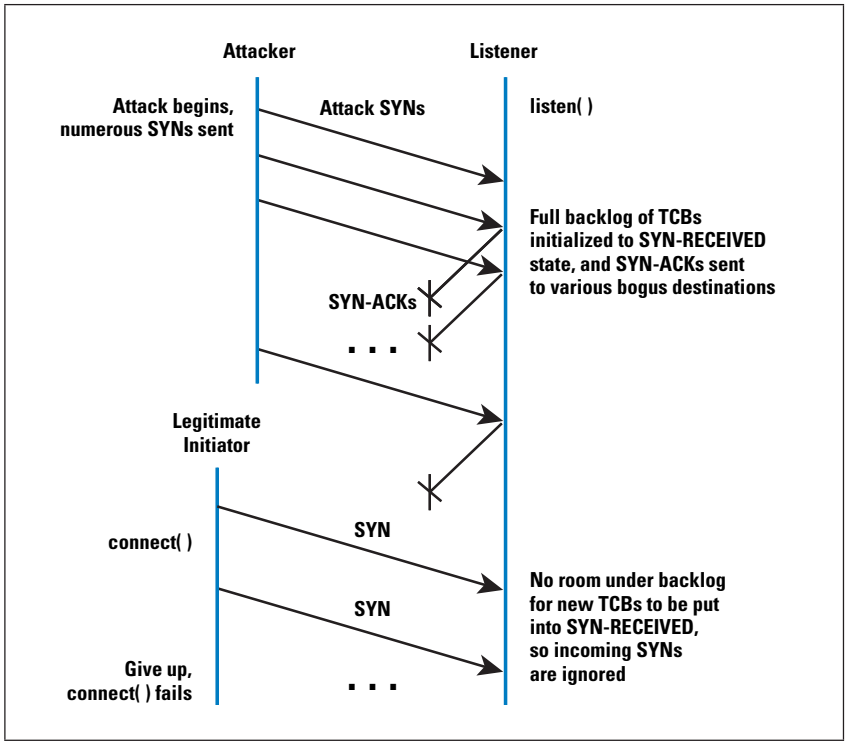
The basis of the SYN flooding attack lies in the design of the 3-way handshake that begins a TCP connection. In this handshake, the third packet verifies the initiator's ability to receive packets at the IP address it used as the source in its initial request, or its return reachability. Figure 1 shows the sequence of packets exchanged at the beginning of a normal TCP connection (refer to RFC 793 for a detailed description of this process).

The *Transmission Control Block* (TCB) is a transport protocol data structure (actually a set of structures in many operations systems) that holds all the information about a connection. The memory footprint of a single TCB depends on what TCP options and other features an implementation provides and has enabled for a connection. Usually, each TCB exceeds at least 280 bytes, and in some operating systems currently takes more than 1300 bytes. The TCP SYN-RECEIVED state is used to indicate that the connection is only half open, and that the legitimacy of the request is still in question. The important aspect to note is that the TCB is allocated based on reception of the SYN packet—before the connection is fully established or the initiator's return reachability has been verified.

This situation leads to a clear potential DoS attack where incoming SYNs cause the allocation of so many TCBs that a host's kernel memory is exhausted. In order to avoid this memory exhaustion, operating systems generally associate a "backlog" parameter with a listening socket that sets a cap on the number of TCBs simultaneously in the SYN-RECEIVED state. Although this action protects a host's available memory resource from attack, the backlog *itself* represents another (smaller) resource vulnerable to attack. With no room left in the backlog, it is impossible to service new connection requests until some TCBs can be reaped or otherwise removed from the SYN-RECEIVED state.

Depleting the backlog is the goal of the TCP SYN flooding attack, which attempts to send enough SYN segments to fill the entire backlog. The attacker uses source IP addresses in the SYNs that are not likely to trigger any response that would free the TCBs from the SYN-RECEIVED state. Because TCP attempts to be reliable, the target host keeps its TCBs stuck in SYN-RECEIVED for a relatively long time before giving up on the half connection and reaping them. In the meantime, service is denied to the application process on the listener for legitimate new TCP connection initiation requests. Figure 2 presents a simplification of the sequence of events involved in a TCP SYN flooding attack.

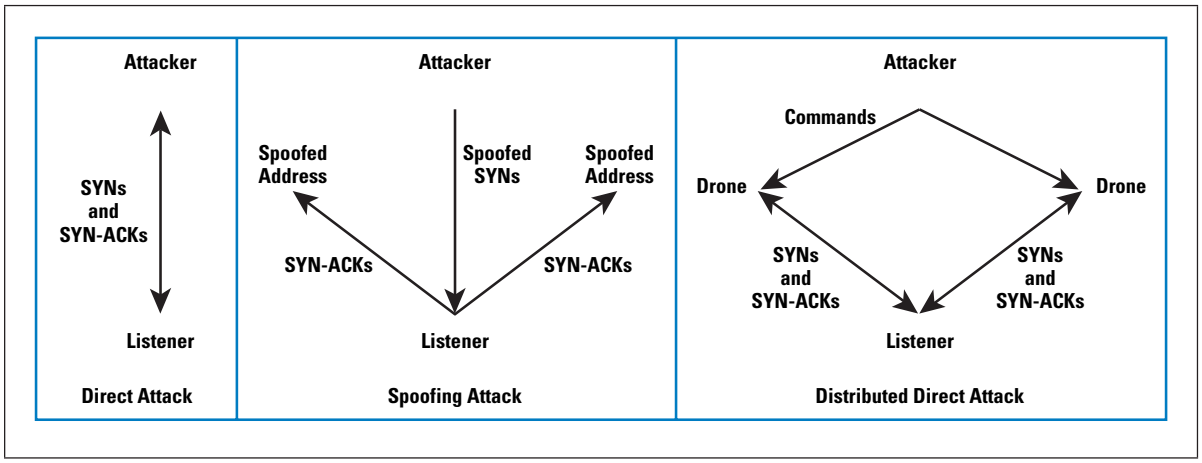
Figure 2: Attack Demonstration: Enough illegitimate TCBs are in SYN-RECEIVED that a legitimate connection cannot be initiated.



Attack Methods

The scenario pictured in Figure 2 is a simplification of how SYN flooding attacks are carried out in the real world, and is intended only to give an understanding of the basic idea behind these types of attacks. Figure 3 presents some variations that have been observed on the Internet.

Figure 3: Some Variants of the Basic Attack



Direct Attack

If attackers rapidly send SYN segments without spoofing their IP source address, we call this a *direct attack*. This method of attack is very easy to perform because it does not involve directly injecting or spoofing packets below the user level of the attacker's operating system. It can be performed by simply using many TCP *connect()* calls, for instance. To be effective, however, attackers must prevent their operating system from responding to the SYN-ACKS in any way, because any ACKs, RSTs, or *Internet Control Message Protocol* (ICMP) messages will allow the listener to move the TCB out of SYN-RECEIVED. This scenario can be accomplished through firewall rules that either filter outgoing packets to the listener (allowing only SYNs out), or filter incoming packets so that any SYN-ACKS are discarded before reaching the local TCP processing code.

When detected, this type of attack is very easy to defend against, because a simple firewall rule to block packets with the attacker's source IP address is all that is needed. This defense behavior can be automated, and such functions are available in off-the-shelf reactive firewalls.

Spoofing-Based Attacks

Another form of SYN flooding attacks uses IP address spoofing, which might be considered more complex than the method used in a direct attack, in that instead of merely manipulating local firewall rules, the attacker also needs to be able to form and inject raw IP packets with valid IP and TCP headers. Today, popular libraries exist to aid with raw packet formation and injection, so attacks based on spoofing are actually fairly easy.

For spoofing attacks, a primary consideration is address selection. If the attack is to succeed, the machines at the spoofed source addresses must not respond to the SYN-ACKS that are sent to them in any way. A very simple attacker might spoof only a single source address that it knows will not respond to the SYN-ACKS, either because no machine physically exists at the address presently, or because of some other property of the address or network configuration. Another option is to spoof many different source addresses, under the assumption that some percentage of the spoofed addresses will be unrespondent to the SYN-ACKS. This option is accomplished either by cycling through a list of source addresses that are known to be desirable for the purpose, or by generating addresses inside a subnet with similar properties.

If only a single source address is repetitively spoofed, this address is easy for the listener to detect and filter. In most cases a larger list of source addresses is used to make defense more difficult. In this case, the best defense is to block the spoofed packets as close to their source as possible.

Assuming the attacker is based in a "stub" location in the network (rather than within a transit *Autonomous System* (AS), for instance), restrictive network ingress filtering[7] by stub ISPs and egress filtering within the attacker's network will shut down spoofing attacks—if these mechanisms can be deployed in the right places. Because these ingress/egress filtering defenses may interfere with some legitimate traffic, such as the Mobile IP triangle routing mode of operation, they might be seen as undesirable, and are not universally deployed. *IP Security* (IPsec) also provides an excellent defense against spoofed packets, but this protocol generally cannot be required because its deployment is currently limited. Because it is usually impossible for the listener to ask the initiator's ISPs to perform address filtering or to ask the initiator to use IPsec, defending against spoofing attacks that use multiple addresses requires more complex solutions that are discussed later in this article.

Distributed Attacks

The real limitation of single-attacker spoofing-based attacks is that if the packets can somehow be traced back to their true source, the attacker can be easily shut down. Although the tracing process typically involves some amount of time and coordination between ISPs, it is not impossible. A distributed version of the SYN flooding attack, in which the attacker takes advantage of numerous drone machines throughout the Internet, is much more difficult to stop. In the case shown in Figure 3, the drones use direct attacks, but to increase the effectiveness even further, each drone could use a spoofing attack and multiple spoofed addresses.

Currently, distributed attacks are feasible because there are several "botnets" or "drone armies" of thousands of compromised machines that are used by criminals for DoS attacks. Because drone machines are constantly added or removed from the armies and can change their IP addresses or connectivity, it is quite challenging to block these attacks.

Attack Parameters

Regardless of the method of attack, SYN flooding can be tuned to use fewer packets than a brute-force DoS attack that simply clogs the target network by sending a high volume of packets. This tuning is accomplished with some knowledge of the listener's operating system, such as the size of the backlog that is used, and how long it keeps TCBs in SYN-RECEIVED before timing out and reaping them. For instance, the attacker can minimally send a quick flight of some number of SYNs exactly equal to the backlog, and repeat this process periodically as TCBs are reclaimed in order to keep a listener unavailable perpetually.

Default backlogs of 1024 are configured on some recent operating systems, but many machines on the Internet are configured with backlogs of 128 or fewer. A common threshold for retransmission of the SYN-ACK is 5, with the timeout between successive attempts doubled, and an initial timeout of 3 seconds, yielding 189 seconds between the time when the first SYN-ACK is sent and the time when the TCB can be reclaimed.

Assuming a backlog of 128 and that an attacker generates 40-byte SYN segments (with a 20-byte TCP header plus a 20-byte IP header), the attacker has to send only 5.12 kilobytes (at the IP layer) in order to fill the backlog. Repeated every 189 seconds, this process gives an average data rate of only 27 bytes per second (easily achievable even over dialup links). This data rate is in stark contrast to DoS attacks that rely on sending many megabits per second of attack traffic. Even if a backlog of 2048 is used, the required data rate is only 433 bytes per second, so it is clear that the ease of attack scales along with increases to the backlog—and more sophisticated defenses are needed.

Lessons Learned

The protocol flaw in TCP that makes SYN flooding effective is that for the small cost of sending a packet, an initiator causes a relatively greater expense to the listener by forcing the listener to reserve state in a TCB. An excellent technique for designing protocols that are robust to this type of attack is to make the listener side operate statelessly[3] until the initiator can demonstrate its legitimacy. This principle has been used in more recent transport protocols, such as the *Stream Control Transmission Protocol* (SCTP)[4], which has a 4-way handshake, with listener TCB state being created only after the initiator echoes back some "cookie" bytes sent to it by the listener. This echo proves to some extent that the initiator side is at the address it appears to be (that is, it has return reachability) and is not attempting a SYN flooding style of attack.

Outside of transport protocols and TCBs, security protocols also commonly use this defense technique. For instance, the *Internet Key Exchange Version 2* (IKEv2)[5] component of IPsec does not create state for a new *Security Association* until it can verify that initiators are capable of responding to packets sent to the address they claims to be using. There are other security protocols in which the listener sends out "puzzles" in response to initiation attempts and grants services or state only when puzzle solutions are returned[6]. This tactic not only verifies the addresses of initiators but also implies a computational burden that causes them to further demonstrate their genuine willingness to communicate productively.

Countermeasures

During the initial Panix attack, random spoofed source addresses were being used, but it was noted that the attack TCP SYNs all used the same source port number. A filter that denied incoming packets from this port was temporarily effective, but easy for the attacker to adapt to, and the attack segments began using random ports. Panix was able to isolate which of its ingress routers the attack was coming from and null-route packets destined for its servers coming through that router, but this solution was obviously a heavy-handed one, and seems to have also been overcome when the attacker started sending packets that were routed through a different upstream provider. Panix had mixed success in getting its providers to assist in tracing and blocking the attack, and the networking community was spurred into devising other solutions.

Two broad classes of solutions to SYN flooding attacks have evolved, corresponding to where the defenses are implemented. The first class of solutions involves hardening the end-host TCP implementation itself, including altering the algorithms and data structures used for connection lookup and establishment, as well as some solutions that diverge from the TCP state machine behavior during connection establishment, as described in RFC 793.

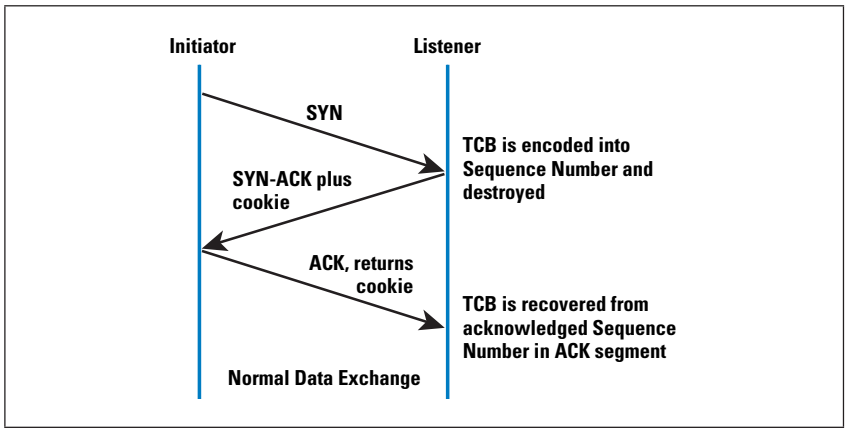
The second class involves hardening the network, either to lessen the likelihood of the attack preconditions (an army of controlled hosts or the propagation of IP packets with spoofed source addresses), or to insert middleboxes that can isolate servers on the networks behind them from illegitimate SYNs.

End-Host Countermeasures

Increasing TCP Backlog: Because the basic attack mechanism relies on overflowing a host's backlog of connecting sockets, an obvious end host-based solution is to simply increase the backlog, as is already done for very popular server applications. In at least some popular TCP implementations, this solution is known to be a poor one because of the use of linear list traversal in the functions that attempt to free state associated with stale connection attempts. Increasing the backlog is typically possible through altering the *listen()* call of an application and setting an operating system kernel parameter named SOMAXCONN, which sets an upper bound on the size of the backlog that an application can request. This step by itself should not be seriously considered as a means to defend against SYN flooding attacks—even in operating systems that can efficiently support large backlogs—because an attacker who can generate attack segments will most likely be able to scale to larger orders than the backlog supportable by a host.

Reducing the SYN-RECEIVED *Timer:* Another simple end host-based mechanism is to put a tighter limit on the amount of time between when a TCB enters the SYN-RECEIVED state and when it may be reaped for not advancing. The obvious disadvantage to this mechanism is that in cases of aggressive attacks that impose some amount of congestion loss in either the SYN-ACK or handshake-completing ACK packets, legitimate connection TCBs may be reaped as hosts are in the process of retransmitting these segments. Furthermore, there is only a linear relationship between the reduction that an administrator makes in the SYN-RECEIVED timer and the corresponding increase in packet rate that the adversary must make in order to continue attacking the server. Other alternative end-host solutions make it much more difficult for an attack to remain viable. For these reasons, a reduction in the SYN-RECEIVED timer is not an advisable defense against SYN flooding attacks.

Figure 4: Connection Establishment with SYN Cookies



SYN *Caches:* Two end-host defenses, called SYN caches and SYN cookies (described later), operate by reducing the amount of state allocated initially for a TCB generated by a received SYN, and putting off instantiating the full state[8]. In a host that uses a SYN cache, a hash table with a limited amount of space in each hash bucket is used to store a subset of the data that would normally go into an allocated TCB. If and when a handshake completing ACK is received, this data can be moved into a full TCB; otherwise the oldest bucket at a particular hash value can be reaped when needed. In Lemon's FreeBSD example[8], the SYN cache entry for a half connection is 160 bytes, versus 736 bytes for a full TCB, and 15359 entries in the SYN cache are supported.

The SYN cache data structure is robust to attackers attempting to overflow its buckets because it uses the initiator's local port number and some secret bits in the hash value. Because stacks are a more effective data structure to search than a simple linked list, stacks that use a SYN cache can have improved speed, even when not under attack. Under Lemon's tests, during an active attack a host using a SYN cache was able to establish legitimate connections with only about a 15-percent increase in latency.

SYN Cookies: In contrast to the SYN cache approach, the SYN cookies technique causes absolutely zero state to be generated by a received SYN. Instead, the most basic data comprising the connection state is compressed into the bits of the sequence number used in the SYN-ACK. Since for a legitimate connection, an ACK segment will be received that echoes this sequence number (actually the sequence number plus one), the basic TCB data can be regenerated and a full TCB can safely be instantiated by decompressing the Acknowledgement field. This decompression can be effective even under heavy attack because there is no storage load whatsoever on the listener, only a computational load to encode data into the SYN-ACK sequence numbers. The downside is that not all TCB data can fit into the 32-bit Sequence Number field, so some TCP options required for high performance might be disabled. Another problem is that SYN-ACKs are not retransmitted (because retransmission would require state), altering the TCP synchronization procedures from RFC 793.

Recent work by Andre Oppermann uses the TCP Timestamp option in conjunction with the Sequence Number field to encode more state information and preserve the use of high-performance options such as TCP Window Scaling, and TCP *Selective Acknowledgment Options* (SACK), and can also be used to preserve *TCP-Message Digest 5* (MD5) support with SYN cookies. This option is a step forward, in that it removes the major negative effect of previous SYN cookie implementations that disabled these features.

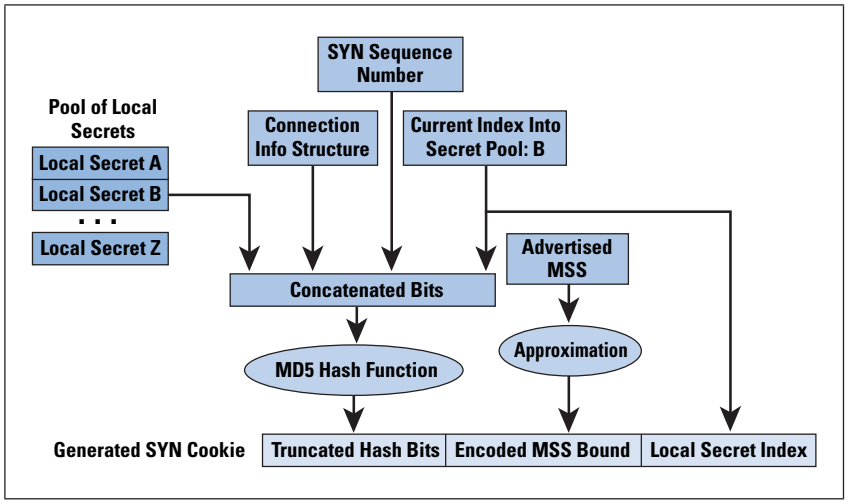


Figure 5: Process for Generation and Validation of TCP SYN Cookies.

The exact format of TCP SYN cookies is not an interoperability issue, because they are only locally interpreted, and the format and procedures for generation and validation can vary slightly among implementations. Figure 5 depicts the general process of SYN cookie generation and validation used by multiple implementations.

To compute the SYN-ACK sequence number (that is, the TCP cookie) when using TCP cookies, a host first concatenates some local secret bits, a data structure that contains the IP addresses and TCP ports, the initial SYN sequence number, and some index data identifying the secret bits. An MD5 digest is computed over all these bytes, and some bits are truncated from the hash value to be placed in the SYN-ACK sequence number. Because the sequence number is about a fourth the size of the full hash value, this truncation is necessary, but generally at least 3 bytes worth of the hash bits are used, meaning that there should still be close to a 2^{24} effort required to guess a valid cookie without knowing the local secret bits. In addition to the hash output, some of the cookie bits indicate a lower bound on the *Maximum Segment Size* (MSS) that the SYN contained, and the index bits identifying the local secret used within the hash.

To validate a SYN cookie, first the acknowledgement number in an incoming ACK segment is decremented by 1 to retrieve the generated SYN cookie. The valid value for the set of truncated hash bits is computed based on the IP address pair, TCP port numbers, segment sequence number minus one, and the value from the secret pool corresponding to the index bits inside the cookie. If these computed hash bits match those within the ACK segment, then a TCB is initialized and the connection proceeds. The encoded MSS bound is used to set a reasonable-sized MSS that is no larger than what was originally advertised. This MSS is usually implemented as three bits whose code points correspond to eight "commonly advertised" MSS values based on typical link *Maximum Transmission Units* (MTUs) and header overheads.

Hybrid Approaches: A hybrid approach combines two or more of the single defense techniques described previously. For instance, some end-host operating systems implement both a large backlog and SYN cookies, but enable SYN cookies only when the amount of the backlog that is occupied exceeds some threshold, allowing them to normally operate without the disadvantages of SYN cookies, but also allowing them to fail over to the SYN-cookie behavior and be strongly protected when an attack occurs.

Network-Based Countermeasures

Filtering: The most basic network-level defense is application of the filtering techniques described in RFC 2827[7]. Using ingress filtering, an ISP refuses to further route packets coming from an end site with IP source addresses that do not belong to that end site. Ingress filtering would be highly effective at preventing SYN flooding attacks that rely on spoofed IP packets. However, it is not currently reliable because ingress filtering policies are not universally deployed. Ingress filtering is also wholly ineffective against SYN flooding attacks that use a distributed army of controlled hosts that each directly attack. Ingress filtering is also a mechanism that an end site wishing to defend itself most often has no control over, because it has no influence upon the policies employed by ISPs around the world.

Firewalls and Proxies: A firewall or proxy machine inside the network can buffer end hosts from SYN flooding attacks through two methods, by either spoofing SYN-ACKs to the initiators or spoofing ACKs to the listener[9].

Figure 6 shows the basic operation of a firewall/proxy that spoofs SYN-ACKs to the initiator. If the initiator is legitimate, the firewall/proxy sees an ACK and then sets up a connection between itself and the listener, spoofing the initiator's address. The firewall/proxy splits the end-to-end connection into two connections to and from itself. This splitting works as a defense against SYN flooding attacks, because the listener never sees SYNs from an attacker. As long as the firewall/proxy implements some TCP-based defense mechanism such as SYN cookies or a SYN cache, it can protect all the servers on the network behind it from SYN flooding attacks.

Figure 6: Packet Exchanges through a SYN-ACK spoofing Firewall/Proxy.

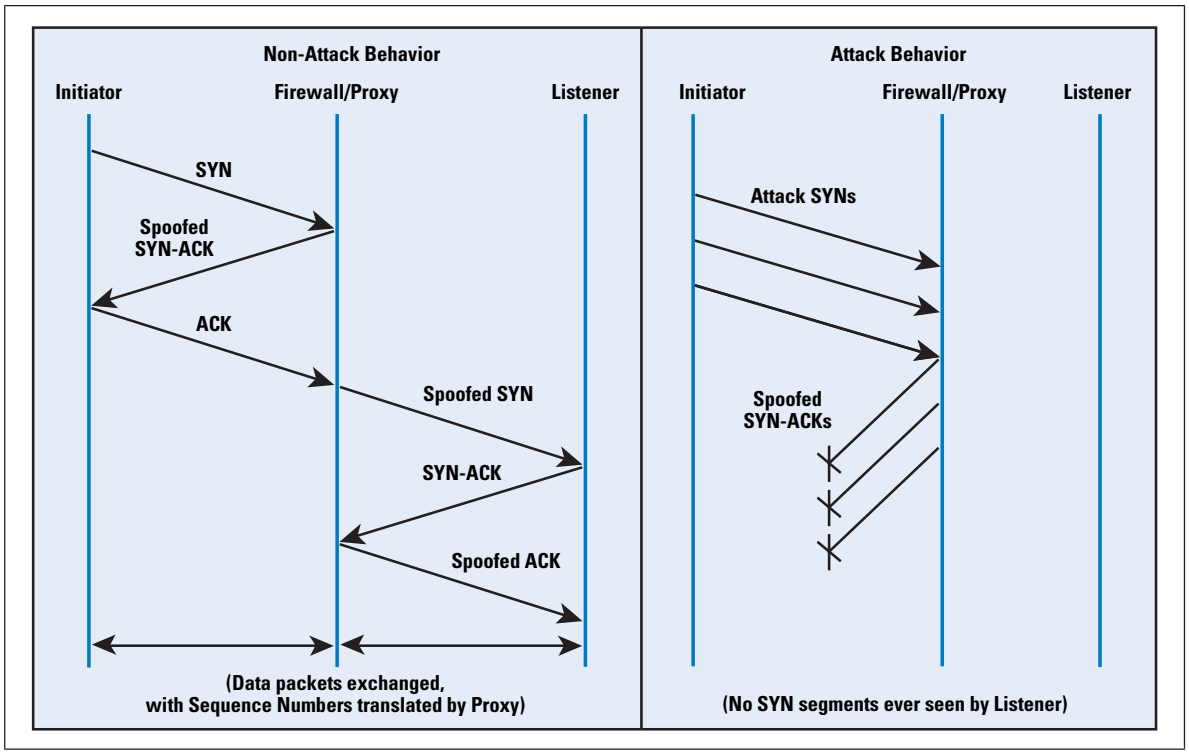
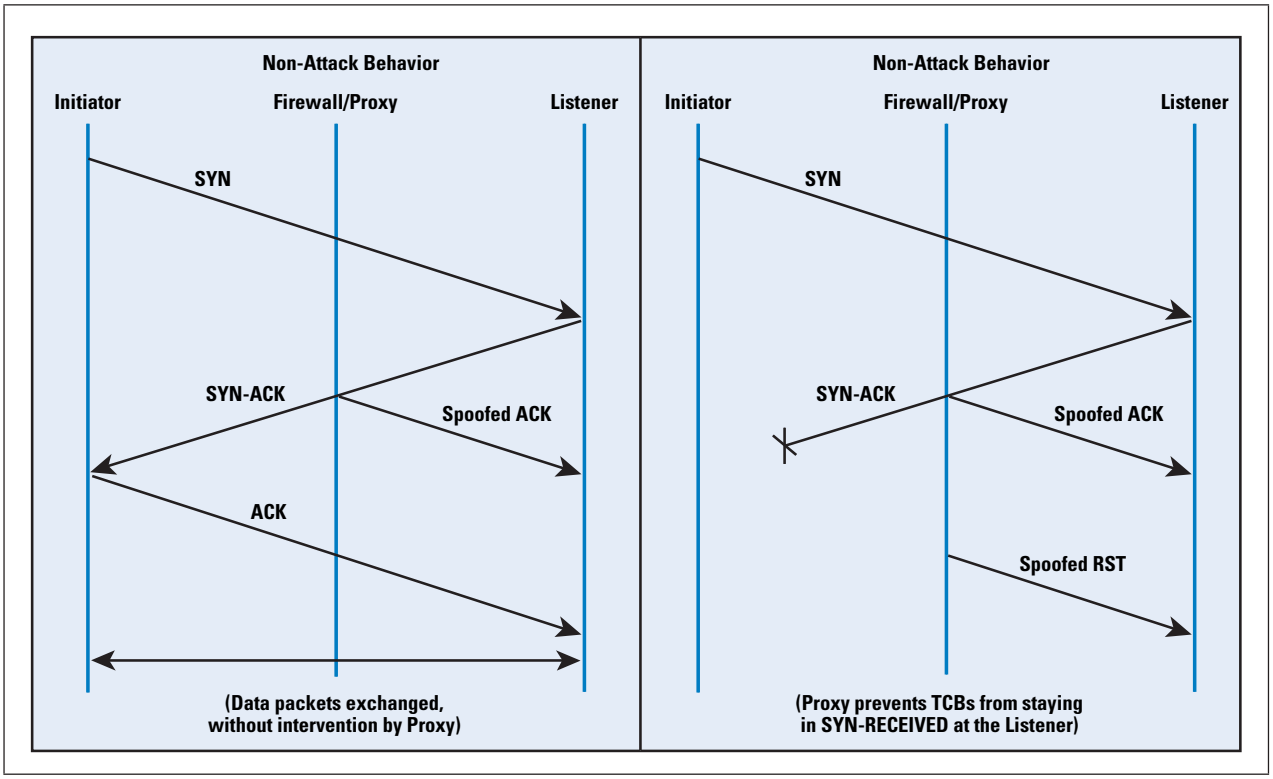


Figure 7 illustrates the packet exchanges through a firewall/proxy that spoofs ACKs to the listener in response to observed SYN-ACKs. This spoofing prevents the listeners TCBs from staying in the SYN-RECEIVED state, and thus maintains free space in the backlog. The firewall/proxy then waits for some time, and if a legitimate ACK from the initiator is not observed, then it can signal the listener to free the TCB using a spoofed TCP RST segment. For legitimate connections, packet flow can continue, with no interference from the firewall/proxy. This solution is more desirable than the mode of operation in Figure 5, where the firewall/proxy spoofs SYN-ACKs, because it does not require the firewall/proxy to actively participate in legitimate connections after they are established.

Figure 7: Packet Exchanges through an ACK-spoofing Firewall/Proxy.



Active Monitor: An active monitor is a device that can observe and inject traffic to the listener, but is not necessarily within the routing path itself, like a firewall is. One type of active monitor acts like the ACK-spoofing firewall/proxy of Figure 6, with the added capability of spoofing RSTs immediately if it sees SYNs from source addresses that it knows to be used by attackers[9]. Active monitors are useful because they may be cheaper or easier to deploy than firewall-based or filtering solutions, and can still protect entire networks of listeners without requiring every listener's operating system to implement an end-host solution.

Defenses in Practice

Both end-host and network-based solutions to the SYN flooding attack have merits. Both types of defense are frequently employed, and they generally do not interfere when used in combination. Because SYN flooding targets end hosts rather than attempting to exhaust the network capacity, it seems logical that all end hosts should implement defenses, and that network-based techniques are an optional second line of defense that a site can employ.

End-host mechanisms are present in current versions of most common operating systems. Some implement SYN caches, others use SYN cookies after a threshold of backlog usage is crossed, and still others adapt the SYN-RECEIVED timer and number of retransmission attempts for SYN-ACKs.

Because some techniques are known to be ineffective (increasing backlogs and reducing the SYN-RECEIVED timer), these techniques should definitely not be relied upon. Based on experimentation and analysis (and the author's opinion), SYN caches seem like the best end-host mechanism available.

This choice is motivated by the facts that they are capable of withstanding heavy attacks, they are free from the negative effects of SYN cookies, and they do not need any heuristics for threshold setting as in many hybrid approaches.

Among network-based solutions, there does not seem to be any strong argument for SYN-ACK spoofing firewall/proxies. Because these spoofing proxies split the TCP connection, they may disable some high-performance or other TCP options, and there seems to be little advantage to this approach over ACK-spoofing firewall/proxies. Active monitors should be used when a firewall/proxy solution is administratively impossible or too expensive to deploy. Ingress and egress filtering is frequently done today (but not ubiquitous), and is a commonly accepted practice as part of being a good neighbor on the Internet. Because filtering does not cope with distributed networks of drones that use direct attacks, it needs to be supplemented with other mechanisms, and must not be relied upon by an end host.

Related Attacks

In addition to SYN flooding, several other attacks on TCP connections are possible by spoofing the IP source address and connection parameters for in-progress TCP connections[10]. If an attacker can guess the two IP addresses, TCP port numbers, and a valid sequence number within the window, then a connection can be disrupted either through resetting it or injecting corrupt data. In addition to spoofed TCP segments, spoofed ICMP datagrams have the capability to terminate victim TCP connections.

Both these other attacks and SYN floods target a victim's TCP application and can potentially deny service to the victim using an attack rate less than that of brute-force packet flooding. However, SYN flooding and other TCP spoofing attacks have significant differences. SYN flooding denies service to new connections, without affecting in-progress connections, whereas other spoofing attacks disrupt in-progress connections, but do not prevent new connections from starting. SYN flooding attacks can be defended against by altering only the initial handshaking procedure, whereas other spoofing attacks require additional per-segment checks throughout the lifetime of a connection. The commonality between SYN flooding and other TCP spoofing attacks is that they are predicated on an attacker's ability to send IP packets with spoofed source addresses, and a similar defense against these attacks would be to remove this capability through more universal deployment of address filtering or IPsec.

Conclusion

At the time of this writing, the TCP SYN flooding vulnerability has been well-known for a decade. This article discussed several solutions aimed at making these attacks ineffective, some of which are readily available in commercial off-the-shelf products or free software, but no solution has been standardized as a part of TCP or middlebox function at the IETF level. The IETF's *TCP Maintenance and Minor Extensions* (TCPM) working group is in the process of producing an informational document that explains the positive and negative aspects of each of the common mitigation techniques[10], and readers are encouraged to consult this document for further information.

In this author's opinion, some variant of the SYN cache technique should be a mandatory feature to look for in a server operating system, and the variant can be deployed in combination with other network-based methods (address-based filtering, ACK-spoofing firewalls, IPsec, etc.) in appropriate situations. It is encouraging to see that protocol designers have learned a lesson from the SYN flooding vulnerability in TCP and have made more recent protocols inherently robust to such attacks.

Acknowledgements

Several individual participants in the IETF's TCPM working group have contributed bits of data found in the group's informational document on SYN flooding[11], some of which is replicated in spirit here.

References

[1] daemon9, route, and infinity, "Project Neptune," *Phrack Magazine,* Volume 7, Issue 48, File 13 of 18, July 1996.

[2] CERT, "CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks," September 1996.

[3] Aura, T. and P. Nikander, "Stateless Connections," Proceedings of the First International Conference on Information and Communication Security, 1997.

[4] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and V. Paxson, "Stream Control Transmission Protocol," RFC 2960, October 2000.

[5] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol," RFC 4306, December 2005.

[6] Aura, T., Nikander, P., and J. Leiwo, "DOS-resistant Authentication with Client Puzzles," *Lecture Notes in Computer Science,* Volume 2133, revised from the 8th International Workshop on Security Protocols, 2000.

[7] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," BCP 38, RFC 2827, May 2000.

[8] Lemon, J., "Resisting SYN Flood DoS Attacks with a SYN Cache," BSDCON 2002, February 2002.

[9] Schuba, C., Krsul, I., Kuhn, M., Spafford, E., Sundaram, A., and D. Zamboni, "Analysis of a Denial of Service Attack on TCP," Proceedings of the 1997 IEEE Symposium on Security and Privacy, 1997.

[10] Touch, J., "Defending TCP Against Spoofing Attacks," Internet-Draft (work in progress), **draft-ietf-tcpm-tcp-antispoof-05**, October 2006.

[11] Eddy, W., "TCP SYN Flooding Attacks and Common Mitigations," Internet-Draft (work in progress), **draft-ietf-tcpm-syn-flood-00**, July 2006.

WESLEY M. EDDY works for Verizon Federal Network Systems as an onsite contractor at NASA's Glenn Research Center, where he performs research, analysis, and development of network protocols and architectures for use in space exploration and aeronautical communications. E-mail: **weddy@grc.nasa.gov**

Boosting the SOA with XML Networking

by Silvano Da Ros

In the 1990s, the widespread adoption of *object-oriented programming* (OOP) and advancing network technologies fostered the development of distributed object technologies, including *Object Management Group's* (OMG's) *Common Object Request Broker Architecture* (CORBA) and Microsoft's *Distributed Common Object Model* (DCOM). Both CORBA and DCOM follow the OOP consumer-producer service model, where applications locally instantiate any number of objects and execute methods for the objects to obtain a service. However, with *distributed* object technologies, a local application can request a service from a remote application by instantiating a remote object and executing the methods of the object using *Remote Procedure Call* (RPC) over the network. The local application executes the methods of the remote object as if the object were an inherent part of the local application.

To push toward a simpler consumer-producer service model than distributed objects, the *Service-Oriented Architecture* (SOA) was created as a worldwide standards-based application interoperability initiative[1]. SOA differs from distributed object technologies, because you no longer deal with object instantiation and method invocation to provide services between your applications[2]. Instead, you can create *Extensible Markup Language* (XML)-based standard Web services to exchange XML documents between your applications using Internet-based application layer protocols, such as *Hyper Text Transfer Protocol* (HTTP) and the *Simple Mail Transfer Protocol* (SMTP).

This is where XML networking comes into the picture. This article shows how to use SOA at the network edge, in conjunction with XML within the network, to help with the work required for enabling interoperability between your applications. The problem with SOA on its own is that to scale applications, hardware and software upgrades are required on the servers where your business logic resides. Because application integration using XML is CPU-intensive, it benefits from XML hardware built specifically for XML computations. However, the applications servers that run your business logic are effectively independent of the underlying XML processing. Therefore, to accelerate the SOA at the network level transparently to the application, XML networking technologies can be used. XML networking can provide SOA acceleration using a special middleware-enabled network layer, which this article explains. This special network layer also provides additional benefits to your applications that SOA alone cannot provide at the edge, such as dynamic message routing and transformation.

To help in the understanding of SOA acceleration with XML networking, the following section discusses SOA and its constituent technologies. Further sections explore the specifics of XML and XML-based network processing.

A Brief History of SOA

Traditionally, hand-coding proprietary interfaces were required to interoperate between your applications, as Figure 1 illustrates. This task is a trivial one if you have only a few applications, but if you have numerous disparate applications, all requiring interfaces into one another, the result is a complex, many-to-many web of connections between them. In the 1980s, *Electronic Data Interchange* (EDI) was developed to standardize the message formats that trading partners use to exchange text-based transaction data residing on mainframes, making it an early predecessor to SOA.

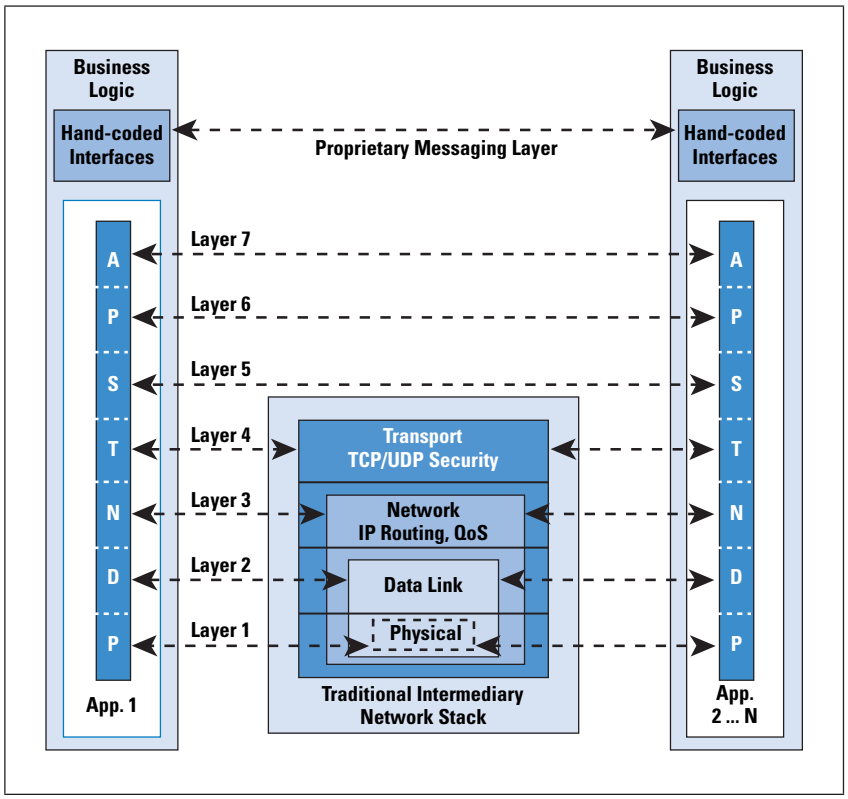
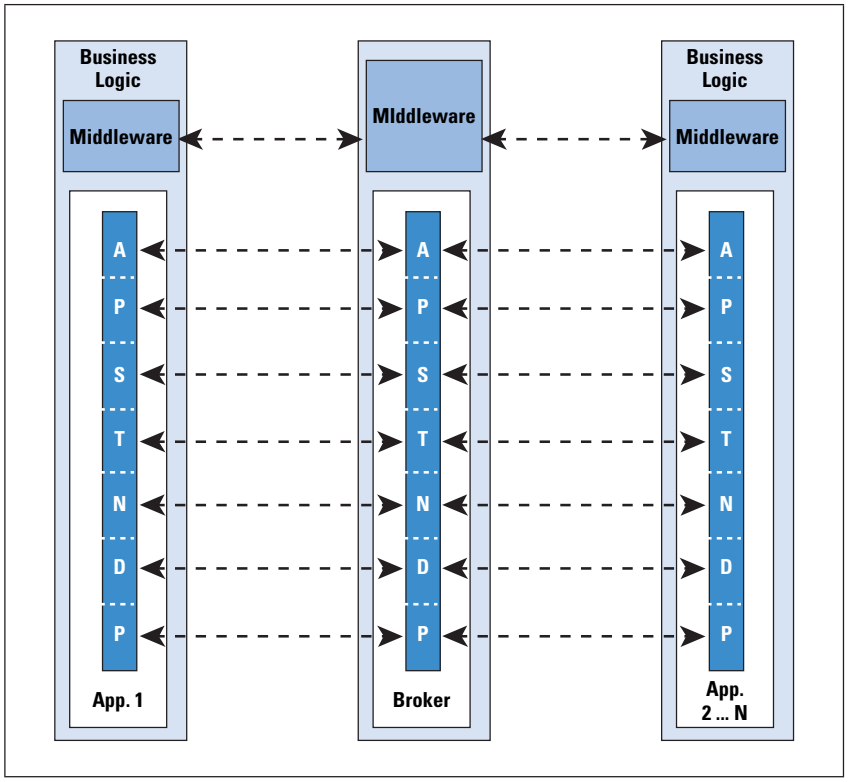


Figure 1: The Proprietary Messaging Layer

In the mid-1990s, standard middleware (or integration brokers) became available, such as CORBA and DCOM mentioned previously, to integrate advanced client-server applications. Figure 2 shows how integration brokers allow you to perform the translations between end systems over a standard messaging layer without creating application-specific interfaces between each system. During the same time, numerous software vendors, such as IBM WebSphere and TIBCO, also developed standard messaging layer protocols, which required adding vendor-specific adapters within the common integration brokers. Additionally, with newer application development environments being adopted, such as *Java 2 Sun Enterprise Edition* (J2EE) and Microsoft .NET, even more programming complexity is required when considering application interoperability without using the SOA. Fortunately, these new platforms currently support the SOA, allowing an application developed in one platform to tap into the data supplied by an application developed in the other.

Figure 2: The Standard Messaging Layer



Figures 1 and 2 illustrate how the proprietary and standard messaging layers sit above the network stack—at best, a traditional network device can operate only up to and including the transport layer. For example, by tracking TCP connection state information, a firewall device allows you to configure security services for your applications. Some firewalls can inspect the context of the application, but only to ensure the application behavior is RFC-compliant and not performing some sort of malicious activity. Additionally, at the next layer down the stack, you can configure Layer 3 *Quality of Service* (QoS) functions, such as *IP Precedence*, *Differentiated Services* (DiffServ), traffic shaping, and resource reservation, to ensure delivery of traffic to your critical applications. Although the network layers can provide these intelligent network services to your applications, they do not add any value toward accelerating your SOA.

Notice how the middleware portion in the proprietary messaging layer in Figure 1 takes up a larger portion of the application stack than the standard messaging layer from Figure 2. This situation occurs because the list of available messages that your standard messaging layer applications support is now much smaller—the broker takes care of the interfacing complexity on behalf of your applications. A reduced number of messages requires that you maintain much less middleware programming code on your applications than if every application in your network had to account for the messages of every other application.

Optimizing the SOA

Now that you understand SOA, you can better understand where XML networking fits into the scheme of things. Figure 3 illustrates how network equipment vendors can add specialized "application-aware" intelligence into Layers 5 through 7 of the OSI model.

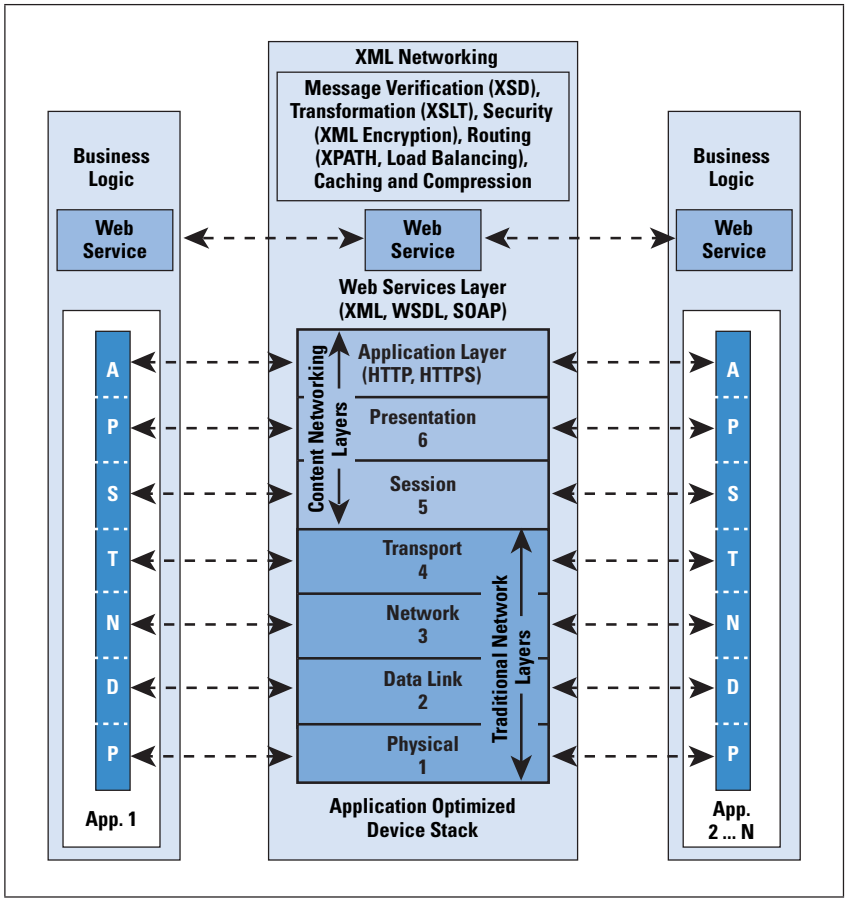
You can start by contrasting XML networking with traditional content networking technologies[3]. As you can see in Figure 3, by incorporating content networking services into the network, such as *Server Load Balancing* (SLB), caching, and *Secure Sockets Layer* (SSL) acceleration, network vendors give you the ability to transparently accelerate your applications without the need of application hardware upgrades. However, by residing only within the OSI model, content networking services and protocols provide a "network-oriented" way to accelerate your applications. In order to achieve full application awareness, you must look not only into the application headers, but also into the application payload. Although the content networking protocols can inspect into the packet payload, they are meant for providing network layer services but *not* application integration services. For example, *Network-Based Application Recognition* (NBAR) allows you to mark the IP DiffServ field in packets containing high-priority application traffic by first detecting the behavior of the application. However, like the network layers, the content networking layers cannot fulfill SOA acceleration requirements either.

In contrast, XML networking provides integration services by inspecting the full context of the application transaction and adding XML standards-based intelligence on top of the TCP/IP stack. An XML-enabled network provides you greater control, flexibility, and efficiency for integrating your applications than integration brokers. Figure 3 shows how you can inspect the XML-based "Web services" layer to accelerate your applications developed within an SOA model without the need of an integration broker.

The most popular Web services protocol is *Simple Object Access Protocol* (SOAP)[4]. With SOAP, your applications can request services from one another with XML-based requests and receive responses as data formatted with XML. Because SOAP uses XML, its Web services are self-descriptive and very simple to use.

You define your SOAP Web services with the XML-based *Web Services Description Language* (WSDL)[5]. The WSDL binds the SOAP messages to the Web services layer, as discussed later in this article. You can then transport your SOAP messages over standard application layer protocols, such as HTTP and HTTPS, between your client-server applications.

Figure 3: The Web Services Layer

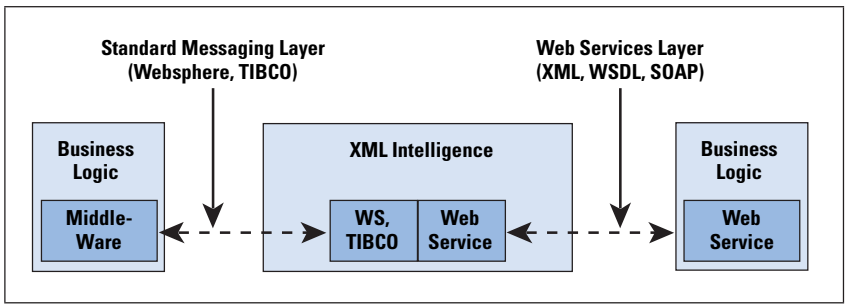


Similar to content networking technologies, XML networking can transparently add value to your applications within the network. When the XML network device receives the standard XML document from the Web services layer, you can configure the device to perform application-oriented services on the document. But because XML networking operates at the middleware layer and uses standard documents to integrate your applications, it provides you with fully standard functions using languages for:

- *Message Verification:* You can develop *World Wide Web Consortium* (W3C) *XML Schema Definitions* (XSDs) that your XML network can use to verify the syntax of your XML documents[6].
- *XML Translation:* Using *XML Stylesheet Language Transformations* (XSLT), you can translate XML documents to other non-XML formats, and conversely, directly within your network[7].
- *Context-Based Routing:* Use *XML Path* (XPATH) to route messages based on data stored within XML documents[8]. A popular example is to route stock exchange quotes to a desired location when the value of the stock drops below a certain threshold.
- *High Availability:* Messages containing specific content can be load balanced across numerous identical origin servers.

- *Data Security:* You can accelerate XML encryption computations using either hardware or software XML-accelerated devices.
- *Compression and Caching:* You can cache frequently requested XML documents and compress XML documents to reduce network bandwidth. Like XML encryption, XML caching and compression can be performed using either hardware or software XML-accelerated devices.
- *Application-Layer Request Translation:* You can use XML networking to convert non-Web service requests into standard Web service requests. As with integration brokers, vendor-specific adapters are required to translate between WebSphere MQ, TIBCO, and SOA Web services. For example, Figure 4 shows how you can use network-level XML intelligence to translate between Websphere or TIBCO messages and Web services layer XML-based messages.

Figure 4: Intelligent Protocol Switching



Introducing XML Service Languages

Now that you have a general understanding of both SOA and XML networking, you can examine the specific XML technologies used for application interoperability, including:

- *XML* is used to format application data for storage and transmission.
- *XSLT* is used to translate between one XML format to another.
- *XSD* is used to describe, control, and verify an XML document format.
- *XPath* is a way to address items in an XML document hierarchy.
- *SOAP* is a messaging protocol used to encode information in Web service request and response messages before sending them over a network.

XML has its roots in the late 1960s from the *Generalized Markup Language* (GML), which was used to organize IBM's mainframe-based legal documents into a searchable form. The *Standard Generalized Markup Language* (SGML) was officially standardized in 1986 as an ISO international norm (ISO 8879). Since then, XML has become the predominant markup language for describing content. XML differs from HTML because it is not concerned with presenting or formatting content; instead, XML is used for describing the data using tags and attributes that you define yourself. Figure 5 is a sample XML document that organizes a police department's traffic ticket information.

Figure 5: A Traffic Ticket XML Example

```
<?xml version="1.0"?>
<dept-tickets>
 <dept-chief>Greg Sanguinetti"/>
 <dept-id>12389289/>
 <ticket id="034567910" code="301">
  <offender>
   <name>John Smith</name>
   <license-number>10003887</license-number>
   <plate-number>9AER9876</plate-number>
  </offender>
  <offence-date>09/30/2005</offence-date>
  <location>
   <state>CA</state>
   <city>SJ</city>
   <intersection>West Tasman Dr.-Great America Pkwy.</intersection>
  </location>
  <officer>
   <officer-name>Paul Greene</officer-name>
   <officer-badge>7652323</officer-badge>
   <cruiser-plate-number>6TYX0923</cruiser-plate-number>
  </officer>
  <description>Failure to stop at red light</description>
  <fine>100</fine>
 </ticket>
 <ticket id="..." code="...">
 ...
 </ticket>
 <ticket id="..." code="...">
 ...
 </ticket>
</dept-tickets>
```

The XML in Figure 5 identifies the group of tickets for a police department by the department ID and the department chief's name. This example gives the data for a single traffic ticket as defined by the "ticket" element (or tag); however, you could include as many tickets as you want within the element "dept-tickets." The "ticket" element has two self-explanatory attributes (in dark blue), called "id" and "code," referring to the identification number for the individual ticket and the offense code, respectively. The sub-elements of the "ticket" element are also self-explanatory: "offender," "offence-date," "location," "officer," "description" and "fine."

In order to build a well-formed XML document, you must embrace the data for each element within its respective open and close tags (for example, <ticket>...</ticket>, or <ticket>.../>), properly nest all elements, and make sure all element names are in the proper case. You must also specify the XML version with the "<?xml version="1.0"?>" tag at the beginning of the XML document. HTML is less rigid than XML because it is case-insensitive and most Web browsers will allow you to leave out the close tags of an element. Because XML is very strict about following the correct syntax (that is, by making sure the XML is well-formed), XML processors are much easier to develop than traditional HTML Web browsers.

To verify that your documents are valid XML, you can check them against XSD files, which define the XML elements and their sequence, as discussed later in this article.

Transforming XML Using XSLT

You can use XSLT to translate one XML-based language into another. For example, you can translate standard XML into HTML. To translate the XML from Figure 5 into HTML for online viewing, you can use the XSLT file in Figure 6.

Figure 6: XSLT Translation – From XML to HTML

```
<?xml version="1.0"?>
<xslt:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xslt:template match="/">
  <html>
   <body>
    <br><b>Chief: </b><xslt:value-of select="dept-tickets/dept-chief"/></br>
    <br><b>Department No: </b><xslt:value-of select="dept-tickets/dept-id"/>
    </br>
    <table border="5">
     <!-- Output the HTML table headings -->
     <th>Ticket Number</th>
     <th>Offender's Name</th>
     <th>License Number</th>
     <th>State of Offense</th>
     <th>Officer's Name</th>
     <!-- Output the HTML table data -->
     <xslt:for-each select="dept-tickets/ticket">
      <tr>
       <td align="center"><xslt:value-of select="@id"/></td>
       <td align="left"><xslt:value-of select="offender/name"/></td>
       <td align="center"><xslt:value-of select="offender/license-number"/></td>
       <td align="center"><xslt:value-of select="location/state"/> </td>
       <td align="left"><xslt:value-of select="officer/officer-name"/></td>
       <td align="right">$<xslt:value-of select="fine"/></td>
      </tr>
     </xslt:for-each>
    </table>
   </body>
  </html>
 </xslt:template>
</xslt:stylesheet>
```

You must use a namespace to differentiate elements among the XML-based languages that you use in your XML document. As Figure 6 illustrates, the namespace is the string "xslt:", which prefixes all of the XSLT elements. The particular application that parses the document (whether it is your XML device or a standalone XSLT parser[9]) will know what to do with the specific elements based on the prefix. For example, an XSLT parser will look for the specific *Universal Resource Indicator* (URI) string constant that the W3C assigned to XSLT (that is, **`http://www.w3.org/1999/XSL/Transform`**) and perform the intended actions based on the elements in the document.

XML parsers do not use the URI of the namespace to retrieve a schema for the namespace—it is simply a unique identifier within the document. According to W3C, the definition of a namespace simply defines a two-part naming system (for example, "xslt:for-each") and nothing else. After you define the namespace, the XML parser will understand the elements used within the document, such as "for-each" and "value-of" specified in Figure 6. For XSD documents, you must use a different namespace URI (that is, **`http://www.w3.org/2001/XMLSchema`**), as the next section discusses.

When you configure an XSLT parser or XML networking device to apply XSLT to an XML document, the parser starts at the top of the XSLT document by matching the root XML element within the source XML file. For example, the <xslt:template match="/"> element in Figure 6 matches the "dept-tickets" root element from the XML file in Figure 5. The XSLT parser then creates the destination XML document (that is, a well-formed HTML file, in this example) and outputs the <html> and <body> tags to the new document. The XSLT parser then outputs the HTML table headers and loops through the XML document "ticket" elements, outputting selected items within the columns of the HTML table. The resulting HTML is given in Figure 7 for three sample tickets.

Figure 7: Resulting HTML Table – Source View

```
<html>
 <body>
  <br><b>Chief: </b>Greg Sanguinetti<br><b>Department No: </b>12389289
  <table border="5">
   <th>Ticket Number</th><th>Offender's Name</th>
   <th>License Plate</th><th>State of Offense</th>
   <th>Officer's Name</th><th>Fine Amount</th>
   <tr>
    <td>034567910</td><td>John Smith</td><td>10003887</td>
    <td>CA</td><td>Paul Greene</td><td>100</td>
   </tr>
   <tr>
    <td>042562930</td><td>Gerald Rehnquist</td><td>11023342</td>
    <td>CA</td><td>Joel Patterson</td><td>200</td>
   </tr>
   <tr>
    <td>182736493</td><td>Jenny Barker</td><td>47281938</td>
    <td>CA</td><td>Emily Jones</td><td>120</td>
   </tr>
  </table>
 </body>
</html>
```

Figure 8 illustrates the resultant HTML table that clients would see within a Web browser after the XSLT translation takes place.

Figure 8: Resulting HTML Table – Browser View

Ticket Number	Offender's Name	License Plate	State of Offense	Officer's Name	Fine Amount
034567910	John Smith	10003887	CA	Paul Greene	100
042562930	Gerald Rehnquist	11023342	CA	Joel Patterson	200
182736493	Jenny Barker	47281938	CA	Emily Jones	120

Verifying XML Using XSD

Because you can customize the structure and tags within an XML document, you should verify its syntax using XSDs. The XSD file in Figure 9 verifies the XML document given previously in Figure 5.

Figure 9: XSD File for Validating Traffic Ticket XML

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="dept-tickets">
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="dept-chief"/>
    <xsd:element name="dept-id"/>
    <xsd:element name="ticket" maxOccurs="unbounded">
     <xsd:complexType>
      <xsd:sequence>
       <xsd:element name="offender">
        <xsd:complexType>
         <xsd:sequence>
          <xsd:element name="name"/>
          <xsd:element name="license-number"/>
          <xsd:element name="plate-number"/>
         </xsd:sequence>
        </xsd:complexType>
       </xsd:element>
       <xsd:element name="offence-date"/>
       <xsd:element name="location">
        ...
       </xsd:element>
       <xsd:element name="officer">
        ...
       </xsd:element>
       <xsd:element name="description"/>
       <xsd:element name="fine"/>
      </xsd:sequence>
      <xsd:attribute name="id"/>
      <xsd:attribute name="code"/>
     </xsd:complexType>
    </xsd:element>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

You must define an XSD namespace with the URI "**http://www.w3.org/2001/XMLSchema**" and prefix all the XSD elements that you use in the XSD file, such as "element," "complex-type," and "attribute," with this namespace. At the top of your XSD file, you must specify the root XML element; the remaining elements within your XML document can be defined within the root element. Using the "complex-type" XSD element, you can specify elements that contain child elements (in contrast, "simple-type" indicates that the element does not contain any child elements). In this example, the "dept-tickets" element may contain a sequence of one or more child elements (as represented by the <xsd:sequence> element), including "dept-chief," "dept-id," and any number of element "ticket."

Routing Messages Using XPATH

XPATH was developed primarily to be used with XSLT to transform the XML tags within an XML document based on the path of the data. Previously, in Figure 6, you saw how to select the entire list of tickets using the XSLT "select" attribute:

```
xsl:value-of select="dept-tickets/ticket"
```

However, within an XML network, you can also use XPATH to search within an XML document to route XML messages based on the values of the document data. For example, a state government may need the headquarters police department to route unpaid tickets that are within a tolerable threshold amount to the motor vehicle department for processing—there, the driver's license can be suspended until the ticket is paid. However, those unpaid tickets that exceed a maximum threshold amount must be routed to the court service government department for processing. The court may decide to press further charges, depending on the driver's previous driving record. Additionally, severe infractions, such as drunken or reckless driving, must be routed automatically to the court, regardless of whether the ticket is paid or not. The XPATH expression "dept-tickets/ticket" given previously returns the entire list of traffic tickets. Alternatively, if you want only the unpaid tickets with a fine value of greater than $100, you could use the XPATH expression:

```
dept-tickets/ticket[@paid='no' and fine>100]
```

The XPATH symbol "@" here indicates that an attribute is being selected, and not an element. To select tickets with codes 309 and 310 (that is, fictitious codes for severe infractions), you can use the following XPATH expression:

```
dept-tickets/ticket[@code=309 or @code=310]
```

Using SOAP Web Services

SOAP provides a standard way to send transaction information over TCP/IP application protocols, such as HTTP. For example, you could create a SOAP request-response operation over HTTP for exchanging traffic ticket information between two applications. As Figure 10 illustrates, the requesting client application sends a "getFineRequest" message to the server, which in turn responds with the appropriate fine amount within a "getFineResponse" message.

Figure 10: A Sample SOAP Request-Response Operation

```
Client Request :
POST /getticketfine HTTP/1.1
Host: www.example.com
Content-Type: application/soap+xml;

<?xml version="1.0"?>
<soap:envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:body>
  <tn:getFineRequest xmlns:tn="http://example.com/getticketfine">
    <tn:ticket-id>034567910</tn:ticket-id>
  </tn:getFineRequest >
 </soap:body>
</soap:envelope>

Server Response:
HTTP/1.1 200 OK
Content-Type: application/soap+xml;

<?xml version="1.0"?>
<soap:envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:body>
  <tf:getFineResponse xmlns:tf="http://example.com/getticketfine">
    <tf:fine>100</tf:fine>
  </tf:getFineResponse>
 </soap:body>
</soap:envelope>
```

You encapsulate each SOAP message within the "Envelope" SOAP element. Within Envelope, you need to prefix the SOAP elements with the SOAP namespace, called "soap:" in this example, which you define as an attribute within Envelope. The "encodingStyle" attribute of the Envelope element defines the data types in the SOAP document. You must also define a custom namespace (that is, "tf," which stands for "ticket-fine"), with which you prefix all the application-specific elements.

To define the structure of the SOAP Web service running within your applications, you can use WSDL, which you develop so that your clients know the exact specification of the services that they can request, the types of responses they should expect to receive, and the protocols (for example, SOAP or HTTP) with which they should send messages.

For example, you can publish the WSDL to your clients, who may not be aware of the messages available within your Web services layer. The clients can retrieve the WSDL file and send the appropriate SOAP messages to the SOAP Web service running on your application. To publish the WSDL file to your clients, you can use a publicly available *Universal Description, Discovery and Integration* (UDDI) registry, such as XMethods[10], or you could create your own UDDI registry[11].

WSDL uses XSD to define your SOAP application data types. For example, for one application to request a fine amount (of XSD type xs:integer) for a given ticket ID (of XSD type xs:string) from your SOAP Web service called "ticketFineService," you could use the WSDL in Figure 11.

Figure 11: WSDL for SOAP Request-Response Operation

```
<?xml version="1.0"?>
<definitions name="TicketInfo"
 targetNamespace="http://example.com/ticketinfo.wsdl"
 xmlns:tns="http://example.com/ticketinfo.wsdl"
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns="http://schemas.xmlsoap.org/wsdl/">

 <message name="getFineRequest">
  <part name="ticket-id" type="xs:string"/>
 </message>

 <message name="getFineResponse">
  <part name="value"  type="xs:integer"/>
 </message>

 <porttype name="ticketFine">
  <operation name="getTicketFine">
   <input message="tns:getFineRequest"/>
   <output message="tns:getFineResponse"/>
  </operation>
 </porttype>

 <binding name="ticketBinding" type="ticketFine">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getTicketFine">
   <soap:operation soapAction="getTicketFine"/>
   <input>
    <soap:body use="encoded"/>
   </input>
   <output>
    <soap:body use="encoded"/>
   </output>
  </operation>
 </binding>

 <service name="ticketFineService">
  <documentation>WSDL File for ticketFineService</documentation>
  <port name="ticketFine" binding="ticketBinding">
   <soap:address location="http://example.com/getticketfine"/>
  </port>
 </service>
</definitions>
```

You start your WSDL file by declaring all the required namespaces. In order for the WSDL file to refer to element names that are defined within the same file (for example, "tns:getFineRequest" within the "porttype" element), you must use the "targetNamespace" element to define a custom URI that your custom namespace uses (that is, "tns," meaning "this name space").

You define the WSDL namespace for SOAP-specific elements with *xmlns: soap=http://schemas.xmlsoap.org/wsdl/soap*. For WSDL-only elements, you can use the default namespace *xmlns= http://schemas.xmlsoap.org/wsdl/*. Note that elements within the file that do not have a prefix use the default namespace.

After you create the namespaces for the WSDL file, you can then create the two messages for the transaction, "getFineRequest" and "getFineResponse," using WSDL "message" elements. WSDL ports create the request-response transaction flow using the "operation" element, by specifying which message is the request (input) and which is the response (output). After you define the transaction, you must bind it to SOAP with WSDL using the WSDL "binding" element. Additionally, to set the transport to HTTP, you must use the "binding" SOAP-specific element. You then link the operation you created previously within the WSDL "port-type" element to SOAP using the "operation" subelement within the parent "binding" element.

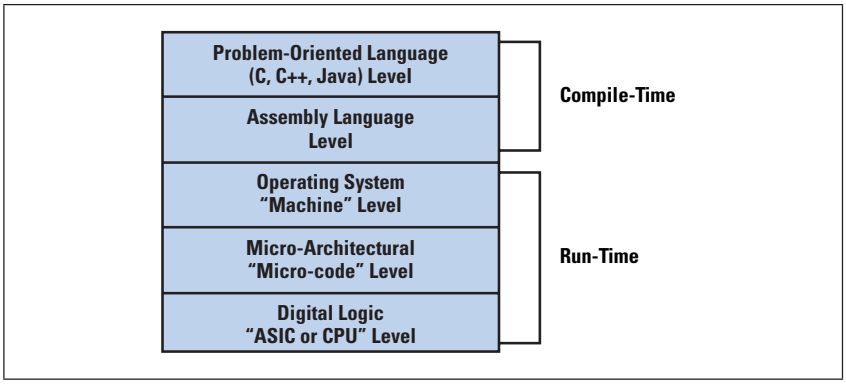
If you set the "use" element to "encoding," you do not need to use an XSD "type" attribute for defining SOAP data types in your SOAP messages. However, you must specify the "encodingStyle" URI to **`http://www.w3.org/2001/12/soap-encoding`**, as you learned previously in Figure 10. Otherwise, if you set "use" to "literal," then you would need to use the *type="xsd:string"* attribute in Figure 10 within the "tn:ticket-id" element when sending a request.

To define the SOAP Web service, you must use the WSDL "service" element. The SOAP element "address" within this element is the location where SOAP clients can send the "getTicketFine" requests, as Figure 10 illustrates.

Hardware vs. Software XML Acceleration

To help you understand the difference between hardware and software XML acceleration, Figure 12 illustrates a typical multilevel computer architecture[12]. The highest level is where you would typically program your applications. When you compile your application, the compiler would typically "assemble" your various objects into assembly language prior to generating the machine-level code. This machine code is what is normally stored in an ".exe" file, which only your operating system can understand. When you execute the ".exe" at run time, the operating system converts the machine-level code into microcode, which the digital logic level within the CPU hardware can execute directly.

Figure 12: A Multilevel Computer Architecture



Examples of software-based network applications in the past that have transitioned to hardware acceleration include IP routing, encryption, firewalling, caching, and load balancing. XML networking is also a recent candidate for hardware acceleration; it is available by XML vendors that use XML *Application-Specific Integrated Circuits* (ASICs) or *Field Programmable Gate Arrays* (FPGAs) in their products[13]. By programming the digital logic layer with the necessary circuits to perform intensive XML computations such as XSLT transformation, XML encryption, and XML schema validation, you can drastically increase the performance of the hardware platform.

However, some vendors have also found clever ways of accelerating XML computations on general-purpose hardware. Accelerating XML in software requires bypassing the additional machine-level step at run time. By "compiling" XML-based language instructions directly into microcode at the micro-architectural level, you can introduce XML computations to the underlying hardware directly at run time. That is, executing XML microcode at the digital logic level bypasses additional processing at the operating system "machine" level.

Summary

When a technology matures as a software agent running within an application, the need often arises to move the agent's functions to the network. Indeed, this was the case with numerous software-based technologies of the past, such as IP routing, encryption, stateful firewall filtration, and server load balancing.

To facilitate the interoperability of diverse applications, SOA was developed as a prescription to complexity problems faced by commonly used distributed-object technologies. As SOA matures, the need to introduce XML-based functions to the network will grow. In order to streamline the responsibilities of an SOA-based application, you can transition your XML technologies, such as XML translation, validation, and security, from within the application to an XML-enabled network.

For Further Reading

[1] Hao He, "What Is Service-Oriented Architecture?" O'Reilly, September 30, 2003, **http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html**

[2] Werner Vogels, "Web Services Are Not Distributed Objects," Computing in Science and Engineering, November-December 2003, **http://computer.org/internet/**

[3] Christophe Deleuze, "Content Networks," *The Internet Protocol Journal*, Volume 7, Number 2, June 2004.

[4] "SOAP Version 1.2 Part 0: Primer," W3C Recommendation, 24 June 2003, **http://www.w3.org/TR/2003/REC-soap12-part0-20030624/**

[5] "Web Services Description Language (WSDL) Version 2.0 Part 0: Primer," W3C Working Draft, 3 August 2005, **http://www.w3.org/TR/2005/WD-wsdl20-primer-20050803/**

[6] "XML Schema Part 0: Primer Second Edition," W3C Recommendation, 28 October 2004, **http://www.w3.org/TR/xmlschema-0/**

[7] "XSL Transformations (XSLT), Version 2.0," W3C Recommendation, 16 November 1999, **http://www.w3.org/TR/xslt**

[8] "XML Path Language (XPath) Version 1," W3C Recommendation, 16 November 1999, **http://www.w3.org/TR/xpath**

[9] **www.xmlspy.com**

[10] **www.xmethods.net**

[11] **www.uddi.org**

[12] Andrew S. Tanenbaum, *Structured Computer Organization,* (5th edition), ISBN 978-0131485211, Prentice Hall, 2005.

[13] Michael John Sebastian Smith, *Application-Specific Integrated Circuits,* ISBN 978-0201500226, Addison-Wesley, June 1997.

SILVANO DA ROS currently works as a networking consultant in Toronto and has worked previously as a Systems Engineer for Cisco Systems. He is the author of *Content Networking Fundamentals,* published by Cisco Press. He holds a Bachelor of Computer Science and a Masters of Engineering (in Internetworking) from Dalhousie University. E-mail: **sdaros@sympatico.ca**

Letters to the Editor

Time to Live

As I read the very fine article entitled "IPv6 Internals" (IPJ Volume 9, No. 3, September 2006), I was prompted to review the history of the *Time to Live* (TTL) as discussed in section 5.3.1 of RFC 1812. Being gray of head, little facts from other eras come quickly to mind. The *Xerox Network Systems* (XNS) Internet Transport on which Novell Netware was based required that no router ever store a packet in queue longer than 6 seconds. Requirements of RFC 791 were also softened in RFC 1812; rather than *requiring* the TTL to be decremented at least once and additionally once per second in queue, that document requires that the TTL be treated as a hop count and—reluctantly—reduces the treatment of TTL as a measure of time to a suggestion.

The reason for the change is the increasing implementation of higher-speed lines. A 1,500-byte datagram occupies 12,000 bits (and an asynchronous line sends those as 15,000 bits), which at any line speed below 19.2 kbps approximates or exceeds 1 second per datagram. Any time there are several datagrams in queue, the last message in the queue is likely to sit for many seconds, a situation that in turn can affect the behavior of TCP and other transports. However, 56-kbps lines became common in the 1980s, and T1 and T3 lines became common in the 1990s. Today, hotels generally offer Ethernet to the room; we have reports of edge networks connected to the Internet at 2.5 Gbps, and residential broadband in Japan and Europe at 26 Mbps per household. At 56 kbps, a standing queue of five messages is required to insert a 1-second delay, and at T1 it requires a queue depth of more than 100 messages. At higher speeds, the issue becomes less important.

That is not to say that multisecond queues are now irrelevant. Although few networks are being built today by concatenating asynchronous links, in developing countries—and on occasion even in hotels here in Santa Barbara, California—people still use dialup lines. In Uganda, some networks that run over the instant messaging capacity of GSM [*Global System for Mobile Communications*], which is to say using 9,600-bps datagrams, have been installed under the supervision of Daniel Stern and UConnect.org (**www.uconnect.org**). Much of the world still measures *round-trip times* (RTTs) in seconds, and bit rates in tens of kbps.

The TCP research community, one member of which recently asked me whether it was necessary to test TCP capabilities below 2 Mbps, and the IETF community in general would do well to remember that the ubiquity of high bandwidth in Europe, North America, Australia, and Eastern Asia in no sense implies that it is available throughout the world, or that satellite communications and other long-delay pipelines can now be ignored.

—Fred Baker, Cisco Systems
fred@cisco.com

The author responds:

Although to the casual observer the evolution of the Internet seems one of continuously increasing speed and capacity, reality is slightly different. The original ARPANET used 50-kbps modems in the late 1960s. In the next three decennia or so, the maximum bandwidth of a single link increased by a factor 200,000 to 10 Gbps. Interestingly enough, the minimum speed used for Internet connections went down to a little under 10 kbps, so where once the ARPANET had a uniform link speed throughout the network, the difference between the slowest and the fastest links is now six orders of magnitude. The speed difference between a snail and a supersonic fighter jet is only five orders of magnitude. Amazingly, the core protocols of the Internet—IP and TCP—can work across this full speed or bandwidth gamut, although changes were made to TCP to handle both extremes better, most notably in RFCs 1144 and 1323.

Even though I don't think keeping track of the time that packets are stored in buffers, as suggested in the original IPv4 specification, makes much sense even in slow parts of the network, Fred makes a good point: many Internet users still have to deal with speeds at the low end of the range; some of us only occasionally when connecting through a cellular network, others on a more regular basis. Even in Europe and the United States many millions of Internet users connect through dialup. For someone who is used to having always-on multimegabit connectivity, going back to 56 kbps or worse, 9,600 bps can be a bizarre experience. Many of today's Websites are so large that they take minutes to load at this speed. Connecting to my mail server using the *Internet Mail Access Protocol* (IMAP) takes 15 minutes. And one of my favorite relatively new applications, podcasting, becomes completely unusable: downloading a 50-minute audio program takes hours at modem speeds.

And that's all IPv4. It is possible to transport IPv6 packets over the *Point-to-Point Protocol* (PPP) that is used for almost all low-speed connections, but in practice this isn't workable because there are no provisions for receiving a dynamic address from an ISP [*Internet Service Provider*]. With IPv4, Van Jacobson did important work to optimize TCP/IP for low-speed links (RFC 1144). By reducing the *Maximum Transmission Unit* (MTU) of the slow link and compressing the IP and TCP headers, it was possible to achieve good interactive response times by avoiding the situation where a small packet gets stuck between a large packet that may take a second or more to transmit over a slow link while at the same time reducing the header overhead. Although the IETF has later done work on IPv6 header compression, it doesn't look like anyone has bothered to implement these techniques, and the minimum MTU of 1,280 bytes creates significant head-of-line blocking when IPv6 is used over slow links.

Another example where low bandwidth considerations are ignored is the widespread practice of enabling RFC 1323 TCP high-performance extensions for all TCP sessions. RFC 1323 includes two mechanisms: a window scale factor that allows much larger windows in order to attain maximum performance over high-bandwidth links with a long delay, and a timestamp option in the TCP header that allows for much more precise round-trip time estimations. With these options enabled, every TCP segment includes 8 extra bytes with timestamp information. In addition to increasing overhead, the timestamp option introduces an unpredictable value into the TCP header that makes it impossible to use header compression, thereby negating the usefulness of RFC 1144. To add insult to injury, almost no applications allocate enough buffer space to actually use the RFC 1323 mechanisms.

Moral of the story for protocol designers and implementers: spend some time thinking about how your protocol works over slow links. You never know when you'll find yourself behind just such a link.

—Iljitsch van Beijnum
iljitsch@muada.com

Gigabit TCP and MTU Size

I appreciated Geoff Huston's thorough description about the current obstacles and research involving Gigabit TCP (IPJ, Volume 9, No. 3, June 2006). I have already shown the article to many of my colleagues. It appears that Geoff did not address one of the solutions, which is to increase the networkwide *Maximum Transmission Unit* (MTU). In theory that would allow the existing TCP congestion control to handle higher-speed connectivity. Perhaps he did not address the issue because it is infeasible to increase the MTU setting Internetwide, especially with 10-Gigabit Ethernet interfaces sporting a default MTU setting of 1,500 bytes. On the other hand, projects that own their own backbone infrastructure may find increasing the default MTU a feasible approach.

For more information about raising the MTU, please see: **http://www.psc.edu/~mathis/MTU/**

—Todd Hansen, UCSD/SDSC
tshansen@hpwren.ucsd.edu

The author responds:

Yes, it's true that increasing the size of the packet makes sound sense when the available bandwidth has increased. If the bandwidth increases by one order of magnitude and the packet size is increased by the same amount, then it is theoretically possible to effectively increase the throughput of the system without changing the packet processing load.

Effectively, if you regard the protocol interaction as a time sequence, then a coupling of increased bandwidth and comparably increased packet size preserves the time sequence interaction. Of course, as bandwidth on the network has increased we have not seen a comparable increase in MTU sizes, and today's networks exhibit a wide variety of MTUs and the importance of *Path MTU Discovery*, and coherent transmission of related MTU ICMP [*Internet Control Message Protocol*] messages becomes more critical as a consequence. Although the article concentrated on modifications to the TCP control algorithm, there is no doubting the importance of high-speed TCP senders and receivers using large TCP buffers to maximize the payload throughput potential.

—Geoff Huston, APNIC
gih@apnic.net

Drop us a Line!

We welcome any suggestions, comments or questions youmay have regarding anything you read in this journal. Send us an e-mail to **ipj@cisco.com**. Also, don't forget to let us know if your delivery address changes. You can use the online subscription system to change your own information by supplying your Subscription ID and e-mail address. The system will then send you an e-mail with a "magic" URL which will allow you to update your database record. If you don't have your Subscription ID or encounter any difficulties, just send us the updated information via e-mail.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Book Review

Internet Measurement

Internet Measurement: Infrastructure, Traffic & Applications, by Mark Crovella, Balachander Krishnamurthy, ISBN 0-470-01461-X, Wiley, 2006.

This book is a comprehensive reference guide to about 900 journal, conference, and workshop papers, and RFCs on the important and rapidly advancing field of Internet measurement. Interest in this growing field arises for three major reasons: commercial, social, and technical. Readers need nothing more than a keen interest in a methodical study of the subject matter from either a practical or research perspective to glean something from this book.

Organization

The book is centered on three architectural pillars relevant to measurement: *infrastructure, traffic,* and *applications*. Within each of these pillars, the topics are organized into four sections: *properties, challenges, tools,* and *state-of-the-art*. In the properties section, the authors review metrics that are important to measure in each area. In the challenges section, they discuss various difficulties and limitations that arise when trying to measure the metrics. The tools section covers some of the popular methods and products used to measure these metrics and work around the challenges mentioned previously. The intent is not to provide "user guides" for these tools. The state-of-the-art section presents the latest measurement results about covered properties and metrics, noting that they are subject to relatively fast obsolescence because of the rapidly evolving Internet.

The first three chapters provide background material. The first chapter provides an obligatory introduction to the Internet architecture, including how the "end-to-end" principle has been used for nearly 20 years to guide many design decisions in the Internet. The second chapter provides the analytic background necessary to study the Internet and cast its measurements in quantitative terms. The third chapter examines the nuts and bolts of Internet measurement, addressing the practical topics to consider in designing and implementing them, including the role of time and its sources.

The second part of this book also consists of three chapters, which cover the three pillars in depth. The first chapter defines metrics of interest for measuring the Internet and describes some of the barriers to their measurement, in particular "middleboxes," *Network Address Translators* (NATs), firewalls, and proxies that deviate from the end-to-end architecture principle, may block *User Datagram Protocol* (UDP) or *Internet Control Message Protocol* (ICMP) packets, or hinder visibility to endpoint IP addresses. The authors next explore various tools and methods for active and passive measurement, estimation, and inference of these metrics.

Readers may wonder why two important metrics are left out—router reliability and high availability—where *Open Shortest Path First* (OSPF) and the *Border Gateway Protocol* (BGP) "Graceful Restart" would be of interest.

The next chapter focuses on traffic properties that are important to understand, measure, and model. The authors examine the challenges in capturing, processing, storing, and managing large volumes of packets and flows, as well as those related to their statistical characterization. Readers engaged in data modeling and performance analysis will benefit from this chapter. The last chapter in this part of the book examines some popular applications: The *Domain Name System* (DNS), Web, and *Peer-to-Peer* (P2P). The authors discuss the shifts in application mix from the 1980s, when FTP was dominant, to the 1990s, when the *Hypertext Transfer Protocol* (HTTP) became dominant, to today, when by most accounts P2P is the dominant Internet protocol. Next, there is a thorough coverage of the what (properties), why (justification), and how to (tools) facets of measurement of the three popular applications, as well as some coverage of online games and streaming media.

The third part of the book covers material that spans multiple areas. Its first chapter deals with anonymization of collected measurement data, which arises because of the need for data sharing, while preserving identity-related, personal-sensitive, or business-sensitive information for applications previously examined. The second chapter provides a short—but important—coverage of the key areas where Internet measurement has played a role in security enforcement. Various attack types and tools to combat them are discussed. The third chapter examines numerous low-level monitoring tools for high-speed traffic capture, as well as an insightful look at the software architecture of two toolsets, *dss* and *Gigascope,* reflecting the experience of one of the authors at AT&T Labs with them. It also reviews some large-scale measurement platforms at the *Cooperative Association for Internet Data Analysis* (CAIDA), the *Réseaux IP Européens* (RIPE) community, and the *High Energy Physics* (HEP) community. The book concludes with a recap of trends, concerns, and emerging questions in Internet measurement.

Synopsis

The authors have blended their academic research and practical experience in Internet measurement and traffic modeling to provide the reader with a structured view to these vast subjects. I would have liked to see a more extensive coverage of *Voice over IP* (VoIP) and its associated performance measurement protocols, *RTP Control Protocol* (RTCP), RTCP *Extended Report* (XR), and RAQMON, given the gradual but inevitable shift of voice traffic from the *Public Switched Telephone Network* (PSTN) to the Internet with *Session Initiation Protocol* (SIP) peering.

Most probably, this book had already been published when the *Federal Communications Committee* (FCC) issued an order in May 2006 for all VoIP service providers to demonstrate compliance with the *Communications Assistance for Law Enforcement Act* (CALEA) wiretapping requirement within a year. This directive represents a notable departure from data anonymization principles covered in the book.

Overall, I consider this book an excellent reference source for diverse research and practical articles published in the field of Internet measurement. I highly recommend it to network planners, engineers, and managers responsible for instrumentation, traffic modeling, or performance analysis.

—Reza Fardid, Covad Communications
rfardid@covad.com

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

Bob Braden and Joyce K. Reynolds receive the 2006 Postel Service Award

Bob Braden and Joyce K. Reynolds are this year's recipients of the Internet Society's prestigious *Jonathan B. Postel Service Award*. The award was presented "For their stewardship of the RFC (*Request for Comments*) series that enabled countless others to contribute to the development of the Internet." The presentation was made by Internet pioneer Steve Crocker (a member of this year's Postel award committee and author of the very first RFC) during the 67th meeting of the *Internet Engineering Task Force* (IETF) in San Diego, California.

The award is named after Dr. Jonathan B. Postel to commemorate his extraordinary stewardship exercised over the course of a thirty year career in networking. Between 1971 and 1998, Postel managed, nurtured and transformed the RFC series of notes created by Steve Crocker in 1969. Postel was a founding member of the Internet Architecture Board and the first individual member of the Internet Society, where he also served as a trustee.

"It is a pleasure and an honor for the Internet Society to recognize the contribution of Bob and Joyce to the evolution of the Internet," said Crocker. "Since its humble beginnings, the RFC series has developed into a set of documents widely acknowledged and respected as a cornerstone of the Internet standards process. Bob and Joyce have participated in this evolution for a very long time and have been primarily responsible for ensuring the quality and consistency of the RFCs since Jon's death in 1998."

Joyce K. Reynolds worked closely with Postel, and together with Bob Braden she has been co-leader of the RFC Editor function at the University of Southern California's *Information Sciences Institute* (ISI) since 1998. In this role she performed the final quality control function on most RFC publications. Reynolds has also been a member of the IETF since 1988, and she organized and led the User Services area of the IETF from 1988 to 1998. In her User Services role, she was an international keynote speaker and panelist in over 90 conferences around the world, spreading the word on the Internet.

Bob Braden, who has more than 50 years of experience in the computing field, joined the networking research group at ISI in 1986. Since then, he has been supported by NSF for research concerning NSFnet and the DETER security testbed, and by DARPA for protocol research. Braden came to ISI from UCLA, where he had technical responsibility for attaching the first supercomputer (IBM 360/91) to the ARPAnet, beginning in 1970. Braden was active in the ARPAnet Network Working Group, contributing to the design of the FTP protocol in particular. He also edited the Host Requirements RFCs and co-chaired the RSVP working group.

The Jonathan B. Postel Service Award was established by the *Internet Society* (ISOC) to honor those who, like Postel, have made outstanding contributions in service to the data communications community. The award is focused on sustained and substantial technical contributions, service to the community, and leadership. With respect to leadership, the nominating committee places particular emphasis on candidates who have supported and enabled others in addition to their own specific actions.

Previous recipients of the Postel Award include Jon himself (posthumously and accepted by his mother), Scott Bradner, Daniel Karrenberg, Stephen Wolff, Peter Kirstein, Phill Gross and Jun Murai. The award consists of an engraved crystal globe and $20,000.

ISOC (**http://www.isoc.org**) is a not-for-profit membership organization founded in 1992 to provide leadership in Internet related standards, education, and policy. With offices in Washington, DC, and Geneva, Switzerland, it is dedicated to ensuring the open development, evolution and use of the Internet for the benefit of people throughout the world. ISOC is the organizational home of the IETF and other Internet-related bodies who together play a critical role in ensuring that the Internet develops in a stable and open manner. For over 14 years ISOC has run international network training programs for developing countries and these have played a vital role in setting up the Internet connections and networks in virtually every country connecting to the Internet during this time.

First Internet Governance Forum Meeting Concludes

The inaugural meeting of the *Internet Governance Forum* (IGF) took place in Athens, Greece from October 30 – November 2, 2006. For more information see: **http://www.intgovforum.org**

The Government of Brazil will host the next IGF meeting. It will take place in Rio de Janeiro November 12 – 15, 2007.

ARIN to Provide 4-Byte AS Numbers

On August 30, 2006, the *American Registry for Internet Numbers* (ARIN) Board of Trustees, based on the recommendation of the Advisory Council and noting that the Internet Resource Policy Evaluation Process had been followed, adopted the following policy proposal: "2005-9: 4-Byte AS Number."

Per the implementation schedule contained in the policy (*Number Resource Policy Manual* [NRPM] Section 5.1), commencing January 1, 2007, ARIN will process applications that specifically request 32-bit AS Numbers.

For more information see: **http://www.arin.net/registration**

[Ed. See also: "Exploring Autonomous System Numbers," by Geoff Huston in *The Internet Protocol Journal,* Volume 9, No. 1, March 2006.]

Celebrating the 25th Anniversary of the TCP/IP Internet Standards

Two of the core protocols that define how data is transported over the Internet are now 25 years old. The *Internet Protocol* (IP) and the *Transmission Control Protocol* (TCP), together known as "TCP/IP," were formally standardized in September 1981 by the publication of RFC 791 and RFC 793.

Vint Cerf and Robert Kahn are widely credited with the design of TCP/IP, and many others involved in the ARPANET project made significant contributions. The core of the documents was RFC 675, published in December 1974 by Cerf together with co-authors Carl Sunshine and Yogen Dalal. The subsequent sequence of documents leading up to RFC 791 and 793 benefited from the participation of many people including Dave Clark, Jon Postel, Bob Braden, Ray Tomlinson, Bill Plummer, and Jim Mathis, as well as other unnamed contributors to the definition and implementation of what became the Internet's core protocols.

"We can't yet say that the Internet is mature," says Brian Carpenter, chair of the IETF, "but it's a great tribute to its pioneers that the two most basic specifications that were published a quarter of a century ago are still largely valid today. I hope the IP version 6 standard will do as well."

The *Request For Comments* (RFC) series, which was launched in 1969 by Steve Crocker at UCLA (and edited for many years by the late Jon Postel), continues today as the public archive of the Internet's fundamental technology. Since 1977 it has been hosted by The University of Southern California's *Information Sciences Institute* (ISI). ARPA support ended in 1998, at which time ISOC took over providing funding for the publication of Internet standards. More recently, ISOC extended its support to include other areas critical to the open development of Internet standards.

See also:

http://www.ietf.org/rfc/rfc0791.txt

http://www.ietf.org/rfc/rfc0793.txt

http://www.isoc.org/standards/tcpip25years

http://www.isoc.org/internet/history/brief.shtml

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

March 2007 Volume 10, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

Every time you dial into a service provider network or connect to a wired or wireless network that offers Internet access, you are most likely using several components of what is referred to as *Authentication, Authorization, and Accounting*, or "AAA" for short. The AAA space is quite complex, so when we asked Sean Convery to give us an overview of these technologies, he decided to divide his survey into two parts. Part One—subtitled "Concepts, Elements, and Approaches"—is included in this issue. Part Two, which discusses protocol details and applications, will follow in our next issue.

The *Domain Name System* (DNS) has been discussed previously in this journal. The most critical part of the DNS is the collection of *Root Servers*. For protocol reasons, there are only 13 "logical" root servers, but a system of more than 100 servers has been deployed using a technique known as *anycast*. Steve Gibbard examines the distribution of the root servers in different parts of the world and discusses operational aspects of the DNS.

If you are tracking any part of the IETF process, you should be aware of several important resources. First, the IETF *Education Team* (**http://edu.ietf.org/**) offers training sessions and educational materials. Second, the *IETF Journal* (**http://www.isoc.org/ietfjournal**) publishes timely reports and updates on the activities of the IETF.

Finally, the IETF *Tools Team* (**http://www.ietf.org/tools.html** and **http://tools.ietf.org**) provides many tools and applications for protocol developers. Marshall Rose and Carl Malamud take a closer look at one of these tools, namely a system for writing Internet Drafts and RFCs using XML.

Please take a moment to renew and update your subscription. You can access your subscription record by clicking on the "Subscriber Services" link at **http://www.cisco.com/ipj**.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Network Authentication, Authorization, and Accounting
Part One: Concepts, Elements, and Approaches

by Sean Convery, Identity Engines

Network *Authentication, Authorization,* and *Accounting* (AAA, pronounced "triple-A") is a technology that has been in use since before the days of the Internet as we know it today. Authentication asks the question, "Who or what are you?" Authorization asks, "What are you allowed to do?" And finally, accounting wants to know, "What did you do?" These fundamental security building blocks are being used in expanded ways today. This article, the first in a two-part series, focuses on the overall concepts of AAA, defines the elements involved in AAA communications, and discusses high-level approaches to achieving specific AAA goals. Part two of the article, to be published in a future issue of IPJ, will discuss the protocols involved, specific AAA applications, and considerations for the future of AAA.

AAA, at its core, is all about enabling mobility and dynamic security. Without AAA, a network must be statically configured to control access; IP addresses must be fixed, systems cannot move, and connectivity options should be well defined. Even the earliest days of dialup access broke this static model, thereby requiring AAA. Today, the proliferation of mobile devices, diverse network consumers, and varied network access methods combine to create an environment that places greater demands on AAA.

AAA has a part to play in almost all the ways we access a network today. Emerging technologies such as *Network Access Control* (NAC) extend AAA even into corporate Ethernet access (historically the "trusted" network that set the benchmark level of security that all other types of access had to match). Today, wireless hotspots need AAA for security, partitioned networks require AAA to enforce segmentation, and remote access of every kind uses AAA to authorize remote users.

It is not clear when the term AAA first gained acceptance, but an examination of academic papers finds "authentication, authorization, and accounting" used as a discrete term (albeit without the AAA acronym) as early as 1983 in an IEEE paper[1]. Though mired in pre-Internet *Open Systems Interconnection* (OSI)-centric terminology, the ordering of the "A's" is the same as today's usage.

For most network administrators, the genesis of AAA coincided with the development of the *Remote Authentication Dial-In User Service* (RADIUS) protocol[2]. RADIUS was developed by Livingston Enterprises (now part of Alcatel-Lucent) in the early 1990s, became an Internet standard through the IETF in 1997, and today is the most widely accepted AAA protocol.

Another widely adopted AAA protocol, which predates RADIUS as an RFC by four years, is the *Terminal Access Controller Access Control System* (TACACS)[3]. Though never an Internet standard, TACACS evolved into XTACACS and then TACACS+, the latter of which is the only version of TACACS in use today.

Before we delve into the details of these protocols, it is important to understand the roles played within a AAA system.

Core Components of AAA

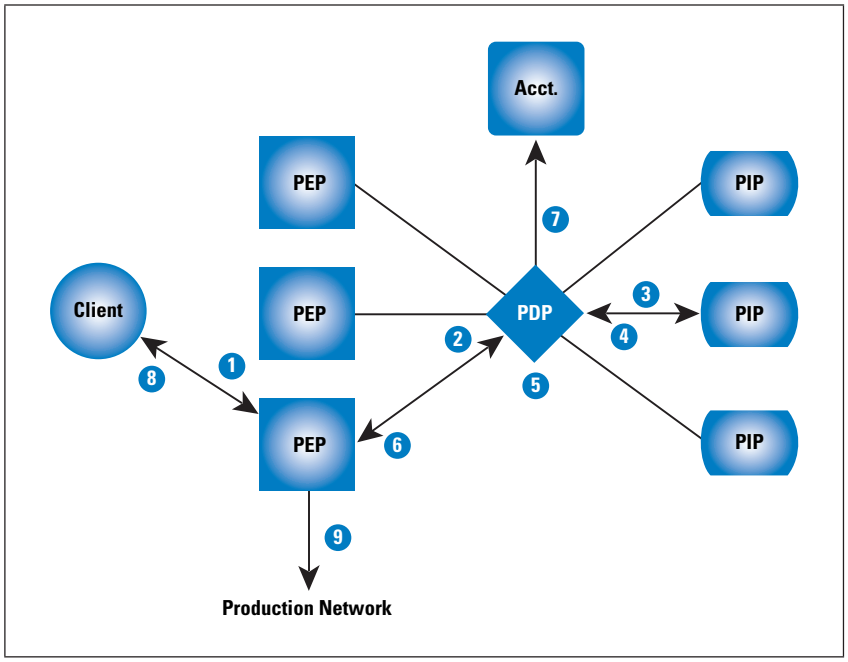
- *Client:* The client is the device attempting to access the network. The client either authenticates itself, or it acts as a proxy to authenticate the user.
- *Policy Enforcement Point (Authenticator):* The Policy Enforcement Point (PEP) is sometimes called the authenticator or *Network Access Server* (NAS). The PEP is the network device that brokers the access request for the client. The PEP can be a dial-in server, VPN concentrator, firewall, gateway *General Packet Radio Service* (GPRS) support node, Ethernet switch, wireless access point, or an inline security gateway. The PEP is responsible for enforcing the terms of a client's access. This enforcement varies based on the capabilities of the PEP and is discussed later in this article.
- *Policy Information Point:* The Policy Information Point (PIP) is a repository of information to help make the access decision. It could be a database of device IDs, a user directory such as the *Lightweight Directory Access Protocol* (LDAP), a *one-time password* (OTP) token server, or any other system that houses data relevant to a device or user access request.
- *Policy Decision Point (AAA Server):* The Policy Decision Point (PDP) is the brain of the AAA decision. It collects the access request from the client through the PEP. It also queries any relevant PIPs to gather the information it needs to make the access decision. The PDP, as its name implies, is the entity that makes the final decision around network access. It also can send specific authorizations back to the PEP that apply settings or constraints to the client's network traffic.
- *Accounting and Reporting System:* Whether on a dedicated system or built as part of a PDP, tracking use of the network with accounting is one of the best features of AAA. With all forms of network access now offering controlled access, the AAA service can tell you who got on the network, from where, and what that person was granted access to.

It is important to note that the preceding categories are logical containers of functions and not necessarily dedicated physical devices. Often elements are combined, such as PEP with PDP, and PDP with PIP.

Example AAA Flow

Now that we have examined the components of a AAA solution, walking through a typical use case will help cement our understanding of the role that each entity plays. Figure 1 shows an example of a client attempting to gain access to the network.

Figure 1: A Client Connects to a AAA-Protected Network



1. The client attempts to connect to the network, is challenged for identity information, and sends this information to the PEP. In this example, let's assume the client is a laptop with a worker attempting to access an organization's VPN from a remote location. Additionally, we'll assume this is a valid, permitted use of the network.
2. The PEP sends the collected identity information to the PDP. In some cases (discussed in part two of this article), the PEP cannot see the specific identity information provided but instead relays the information directly to the PDP.
3. The PDP queries any configured PIPs for information about the client and validates that the credential provided by the client is valid. In this example, the PIP is an LDAP directory.
4. The PIP returns a success or failure message from the credential validation step and sends additional information about the client to the PDP for evaluation. This information could include the role of the user, the home location for the user, and so on.
5. The PDP evaluates information learned about the client through the client, PEP, and PIP; the role of the PEP and PIP that serviced the request; and any contextual information (such as time of day) against its configured policies. Based on this information, the PDP makes an authorization decision.

6. The PDP sends the PEP the authentication result and any authorizations specific to the client. These authorizations trigger specific PEP actions to apply to the client. For example, the authorization data might trigger specific *Access Control Lists* (ACLs) or IP pool assignments for the client.
7. The PDP also sends the result of this transaction to the accounting system.
8. The PEP applies the authorization profile learned from the PDP and sends the "authentication successful" message to the client. The PEP can also be configured to send accounting information on this new connection to the accounting and reporting system.
9. The client accesses the production network through the PEP.

Elements of Authentication

When performing authentication, numerous elements can be evaluated before a PDP reaches its access decision. At a high level, these elements can be broken down into three categories: the principal itself (the user, device, or service requesting access), the credential the principal submits (shared key, one-time password, digital certificate, or biometric credential), and the contextual information describing the transaction (location, time of day, software state, and so on).

- *Principal:* The principal is the entity requesting authorization. It is generally some combination of user, device, or service. When concerned with a user, the PIP can provide attributes about the user such as role or group affiliations, job title, e-mail address, physical address, and so on. In specific applications, it can include much more granular information. For example, a higher-education facility might be interested in knowing a student's class schedule when servicing the student's authentication request. When the principal is a device, the same thinking applies. The PIP can inform the PDP if the device is a managed asset, what its basic usage parameters are, and so on. User and device authentication can be carried out sequentially for the same transaction, often involving device authentication first and then user authentication. Lastly, a service such as a network management process can authenticate. In this case, the service almost always looks like a user to the AAA infrastructure and is handled accordingly.
- *Credential:* The next element the PDP considers is the credential the user or device submits as proof of identity. There are four main types of credentials: shared key (password), *one-time password* (OTP), digital certificate, and biometric credential. This section examines each of these types. The first and most widely used form of credential is the shared key, typically a user password. AAA deployments that use shared keys can be subdivided based on the protocol the system uses to verify the password, including the *Password Authentication Protocol* (PAP)[4], *Challenge Handshake Authentication Protocol* (CHAP)[5], and *Microsoft CHAP Extensions* (MS-CHAP) Versions 1[6] and 2[7]. PAP authentication is a plaintext authentication method that is not recommended for use in security-sensitive environments.

However, many newer protocols provide a secure transport for PAP, making its use in AAA still quite common. Some of these methods are discussed in part two of this article. CHAP improves on the security of PAP by not sending the password in the clear but rather a challenge based on a hash of the password. MS-CHAP is a Microsoft extension to CHAP that tunes things a little bit for Microsoft environments. Version 2 of MS-CHAP addresses security weaknesses in Version 1. MS-CHAPv2 is quite common today in Microsoft environments. CHAP in all its forms is vulnerable to dictionary attacks because even if a hash cannot be decrypted, common passwords can be guessed and those hash values can be computed.

A second, also widely used credential type is the OTP. At login time, users refer to their personal token to get the OTP they will type in. The token is generally provided in hardware or software form. Tokens are designed to generate seemingly random passwords that are synchronized with a token server acting as a PIP. The OTP can be sent in the clear because it is used only once; after a configurable time (for example, 30 seconds) a new password is generated. When an OTP is combined with a *Personal Identification Number* (PIN), two-factor authentication is achieved because the client needs to have something (the token) and know something (the PIN).

The third type of credential is the *digital certificate*. Digital certificates can be stored either locally on the client or on some sort of removable device such as a smartcard. A full discussion of asymmetric-key cryptography is outside the scope of this article, but at a high level, certificates work by asserting the identity of their bearer by having the certificate signed by a trusted *Certificate Authority* (CA). CAs can be external entities such as a government or commercial enterprise or they can be internal to a given organization. The certificate itself can be freely distributed, because the only way it can be validated as belonging to the rightful owner is in combination with the private key. Because they reside on the client, certificates are most often used to authenticate a physical entity rather than an individual. However, smartcards are changing this paradigm by enabling users to take their digital certificate (and private keys) with them, thereby disassociating the certificate from the machine itself. Similar to an OTP without a PIN, a digital certificate or smartcard alone does not provide two-factor authentication. Certificate deployments, particularly smartcards, are addressing this problem by requiring a PIN to unlock access to the credential.

The fourth and least widely deployed type of credential is the *biometric credential*. Biometrics[12] ignores something you *have* and something you *know* and instead focus on something you *are*. Fingerprint scanners, iris scanners, and facial recognition are all forms of biometric authentication. Because biometrics is the newest form of credential, it is currently experiencing heightened anticipation among users regarding potential applications—and also scrutiny for potential weaknesses.

- *Contextual:* The last element the PDP typically considers in its authentication decision is the contextual information associated with the AAA request, including the network and physical location of the request, the type of access provided by the PEP, the time of day, and potentially other elements such as network load, security threat level, and so on. A relatively new entrant into this set of contextual information is client device posture, typically discussed under the rubric of *Network Access Control* (NAC). NAC or posture checks examine the software state of the client before it connects. NAC data allows the PDP to assess the degree of risk posed by the connecting client before granting the client access to the network. For example, if a system is running an out-of-date operating system, has no current security applications running, or otherwise exhibits high-risk behavior, it may not be granted access to the network. NAC will be discussed in more detail in part two of this article.

Authorization Approaches

At its core, authorization means determining what a client is allowed to do on the network. However, the granularity of this authorization is only as good as the sophistication of the PDP and the enforcement capabilities of the PEP. This section examines the authorization options for network AAA, including Layer 2 segmentation, Layer 3 filtering, and Layer 7 entitlements. It closes with an examination of some of the challenges encountered when sending or "provisioning" the authorizations from the PDP to the PEP.

- *Null Authorization (Authentication Only):* Strangely the most common authorization in AAA is no authorization at all. After the authentication event occurs, the client is immediately granted full access to the network. This characteristic is a holdover from the original goal of remote-access AAA: to perform an authentication check that simply determines whether the client should be trusted as if it were connected to the organization's home network. Because these home networks employed no segmentation or filtering within them, it was natural that remote-access techniques such as dialup and VPN would likewise employ neither. Today however, authentication is increasingly being used for all forms of network access, with a goal of providing clients with network rights commensurate with their role in the organization. This latter goal requires a strong authorization foundation through the cooperation of the PDP and PEP.
- *Layer 2 Segmentation:* For wireless access points and Ethernet switches, the most common form of authorization enforcement is Layer 2 segmentation, which works by splitting the network into multiple logical segments, isolating certain classes of client from one another. This process is most typically achieved by deploying *Virtual LANs* (VLANs), which separate the members of one VLAN from other VLANs in the same Layer 2 network—even though the VLANs traverse the same physical network infrastructure.

VLANs can be used to restrict access to specific resources by working in coordination with VLAN-specific ACLs on Layer 3 devices upstream from the Layer 2 device. For access points, a given wireless *Service Set Identifier* (SSID) can be associated with a VLAN on the wired side of the access point. *Multiprotocol Label Switching* (MPLS) is more commonly associated as a WAN transport, but there is nothing to prevent labels for traffic based on AAA. More commonly, the client is associated with a VLAN and the VLAN is associated with an MPLS label further into the infrastructure.

- *Layer 3 Filtering:* Layer 3 filtering authorizes access to resources through ACLs configured on Layer 3 devices (routers, Ethernet switches, security gateways, and so on). These ACLs (which generally encompass Layer 4 of the OSI stack as well) can enforce authorizations to a range of hosts, specific hosts, or services on those hosts. As mentioned earlier, Layer 3 filtering can be combined with Layer 2 segmentation to provide aggregate authorizations for an entire VLAN. This filtering is the most common technique on network infrastructure devices, whereas security gateways tend to apply ACLs to specific clients. Additionally, technologies such as *IP Security* (IPsec)[8] provide a Layer 3 filtering capability by allowing only certain types of traffic to travel through the VPN tunnel.
- *Layer 7 Entitlements:* Increasingly, security gateways are able to go beyond Layer 3 and 4 filtering and are starting to become application-aware, meaning that the authorizations handed from the PDP to the PEP can be very granular, focusing on the specific applications that are needed rather than broader filters based on segments or hosts on the network. Because this technology is still relatively new, there are no standards yet to make this interaction work transparently. As a result, most granular application filters are written on the PEP itself in order to allow the PDP to trigger a preexisting profile on the PEP. These sorts of provisioning challenges are discussed further in the next section.
- *Provisioning Challenges:* In AAA parlance, the term "provisioning" refers to communicating a user's session rights and constraints to the PEP so that the PEP can grant and enforce these permissions. One of the most difficult aspects of provisioning access rights on a PEP is communicating the decision of the PDP in a format the PEP can understand. This fact is one of the reasons that many PEPs come with a lightweight PDP. This approach solves the narrow problem for that PEP but creates management challenges when coordinating network AAA across a broader enterprise, because the enterprise AAA policies must be implemented individually on each unique type of PEP on the network. Because RADIUS is the most commonly used network AAA protocol, it is natural to communicate the PDP decision using that protocol. RADIUS attributes such as the "filter-id" allow the PDP to trigger a preexisting filter on the PEP.

In addition, many PEP vendors support *Vendor Specific Attributes* (VSAs) in RADIUS to enable the PDP to speak the language of the PEP more specifically. This process works well but creates a significant amount of work on the PDP to enable it to translate the policy result and correctly communicate it to each type of PEP. Another option soon to be sanctioned by the standards bodies is an extension to RADIUS that enables the sending of standard IP ACLs using RADIUS attributes[9].

One further option for provisioning is through the *Simple Network Management Protocol* (SNMP), which is typically used to assign Layer 2 ports to VLANs or to enable or disable interfaces. This process can work, but remember that the version of SNMP typically in deployment is still SNMPv2c, which is *User Datagram Protocol* (UDP)-based (connectionless) and unencrypted. Therefore, the SNMP traffic is prone to packet loss when links are congested or devices are busy, thereby requiring costly application layer retransmission schemes. It also means the transmissions themselves are vulnerable to inspection or modification. These attributes make SNMP generally a poor choice for security-sensitive tasks. RADIUS also uses UDP, but supports basic retransmission as part of the protocol.

Another provisioning method used today is standard *Secure Shell* (SSH) Protocol or HTTPS-based configuration. This method manages a device through standard administrative interfaces to set enforcement techniques. Although this method gives the PDP full access to the features of the PEP, it is very difficult to coordinate the dynamic aspects of the client AAA event with the static elements of the running configuration of the PEP. Finally, new protocols are emerging to make provisioning easier. NETCONF[10] is an *Extensible Markup Language* (XML)-based protocol designed as a replacement for network management applications connecting to devices over the *command-line interface* (CLI).

As this section has shown, there are numerous approaches to authorization in AAA. Each PEP has its own capabilities, but the challenge for a diverse network is to consistently authorize clients, regardless of the given PEP they access the network through.

Accounting Techniques

Accounting is an increasingly critical step in the overall AAA process. Regulatory controls are starting to mandate better auditing of network access. The last stage of AAA, accounting simply records which clients accessed the network, what they were granted access to, and when they disconnected from the network. Accounting has always been widely used in the *Internet Service Provider* (ISP) space because auditing network access is the basis for billing ISP customers. Increasingly, accounting is being used as a way to correlate client attribute information (username, IP address, etc.) with actions and events on the network.

This correlation can make other systems that are not user-aware more intelligent in the security decisions that they make. For example, a network *Intrusion Detection System* (IDS) can learn a lot about the behavior of a given IP address. However, when that information is correlated with the user assigned to that IP address—and the permissions that user should have—the relevance of the IDS data increases dramatically.

One of the design considerations of accounting systems is that, given the centralized nature of audit and the decentralized nature of access, they are generally out-of-band with the client's normal communications. This makes them excellent resources to refer to when the network administrator wants to know when the client connected and what the client was granted access to. However, their out-of-band nature makes them poor resources for determining what the client actually did while connected to the network. This information can be learned by the network, as mentioned earlier, by coordinating the AAA accounting information with the rest of the network enforcement and monitoring systems.

Summary and Part Two Teaser

This first part of this article introduced AAA and described many of the foundation concepts necessary to gain a sound understanding of the overall system. After defining the elements involved, a sample flow of a AAA event was described. Additionally, the high-level approaches to authentication, authorization, and accounting were discussed. Part two of this article will discuss the protocols used in AAA, including not just RADIUS, *Extensible Authentication Protocol* (EAP), TACACS+, and Diameter, but many others. Additionally, specific applications of AAA technology will be described, and some conclusions will be drawn as to what the future holds for AAA.

References

[1] Lagsford et. al., "OSI Management and Job Transfer Services," *Proceedings of the IEEE,* Volume 71, No. 12, December 1983.

[2] Rigney et. al., "Remote Authentication Dial In User Service (RADIUS)," RFC 2865 (Obsoletes RFC 2138, 2058), June 2000.

[3] Finseth C., "An Access Control Protocol, Sometimes Called TACACS," RFC 1492, July 1993.

[4] Lloyd et. al., "PPP Authentication Protocols," RFC 1334, October 1992.

[5] Simpson W., "PPP Challenge Handshake Authentication Protocol (CHAP)," RFC 1994, August 1996.

[6] Zorn et. al., "Microsoft PPP CHAP Extensions," RFC 2433, October 1998.

[7] Zorn et. al., "Microsoft PPP CHAP Extensions, Version 2," RFC 2759, January 2000.

[8] Kent et. al., "Security Architecture for the Internet Protocol," RFC 2401, November 1998.

[9] Congdon et. al., "RADIUS Filter Rule Attribute," Internet Draft, Work in Progress, January 2007, **draft-ietf-radext-filter-08.txt**

[10] Enns et. al., "NETCONF Configuration Protocol," RFC 4741, December 2006.

[11] Dory Leifer, "Visitor Networks," *The Internet Protocol Journal,* Volume 5, No. 3, September 2002.

[12] Edgar Danielyan, "The Lures of Biometrics," *The Internet Protocol Journal,* Volume 7, No. 1, March 2004.

SEAN CONVERY is CTO at Identity Engines, a venture-backed startup developing innovative identity management solutions for enterprise networks. Prior to Identity Engines, Sean (CCIE® no. 4232) worked for seven years at Cisco Systems, most recently in the office of the security CTO. Sean is best known as the principal architect of the SAFE Blueprint from Cisco and the author of *Network Security Architectures* (Cisco Press, 2004). Sean has presented to or consulted with thousands of enterprise customers around the world on designing secure networks. Before Cisco, Sean held various positions in IT and security consulting during his 14 years in networking. E-mail: **sconvery@idengines.com**

Geographic Implications of DNS Infrastructure Distribution

by Steve Gibbard, Packet Clearing House

The past several years have seen significant efforts to keep local Internet communications local in places far from the well-connected core of the Internet. Although considerable work remains to be done, Internet traffic now stays local in many places where it once would have traveled to other continents, lowering costs while improving performance and reliability. Data sent directly between users in those areas no longer leaves the region. Applications and services have become more localized as well, not only lowering costs but keeping those services available at times when the region's connectivity to the outside world has been disrupted. I discussed the need for localization in a previous paper, "Internet Mini-Cores: Local connectivity in the Internet's spur regions."[1] What follows here is a more specific look at a particular application, the *Domain Name System* (DNS).

Most Internet applications depend on the DNS, which maps human-readable domain names to the *Internet Protocol* (IP) addresses computers understand. Two Internet hosts may have connectivity to each other but be unable to communicate because no DNS server can be reached. This article examines the placement of DNS servers for root and top-level domains and the implications of that placement on the reliability of the services these servers provide in different parts of the world. It is not a "how-to" guide to the construction of DNS infrastructure and does not contain recommendations on DNS policy; it is rather a look at the placement of DNS infrastructure as currently constructed.

Although it is possible to access Internet resources without the DNS by entering numeric IP addresses directly, this type of access is not generally done. IP addresses, such as **`209.131.36.158`**, are difficult to remember, are generally unpublished outside the DNS, and often change without notice. Local caching of DNS information can mask temporary problems with DNS data for commonly accessed domain names, but caches are emptied when caching resolvers are restarted, data in caches expires, and nothing is cached until the first time it is accessed by a local user.

It should be noted that information about DNS deployment is changing rapidly. Several organizations are working on new DNS deployment. Information in this article can be considered current, to the best of my knowledge, as of May 2006.

DNS Hierarchy

The DNS is a hierarchy of domains within domains. The levels of the hierarchy are separated by dots. At the top of the hierarchy is the *root,* usually invisible but sometimes represented as a trailing dot. Using **www.yahoo.com** as an example, the **com** domain is contained within the root. **Com** contains **yahoo**, and **yahoo** contains **www**. Domains in the position **com** takes in this example are known as *Top-Level Domains,* or TLDs; they are the first level in the root domain. Domains in the position of **yahoo** are known as *Second-Level Domains*. In this example, **www** occupies the third level, and so forth.

The information that makes up the Domain Name System is stored on DNS *servers*. That information is divided into *zones,* which for our purposes are synonymous with domains. Each zone is stored on a set of *authoritative servers,* which are queried when users or applications attempt to access a service on the Internet. In the simplest case, a domain name query works like the following:

A *caching resolver* (so named because it caches information it receives) that has not yet cached any DNS zone data receives a query for **www.yahoo.com**. Because its cache is empty, it uses the *hints* distributed with the DNS software to contact one of the root servers and asks, "Where is **www.yahoo.com**?" The root server replies with a list of servers for **.com**. The caching server then asks one of the **.com** servers, "Where is **www.yahoo.com**?" and gets a response that directs it to servers for **yahoo.com**. It asks the same question of those servers and finally gets an answer to the question it was asking.

Generally several servers can answer questions about any domain, but if all the servers for any single level are broken or unreachable, the query fails and the service the user is looking for is unreachable. It is therefore important that the DNS be reliable, and that the servers for each zone throughout the hierarchy be reachable from anywhere the servers they point at are being used.

Root Servers

Without root servers, none of the DNS works. As of this writing, 117 root servers exist worldwide, operated by 12 different organizations.[2] Root servers are added frequently, so the number may be significantly greater by the time this article is in circulation.

Because of protocol limitations, the root servers can use only 13 IP addresses. Each root-server operator is responsible for one or two of those addresses. Using a technique called *anycast,* which allows servers in separate locations to share a *single* IP address, six of those operators operate multiple servers using the same IP address[3], meaning that only 13 of them are visible at the same time from any single location, but those 13 should in most cases include the topologically closest one.[4]

The distribution of root servers is rather uneven. North America and Europe have similar numbers: 38 in North America and 35 in Europe. The 35 in Europe are distributed fairly evenly, with the largest concentrations (four each) in London and Amsterdam, Europe's two largest Internet hubs. North America has 8 in Washington, D.C., 8 in the San Francisco Bay Area, and 5 in Los Angeles. In the United States, all cities that host root servers are on the coasts except Atlanta and Chicago. All seven of the remaining IP addresses represented by only a single server, known as *unicast roots,* are in the Washington, D.C., San Francisco, and Los Angeles areas.

Australia has two root servers in Brisbane, one in Perth, and one in Sydney. New Zealand has two, one in Wellington and one in Auckland. Singapore and the wealthier parts of East Asia are well-covered, and there are two root servers in Jakarta and one each in Bangkok and Kuala Lumpur. A year ago, there were none in the vast expanse between Bangkok and Dubai, but three have recently been added in India, along with others in Dhaka and Karachi. Mainland China and the former USSR each have two. There are three in Africa: two in Johannesburg and one in Nairobi. Another will be installed in Nairobi shortly, but most of the rest of Africa lacks direct connectivity to Johannesburg or Nairobi and must cross satellite or intercontinental fiber links to reach the nearest root servers. All four of the root servers in South America are in Brazil and Chile, with two in Sao Paulo and one each in Brasilia and Santiago de Chile.

With some exceptions, root-server density tends to correlate strongly with per-capita income. This fact is not surprising—it is true for other forms of infrastructure as well—but it means that those with the greatest dependence on external infrastructure are those least able to pay for external connectivity.

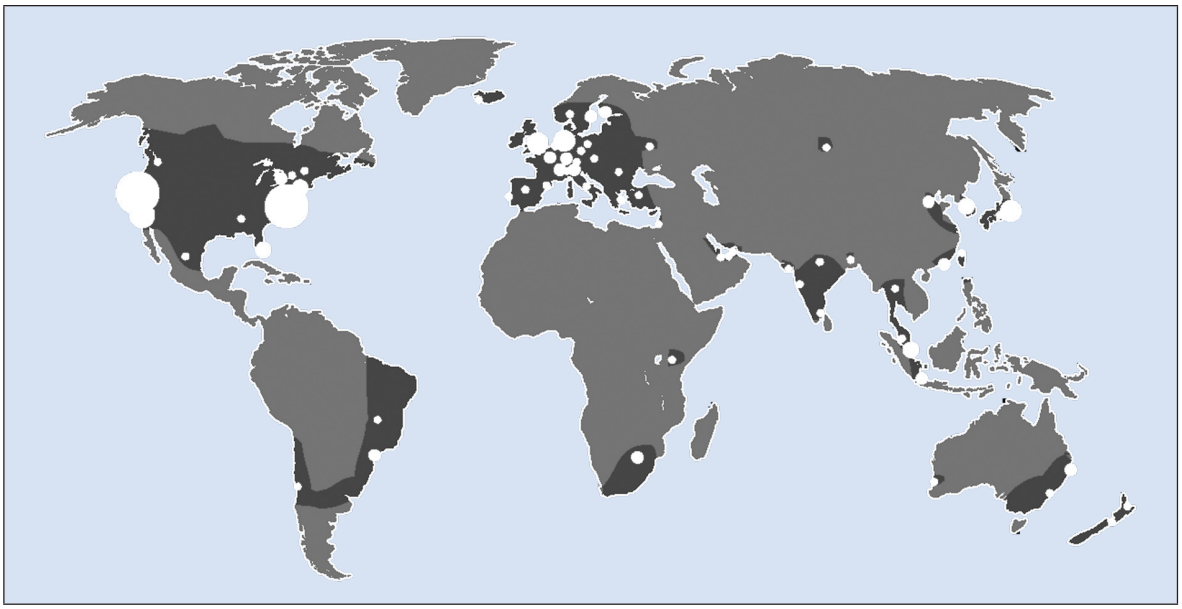
Root-Server Placement

In areas that have local root servers, finding the name servers for a top-level domain should be fairly reliable. In areas without local root servers, the ability to query the root servers is dependent on other long-distance infrastructure. In some places this infrastructure is well-developed, so this problem is not a significant one. Elsewhere long-distance infrastructure is slow, expensive, and unreliable, consisting of satellite links or a single fiber connection that may take several days to fix if it breaks.

Sri Lanka, for example, is connected to the rest of the world by a single fiber connection, which was cut in 2004 by a ship that dragged anchor in the Colombo harbor.[5] Although Sri Lanka has an exchange point that should have allowed connectivity to local Internet services, news reports said that Sri Lanka's "Internet and long-distance phone service" had been cut off. I have not received a good account of what Internet connectivity looked like from anyone in Sri Lanka at the time, but it is likely that even local Internet connections would not have worked without a local root server.

Sri Lanka is not an isolated case. The dots in Figure 1 show the locations of all root servers. The light grey areas are regions in which multiple fiber paths are available to root servers. The remainder of the world can reach root servers only by a single fiber path or by satellite. Large areas of the world are poorly covered.

Figure 1: Root Server Locations and Areas of Redundant Connectivity



Root-Server Expansion

Four of the 12 root-server operators are presently working to install root servers in areas that lack them. Although the 117 root servers currently in operation are a big improvement over the 13 that were in operation three years ago, many regions still do not have any. Those root-server operators are installing servers wherever they can get the funding to do so.

Funding is generally provided either by grants, especially from the *Asia Pacific Network Information Centre* (APNIC) in the Asia-Pacific region, by local governments or *Internet Service Provider* (ISP) associations. Because the addition of new anycast copies of root servers is relatively easy given sufficient funding, the main limitation preventing the installation of root servers in new locations is lack of funding.

One question probably best addressed in a more central manner is whether it makes sense to have many copies of one or two root-server IP addresses in some regions or whether it would be better to have more of a mix of root-server IP addresses. Currently, only 6 of the 13 root-server addresses are anycasted, only 4 are anycasted in large numbers, and 2 of those focus on specific regions, meaning that in many of the more remote parts of the world the only nearby root servers are *Internet Systems Consortium* (ISC)'s "F" and Autonomica's "I" roots, and some places have several of one of those closer than the next one of the other.

Because some DNS resolvers have their own mechanisms for finding the closest server and for handling failures of types that do not include route withdrawals, having multiple IP addresses nearby seems like a good thing. A more complex question is whether it would be worthwhile to anycast all 13 of them widely, or if there is some smaller number that would be sufficient to have nearby. Previous research on this topic has assumed a limit of 13 root servers, producing conclusions that are not applicable to the modern Internet.[6]

This article should not be seen as a criticism of the places with large numbers of root servers. Although the U.S. distribution looks strange, with the San Francisco Bay and D.C. area clusters perhaps excessive, it comes close to following the Internet topology in the United States. Indeed, the U.S. concentration may be appropriate to handle server load. Western Europe's dense but relatively even distribution of root servers through the region appears to be an optimal distribution, because most populated areas have multiple root servers nearby. Likewise, Jakarta is one of the very few cities in the developing world to have more than one, and that provides local redundancy that much of the developing world lacks. If root-server deployment were funded from a single global budget, the distribution across the world's regions would look very unfair. But because Internet infrastructure is mostly funded locally, Jakarta and Western Europe are examples other regions could emulate.

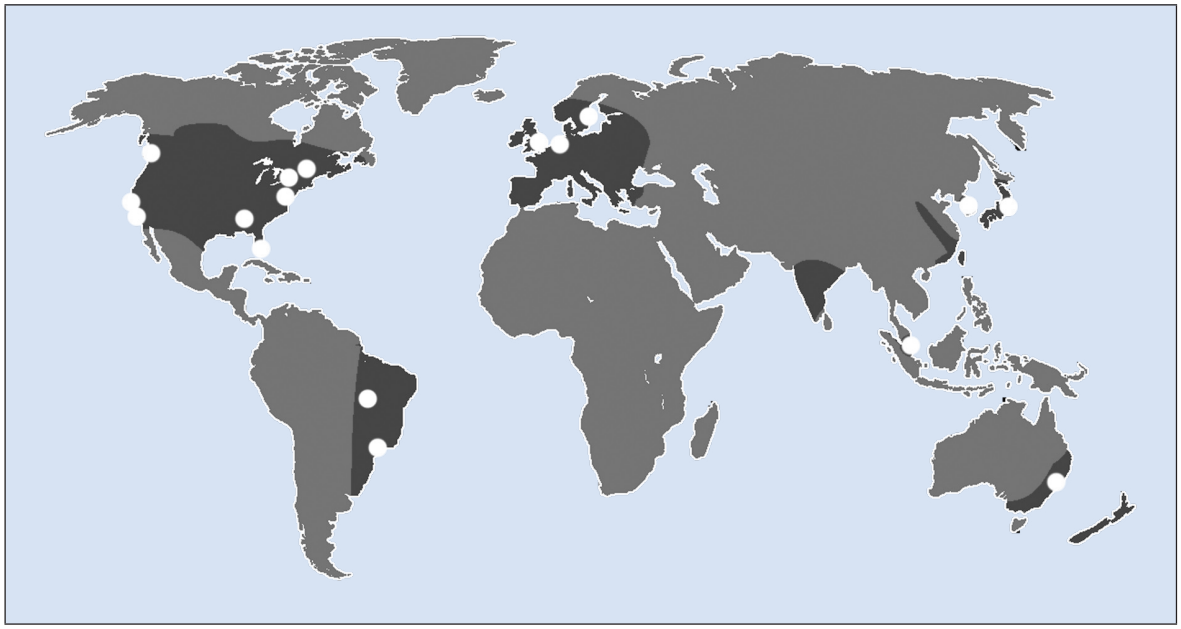
TLD Distribution

Use of the DNS also requires access to TLD servers. To access something in the **.com** domain, a user's local DNS resolver must be able to reach the **.com** servers. This statement is true for any TLD, whether it is a *generic TLD* (gTLD), such as **.com**, **.net**, and **.org**, or a *country code TLD* (ccTLD). Unlike the root, it is not necessary that all TLDs be reliable from all locations; if a TLD is not used to name local resources in a region, having local access to that TLD will not help if that the region gets cut off from the rest of the world.

gTLD Distribution

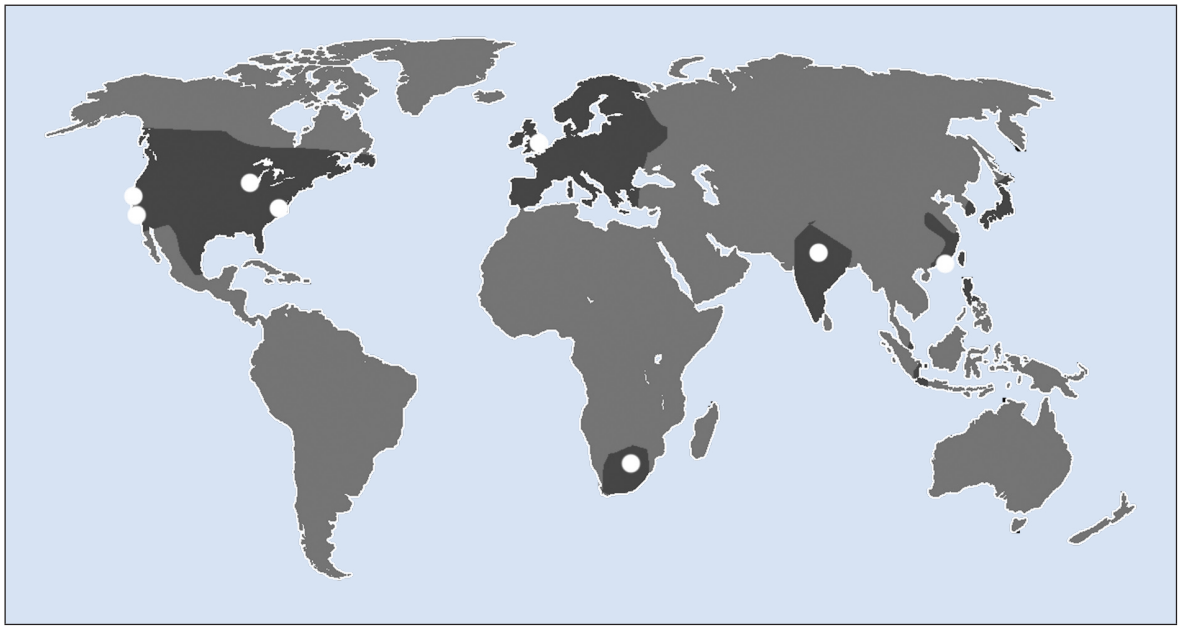
Of the gTLDs, **.com** is by far the largest. It is well-connected to the Internet core, the area with well-meshed internal connectivity mainly comprising North America, Western Europe, East Asia, and Singapore. (See Figure 2.) The **.com** servers are located in Australia, Brazil, Japan, South Korea, the Netherlands, Sweden, the United Kingdom, and the U.S. states of California, Florida, Georgia, Virginia, and Washington. The **.com** servers are well-connected to areas well-connected to those regions but poorly connected to Africa, South Asia, and parts of South America.

Figure 2: Server Locations for .com and .net and Areas of Redundant Connectivity



UltraDNS, the operator of **.org**, **.info**, **.mobi**, and **.coop**, among others, is also somewhat well-connected to the Internet core, although not to the extent the **.com** servers are. It has publicly accessible servers in four metropolitan areas in the United States as well as in London and Hong Kong. It has a couple of noncore locations, in Delhi and Johannesburg. UltraDNS also has servers in other locations, accessible only to the resolvers of certain large ISPs. Because those servers are not available to the general public in their regions, they are omitted from discussion here. (See Figure 3.)

Figure 3: Server Locations for .org, .info, and .mobi and Areas of Redundant Connectivity



Other gTLDs do not do considerably better. Table 1 shows the locations of all the gTLDs.

Table 1: Locations of TLD Servers

gTLD	Locations by Country or U.S. State
.aero	Switzerland, Germany, India, Hong Kong, United Kingdom, and the following states in the United States: California, Illinois, and Virginia
.biz	Australia, Hong Kong, Netherlands, New Zealand, Singapore, United Kingdom, and the following states in the United States: California, Florida, Georgia, New York, Virginia, and Washington
.com	Australia, Brazil, Canada, Japan, South Korea, Netherlands, Sweden, Singapore, United Kingdom, and the following states in the United States: California, Florida, Georgia, Virginia, and Washington
.coop	United Kingdom and the following states in the United States: California, Illinois, and Massachusetts
.edu	Netherlands, Singapore, and the following states in the United States: California, Florida, Georgia, and Virginia
.gov	Canada, Germany, and the following states in the United States: California, Florida, New Jersey, Pennsylvania, and Texas
.info	India, Hong Kong, South Africa, United Kingdom, and the following states in the United States: California, Illinois, and Virginia
.int	Netherlands, United Kingdom, and California in the United States
.jobs	Netherlands, Singapore, and the following states in the United States: California, Florida, Georgia, and Virginia
.mil	The following states in the United States: California, Maryland, Virginia, and other unknown locations
.mobi	India, Hong Kong, South Africa, United Kingdom, and the following states in the United States: California, Illinois, and Virginia
.museum	Sweden and California in the United States
.name	Singapore, United Kingdom, and the following states in the United States: California, Florida, Georgia, Virginia, and Washington
.net	Australia, Brazil, Canada, Japan, South Korea, Netherlands, Sweden, Singapore, United Kingdom, and the following states in the United States: California, Florida, Georgia, Virginia, and Washington
.org	India, Hong Kong, South Africa, United Kingdom, and the following states in the United States: California, Illinois, and Virginia
.pro	Canada and the following states in the United States: Illinois and Texas
.travel	Australia, Hong Kong, Netherlands, New Zealand, Singapore, United Kingdom, and the following states in the United States: California, Florida, Georgia, New York, Virginia, and Washington

Although gTLDs are typically marketed for their applicability to specific types of organization, or in the case of **.com** because it is the only domain many people have heard of, geography should also be considered in selecting domains. Most of the gTLDs have reasonable coverage throughout the Internet core region, but there are exceptions. The **.int** and **.museum** domains are hosted only in North America and Europe, and **.pro** is hosted only in North America.

Outside the Internet core there is little gTLD presence. Only **.biz**, **.travel**, **.com**, and **.net** are present in Australia and New Zealand. South Africa and India have **.aero**, **.info**, **.mobi**, and **.org**, making them the only gTLDs hosted in either Africa or the South Asian region. South America hosts only **.com** and **.net**, with servers in two cities in Brazil. Taken together, these are the only Southern Hemisphere gTLD locations as of this writing, and no gTLD has any presence in parts of the world without external fiber-optic connectivity, although that may be changing.

Where gTLDs should be hosted, and with what scope, are somewhat open questions. Should these domains address resources anywhere, or should their scope be local? This question is really one for the *Internet Corporation for Assigned Names and Numbers* (ICANN), or for the gTLD sponsors or registries, and beyond the scope of this article. Verisign, the company that administers **.com** and **.net**, points out that database replication with the amount of changes in the **.com** zone is a significant problem over slow network links.

ccTLD Distribution

Questions about where ccTLDs, the top-level domains assigned to individual countries, ought to work seem more straightforward. Working effectively in their own countries seems like the top priority, with connectivity to the Internet core and to other regions with which people in the country communicate regularly being somewhat lower priorities. Just over two-thirds of ccTLDs are hosted in their own countries; refer to Figure 4 for the bigger countries, and the online appendices for the full list. Although the third of ccTLDs not hosted in their own countries include some marketed more for international use than global use—Cocos Island's **.cc**, Tonga's **.to**, Turkmenistan's **.tm**, and Tuvalu's **.tv**, among others—those are very much the exception.

Indonesia has local access to the root and to its ccTLD (**.id**). Pakistan has a root server, but no local access to its ccTLD (**.pk**). Let's compare what happens when someone in Indonesia does a lookup on an **.id** domain name with what happens when someone in Pakistan does a lookup of a name in the **.pk** domain.

In Indonesia, the query goes to a root DNS server at the Indonesian Internet Exchange in Jakarta, where it is answered with the locations of the **.id** servers, several of which are also in Indonesia. The query then goes to the local **.id** server and is answered locally, whereupon the user can start sending traffic to the host he or she was trying to connect to, which is presumably also local. The traffic need not leave Indonesia, and if all the parties involved are in Jakarta it need not leave town.

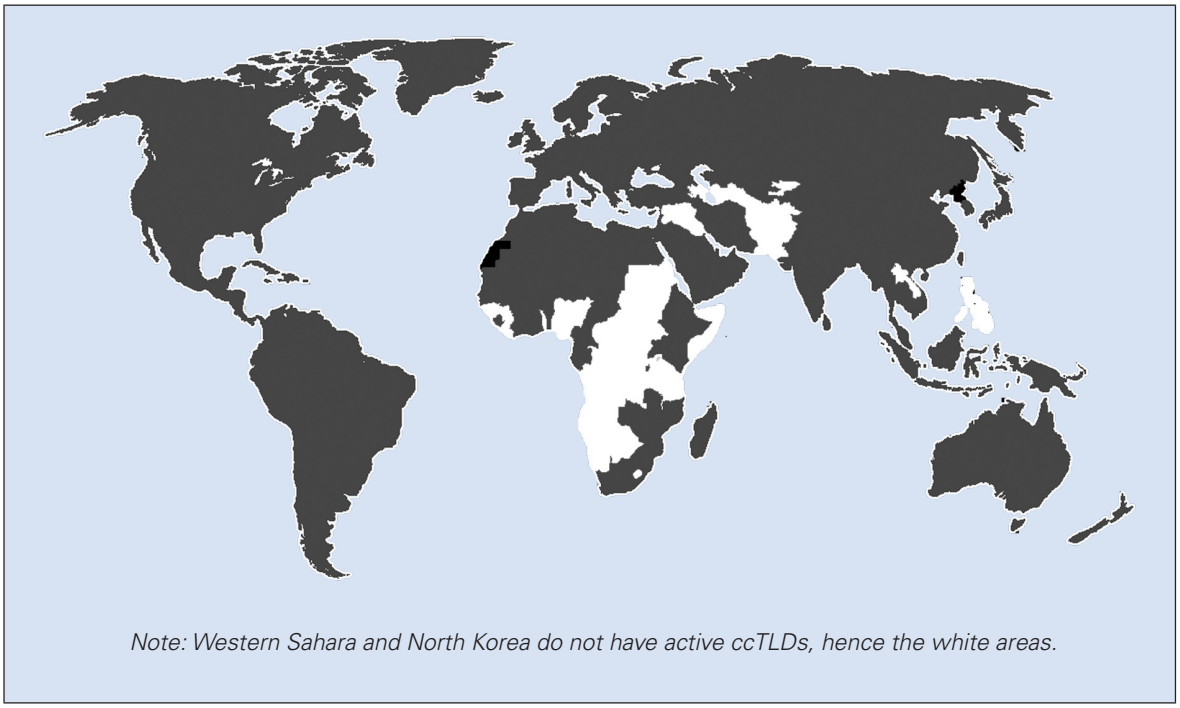
The Pakistani case is quite different. Until early 2006, there were no root servers in Pakistan, nor were there local servers for the **.pk** domain. There is now a root server in Karachi, but lacking servers for any TLDs it is of limited utility. DNS resolvers start out querying the local root server, but the response directs them to servers for the **.pk** domain, all located in the United States, at least 10 time zones away. They then send their lookup packets across the single fiber connection all the way to the United States and wait for the response. At best, this process is slow. If that fiber connection goes down, or if there is any other problem between Pakistan and these U.S. servers, local communications in Pakistan are crippled.

The situation with traditional gTLDs (**.com**, **.net**, and **.org**) in Indonesia and Pakistan is somewhat different. In Indonesia, local root servers provide addresses for the **.com**, **.net**, and **.org** servers. The **.com** and **.net** lookups can be handled in Singapore, 18 milliseconds away. Theoretically, **.org** lookups can be handled in Hong Kong, but *traceroutes* indicate **.org** queries being answered in California instead. Thus, in Indonesia, **.id** is hosted mostly locally, **.com** and **.net** are nearby, and **.org** is considerably farther away. In Pakistan, in contrast, **.pk** queries and **.org** queries are answered from the United States, more than 200 milliseconds away, while **.com** and **.net** are answered from Singapore, 80 milliseconds away. For Pakistani users of all TLDs, there are single points of failure, but **.com** and **.net** do appear to be somewhat better connected than **.pk**.

In Nairobi, Kenya, there are local copies of a root server and the local ccTLD (**.ke**). All external connectivity is by satellite, and most ISPs have only a single satellite link. Two Internet users in Nairobi wanting to communicate can do a lookup on the local root server to find the servers for **.ke** and can do a lookup on a local **.ke** server to find the servers for a subdomain of **.ke**. Assuming the subdomain being used is served locally, they can do a local lookup for a host within that subdomain and then send data across the local exchange point. Thus the two users in the same town can send data back and forth without having to send any data elsewhere.

According to Verisign, Nairobi will soon have servers for **.com** and **.net** as well. In contrast, to use the **.org** domain they can again obtain addresses of the **.org** server from their local root server, but the lookup of the **.org** domain must go over a satellite link to Europe in order to be answered by a server in London. If the satellite link is up, this process adds half a second of latency to the query. If the satellite link is down, whatever local resource they are trying to connect to is out of reach.

Figure 4: Countries that Host Their Own ccTLDs in Grey; Those that Do Not in Black



Note: Western Sahara and North Korea do not have active ccTLDs, hence the white areas.

There is also a concern about ccTLDs not served from the global core; if their region or upstream provider is cut off from the Internet outside their region, the rest of the world is unable to see that ccTLD. (See Table 2). This situation may or may not be of concern; if all Internet resources within that ccTLD become unreachable in the same outage, the DNS portion of the outage may have no additional effect. However, if there is anything in that ccTLD that is not in the ccTLD's region, or if people or systems outside prefer to get a DNS response for an unreachable IP address rather than no DNS response at all, it may be of concern. Indeed, having servers that are well-connected to "the Internet as a whole" is a recommendation of RFC 2182, though the RFC does not consider the case of large portions of the Internet not being well-connected to each other.

Table 2: TLDs Not Served in the "Internet Core" Region

TLD	Country	Location of DNS Servers
BB	Barbados	Barbados
BD	Bangladesh	Bangladesh
BH	Bahrain	Bahrain
CN	China	China
EC	Ecuador	Ecuador
GF	French Guiana	French Guiana and Guadeloupe
JM	Jamaica	Jamaica
KG	Kyrgyzstan	Kazakhstan
KW	Kuwait	Kuwait
MP	Northern Mariana Islands	Guam
MQ	Martinique	Guadeloupe and Martinique
PA	Panama	Brazil, Chile, Costa Rica, and Panama
PF	French Polynesia	French Polynesia
QA	Qatar	Qatar
SR	Suriname	Suriname
TJ	Tajikistan	Tajikistan
ZM	Zambia	South Africa and Zambia

Lack of Exchange Points and Local Peering

In the "Internet Mini-Cores" article[1], I noted that local hosting of critical infrastructure is moot if there is not either a local exchange point or a monopoly transit provider in the region. If data needed in a poorly connected region must leave the area and return to reach the user requesting it, the communication has double the latency, and possibly double the reliability problems, that it would have if it were hosted somewhere in the core. For the specific examples used in this article, I have mostly chosen areas that do have exchange points. I have not analyzed the underlying local infrastructure in all countries.

Methodology

The addresses of DNS servers for a TLD are available through several means: by looking at the root zone, by doing *digs* for the name servers, and by looking in the *Internet Assigned Numbers Authority* (IANA) *whois* data, among others. I did lookups against an anycast root server on my own network, because that seemed easiest to automate. My script then did a lookup for the address of each name server, stripped off the last octet, and produced a list of TLDs hosted in each /24 subnet.

There are 635 /24s containing name servers for TLDs; 142 of them host multiple TLDs; the rest host just one. I assumed that all DNS servers in a given /24 were likely to be in the same or nearby locations. This situation appears not to be the case for the UUNet name servers, and there are probably a few other exceptions that will show up as errors in my data.

I looked at a few automated geolocation systems to attempt to attach locations to the DNS servers, but none of them appeared to be producing accurate information. Instead, I guessed at the locations of the 600 subnets, using *traceroutes* from a variety of locations, paying attention to DNS, latency, and the results of whois queries for address space along the way. I also asked lots of questions of DNS operators and others and am particularly grateful to several anycast DNS operators, whose locations would not have all been found by my *traceroutes*. Some of my guesses are likely incorrect, and corrections are appreciated.

I may be missing some information about the UltraDNS TLD servers, because UltraDNS has locations it regards as confidential. This information about UltraDNS servers is from Afilias's **.net** application, *traceroutes* from a variety of locations, and UltraDNS.[7]

Locations of root servers are easier to find; they are listed at **http://www.root-servers.org**. Some supplemental information about **j.root-servers.net** was supplied by Verisign. If there are operational root servers not included on **www.root-servers.org** other than the J-roots, I did not count them.

The full lists of locations of all TLDs and TLD servers are in the appendices to this article, at:
http://www.pch.net/resources/papers/infrastructure-distribution/dns-distribution-appendices.pdf.

References

[1] Steve Gibbard, "Internet Mini-Cores: Local connectivity in the Internet's spur regions" (2005):
http://www.pch.net/resources/papers/Gibbard-mini-cores.pdf

[2] Root Server Technical Operations Association:
http://www.root-servers.org

[3] Joe Abley, "Hierarchical anycast for global service distribution," ISC Tech Notes (2003):
http://www.isc.org/index.pl?/pubs/tn/index.pl?tn=isc-tn-2003-1.html

[4] Bradley Huffaker, "Two days in the life of three DNS root servers" (2006):
http://www.caida.org/publications/presentations/2006/brad_wide0611_anycast_analysis

[5] Tim Richardson, "Ship's anchor cuts cable to Sri Lanka," *The Register,* August 24, 2004:
http://www.theregister.co.uk/2004/08/24/sri_lanka_anchor

[6] Tony Lee, Bradley Huffaker, Marina Fomenkov, and kc claffy, "On the problem of optimization of DNS root servers' placement" (2003):
http://www.caida.org/publications/papers/2003/dns-placement/

[7] Afilias, ".NET Application Form":
http://www.icann.org/tlds/net-rfp/applications/afilias.htm

STEVE GIBBARD is a Network Architect for the nonprofit organization, Packet Clearing House (**www.pch.net**), based in Berkeley, California. He runs an anycast DNS network that hosts the top-level domains for several countries and several of the "I" root anycast DNS servers, maintains PCH's network of route collectors and route servers at exchange points around the world, and researches the interconnection of Internet networks. In addition, Steve carries out network architecture and peering work as a consultant for several ISPs in the San Francisco Bay Area and elsewhere. Steve is a former Senior Network Engineer at Cable & Wireless, and has held network engineering positions at Digital Island and World Wide Net. E-mail: **scg@pch.net**

Writing Internet Drafts and RFCs Using XML

by Marshall T. Rose, Dover Beach Consulting, Inc. and Carl Malamud, Public Resource, Inc.

What is the work product of the *Internet Engineering Task Force* (IETF)? Some cynical observers might suggest "many fine lunches or dinners," but we argue that those niceties are merely the means to an end. The goal of the IETF is to provide open standards for the Internet community, and those standards are memorialized as written documents called *Request For Comments* (RFCs).

In general, two organizations control the publication of documents as RFCs:

- The *Internet Engineering Steering Group* (IESG) determines which documents are suitable for publication as RFCs—typically by chartering working groups, reviewing their progress (through reading the work-in-progress *Internet Drafts*)—and ultimately approving their documents for publication.
- The RFC Editor strives for "quality, clarity, and consistency of style and format," and has developed a particular editorial style. The latest RFC that documents this style, RFC 2223[1], is about a decade old. A somewhat more current version can be found in a text file maintained by the RFC Editor.[4]

For a more detailed discussion of the interaction between these two organizations, consult RFC 3932[2].

As an organization, the IETF excels at "eating its own dog food," including its work product: just as a protocol specification describes interactions on the wire but does not dictate the programming language used for implementation, so too, the IETF has not really cared which document preparation tools are used. The IESG worries about technical quality, and the RFC Editor worries about stylistic consistency (and, to be fair, technical quality as well). This policy works because of the careful choices made by the early Internet community, and in particular the RFC Editor, with respect to the "final form" footprint of the documents. (A discussion of these design decisions is far beyond the scope of this short article—for now, we note that it is hard to argue with success.)

An unfortunate side effect of this focus on stylistic consistency is that, for many years, the RFC Editor has had to recode documents for consistent formatting. Internally, the RFC Editor used *nroff*[5] for this purpose, and sophisticated authors wishing to minimize RFC Editor "downtime" tended to use the same *nroff* boilerplate. The *nroff* text-formatting program has many strengths, but it can also be fairly viewed as a textual "assembly language," with the result that authors spent a lot of time dealing with low-level formatting concerns.

In some limited cases, the high degree of formatting-specific expertise is warranted, but for the vast majority of documents, the high entry cost is not.

From Assembly Language to Markup

In early 1999 we were working at a startup company, and we needed a way to organize, search, and retrieve information from documents. We decided to use a markup language for this purpose. We also decided to use the RFC series as one of the testing grounds for the technology, because this series was one we were familiar with. Although today everyone knows what the *Extensible Markup Language* (XML) is, then there were only two widely known markup languages for authoring: SGML and HTML.

The "SG" in SGML is an abbreviation for *Standard Generalized* and not *Simple Generic*. SGML is used for the formatting of a great many books; further, it is used in large projects with long lifetimes. Although truly excellent from an "enumerate every possibility" standpoint, it has a very high cost of entry, making it difficult to use for anything other than specialized applications.

In contrast, the *Hypertext Markup Language* (HTML) embodies elegance of design, but (in the absence of *Cascading Style Sheets* [CSS]), is a presentation language, not unlike *nroff* in many respects. In other words, we needed something with the structural richness of SGML and the elegant simplicity of HTML. The newly invented XML seemed to meet the requirements.

This process led us to develop a language based on XML, which captured high-level RFC constructs (for example, authorship information) and largely ignored presentational concerns. The result is called the *2629 format*[3] (also known as the "xml2rfc format," named after the initial processor for this language).

The Advantages of Markup

To understand the advantages of this approach, let's look at one example: references. Like most archival series, the RFC Editor has a very rigorous, yet unstructured, syntax for citations. Although this consistency is good for readers of RFCs, achieving consistency of references using tools such as *nroff* was often the hardest part of creating a new document. With the 2629 format, the **`<reference>`** element contains a small number of subordinate elements that capture all the semantics of the reference. The XML processor takes this information and produces a properly formatted document.

Further, because this information is structured, it is possible to develop automated bibliographic databases for a wide range of data sources. In fact, using the XML "include" mechanism, a document author usually includes just a pointer to the reference, and lets the processor do all the complicated work.

A second advantage is that processors can produce different kinds of output. Some people prefer to view their documents in HTML rather than the canonical textual format. Julian Reschke has written a library of XSLT files that convert to various HTML formats (Strict, Transitional, XHTML, and so on). For example, references are hyperlinked in line, allowing for easy traversal of citations. Still others prefer the *Portable Document Format* (PDF) for printing. By using one of Julian's XSLT scripts and the truly excellent Prince[6] XML/CSS processor, the result is high-quality, printer-ready output.

However, the primary advantage is that the "high-level" approach allows the author to focus more on content and less on format: a processor can enforce the vast majority of the esoterica associated with the RFC Editorial style, including:

- Inserting required boilerplate (and in particular, the desired revision of the boilerplate)
- Checking for mandatory sections such as "Security Considerations" or "Normative References"
- Generating a specialized table of contents, etc.

To Infinity and Beyond

After publishing RFC 2629, an unexpected result occurred: people outside the IETF started using the 2629 format for their projects. Most credit for this side effect goes to the universality of the canonical textual format. However, some authors are using the 2629 format when writing books (they convert the 2629 format to SGML, which is sent to the publisher), business plans, and software documentation—and even to create a new series of non-IETF technical documents. The constituency here seems to revolve around having a simple yet structured way to author documents.

For the last few years, a large number of XML editing programs have been deployed, and many of these support the 2629 format. These editors offer two advantages: first, they provide a natural paradigm for editing nested content; and, second, sophisticated editors can be integrated into an automated work flow. (Having said that, the authors still use *Emacs* and *vi* for their XML editors.)

A good example of the use of XML editors is a "plug-in" for the *XMLMind* Editor[7]. This plug-in, written by Bill Fenner, provides a variety of services to the author, such as graphical editing of sections, templates for common constructs, and validation of references.

Over the last 10 years, the 2629 format has evolved in true IETF fashion, based on running code and a rough consensus. Originally created by the authors for our own convenience, we have been more than pleased to see this format used first by an informal community of developers and writers, and more recently by the IETF secretariat, tools team, and administrative entity and by the RFC Editor.

Today, many people use a common high-level markup language for writing RFCs. The next step in this natural evolution will be making the repository of XML-tagged RFCs available to those involved in document distribution, so that RFC repositories will be able to take advantage of the meta-data in the creation of search engine, alternative formats, and any other value-added constructs that would be of use to the community. (At present the RFC Editor prefers input in the 2629 format, but ultimately runs a processor that generates *nroff* for "tweaking"—in the near future, we hope that the xml2rfc textual output can be tuned to avoid this final step.)

To find out more, go to the xml2rfc Website[8] or visit the official directory of IETF authoring tools[9].

References

[1] Postel, J. and J. Reynolds, "Instructions to RFC Authors," RFC 2223, October 1997.

[2] Alvestrand, H., "The IESG and RFC Editor Documents: Procedures," BCP 92, RFC 3932, October 2004.

[3] Rose, M., "Writing I-Ds and RFCs using XML," RFC 2629, June 1999.

[4] Reynolds, J. and R. Braden, "Instructions to Request for Comments (RFC) Authors," August 2004.
ftp://ftp.rfc-editor.org/in-notes/rfc-editor/instructions2authors.txt

[5] Ossanna, J., "Nroff/Troff User's Manual," UNIX Programmer's Manual – Volume 2 (Bell Laboratories), 1979.
http://en.wikipedia.org/wiki/Nroff

[6] **http://princexml.com/**

[7] **http://www.xmlmind.com/xmleditor/**

[8] **http://xml.resource.org**

[9] **http://tools.ietf.org/inventory/author-tools.shtml**

CARL MALAMUD is the co-founder of Public Resource, a nonprofit public-benefit engineering firm. *Exploring the Internet* was published in 1992 as a book, but today would be called a blog. "Geek of the Week" was published in 1993 as an audio file available for download with FTP, but today would be called a podcast.
E-mail: **carl@media.org**

MARSHALL T. ROSE is Principal of Dover Beach Consulting, Inc. He has authored 9 books, 74 RFCs, and 4 patents. With respect to his work on the 2629 format, he claims "self defense." E-mail: **mrose@dbc.mtview.ca.us**

Fragments

ICANN Board Rejects .xxx Domain Application

On March 30th, 2007 the Board of the *Internet Corporation for Assigned Names and Numbers* (ICANN) voted to reject the **.xxx** *sponsored Top Level Domain* (sTLD) application from ICM Registry, Inc.

"This decision was the result of very careful scrutiny and consideration of all the arguments. That consideration has led a majority of the Board to believe that the proposal should be rejected," said Dr Vint Cerf, Chairman of ICANN. "I thank my fellow Board members and the community for their input," Dr Cerf said.

A copy of the resolution from the Board meeting is available at:

http://www.icann.org/minutes/minutes/resolutions-30mar07.htm

A transcript of the Board meeting is also available at:

http://icann.org/meetings/lisbon/transcript-board-30mar07.htm

ISOC Fellowship to the IETF

The *Internet Engineering Task Force* (IETF) is the world's premier Internet standards setting organization. It operates as a large, open international community of network designers, operators, vendor experts, researchers, and other interested technologists. While much of the IETF's work takes place over mailing lists, the in-person experience promotes a stronger understanding of the standardization process, encourages active involvement in IETF work, and facilitates personal networking with others that have similar technical interests.

Presently, there is limited participation at the IETF by technologists from developing countries. There are, however, many talented individuals in developing regions that have an interest in and follow IETF work and would benefit from the opportunities that attending an IETF meeting presents. As such, the main purposes of the Internet Society (ISOC)'s *IETF Fellowship Program* are to:

- Raise global awareness about the IETF and its work
- Foster greater understanding of and participation in the work of the IETF by technologists from the developing world
- Provide an opportunity for networking with individuals from around the world with similar technical interests
- Identify and foster potential future leaders from developing regions
- Demonstrate the Internet community's commitment to fostering greater global participation in Internet Forums such as the IETF

ISOC successfully piloted the IETF Fellowship program at the 66th IETF meeting in Montreal in June 2006. Two individuals from Africa participated in this first pilot. Three individuals from the Pacific and Latin America participated in a second pilot phase at the 67th IETF meeting in San Diego in November 2006. All found the experience highly beneficial. Based on the success of the pilots, ISOC decided to formalize the program beginning in 2007.

The ISOC Fellowship pays for the Fellow's IETF meeting registration and social event fees, a round-trip economy class airfare to the meeting, hotel accommodation, and a small stipend to offset incidental expenses.

The program provides fellowships for up to five individuals per IETF meeting. ISOC will be putting out a call for candidates, including through ISOC chapters, at least 3 months before an IETF meeting. A small selection committee comprised of individuals knowledgeable about the IETF will evaluate the applicants against selection criteria and make their fellowship recommendations.

Fellowship recipients will have an obligation to present or otherwise share their experiences at the IETF meeting they attend with their local community and to provide feedback on their experience to ISOC so that the program can be continuously improved. An *ISOC Fellowship Alumni Network* will be established to extend the fellows IETF experience and relationship-building opportunities after the meeting.

For further information on the specifics of the program and how to apply for an ISOC Fellowship see:

`http://www.isoc.org/educpillar/fellowship/application.shtml`

BGP: The Movie

Statistics on Internet resources have been animated to provide a high-level overview of the consumption and use of IPv4 addresses and AS numbers since 1983. The animated video also clearly shows the effect of *Classless Interdomain Routing* (CIDR) and *Regional Internet Registries* (RIR) allocation policies on consumption rates and routing. This animation was developed by *Asia Pacific Network Information Centre* (APNIC) staff members, Geoff Huston and George Michaelson. You can download the 58MB movie from:

`http://www.apnic.net/news/hot-topics/docs/bgp-movie.mpg`

Internet Governance Articles and References

APNIC is also maintaining a collection of articles and references on Internet governance to help the community understand the issues and stay abreast of developments. You can find these at:

`http://www.apnic.net/news/hot-topics/internet-gov/index.html`

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

June 2007 Volume 10, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

Part One of a two-part article on *Authentication, Authorization, and Accounting* (AAA) was published in our previous issue. This time Sean Convery presents Part Two—subtitled "Protocols, Applications, and the Future of AAA."

Interest in *IP Version 6* (IPv6) is growing in many parts of the Internet technical community; see, for example, the announcement from ARIN on page 39 of this issue. Transition to IPv6 is likely to be one of the greatest technical challenges in the history of the Internet. Several groups are developing parts of the overall solution by creating IPv6-capable versions of protocols such as the *Dynamic Host Configuration Protocol* (DHCP) or including support for IPv6 in the *Domain Name System* (DNS). Although not yet widely deployed, IP *Network Mobility* is expected to play an important part in the Internet of the future. For this reason the IETF is working on IP mobility with an eye toward IPv6. Our second article looks at the *Network Mobility* (NEMO) *Basic Support Protocol,* which is being developed by the NEMO working group in the IETF.

Depletion of IPv4 address space is not the only concern for network operators and developers these days. Questions about the long-term viability of today's routing protocols and the associated addressing systems center around a basic concern about how we can scale our networks to a size orders of magnitude larger than what we have today. A recently formed *Routing and Addressing Problem Directorate* (ROAP) is tasked to examine these problems in detail. Several ROAP-related sessions took place during the most recent IETF meeting, and Geoff Huston reports on these sessions and gives his analysis and commentary. Incidentally, Geoff was not present in person at this IETF meeting, but the facilities to follow an IETF meeting remotely are now of such a quality that he was able to participate from the other side of the world.

Protocol replacement or enhancement is also the theme in our final article. Dave Crocker asks the question "Is it time to replace SMTP?" Since this is an opinion piece, we invite your feedback or rebuttals.

New on our Website is a linked article index. Visit **cisco.com/ipj** and click on "Index Files" to explore this feature.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Network Authentication, Authorization, and Accounting
Part Two: Protocols, Applications, and the Future of AAA

by Sean Convery, Identity Engines

Network *Authentication, Authorization,* and *Accounting* has been used since before the days of the Internet as we know it today. Authentication asks the question, "Who or what are you?" Authorization asks, "What are you allowed to do?" And finally, accounting wants to know, "What did you do?" These fundamental security building blocks are being used in expanded ways today. The first part of this two-part series focused on the overall concepts of AAA, the elements involved in AAA communications, and high-level approaches to achieving specific AAA goals. It was published in IPJ Volume 10, No. 1[0]. This second part of the series discusses the protocols involved, specific applications of AAA, and considerations for the future of AAA.

AAA Protocols

Although AAA is often thought of as the exclusive province of the *Remote Authentication Dial-In User Service* (RADIUS) protocol, in reality a range of protocols is involved at various stages of the AAA conversation. This section introduces these AAA protocols, organized according to the parties involved in the communication. We divide AAA communications into the following categories: Client to *Policy Enforcement Point* (PEP), PEP to *Policy Decision Point* (PDP), Client to PDP, and PDP to *Policy Information Point* (PIP). For easy reference, the AAA flow diagram from Part One of this article is reproduced here. Please refer to Part One[0] for the explanatory text associated with the diagram.

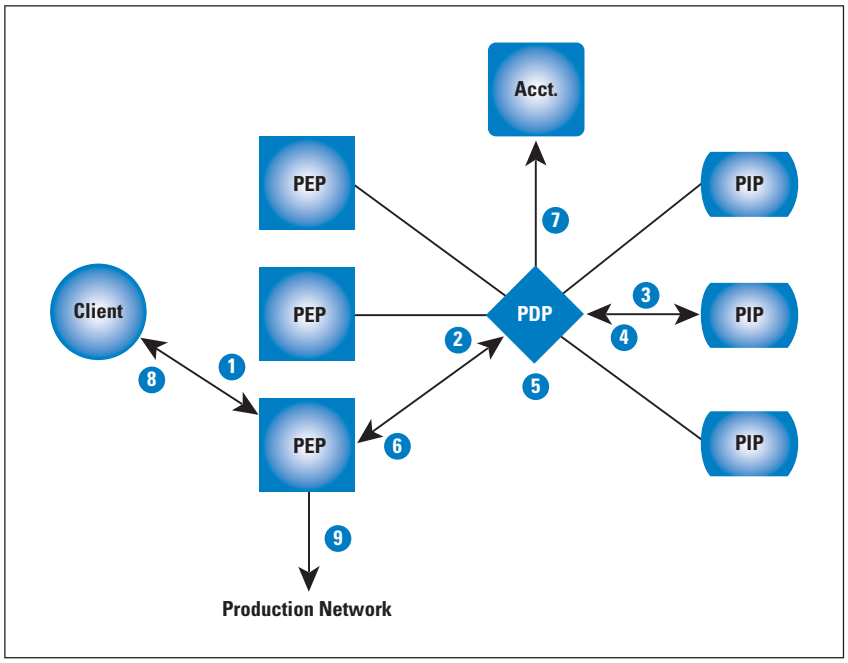


Figure 1: A Client Connects to a AAA-Protected Network (from Part One)

Client to PEP

AAA communications between the client and the PEP can travel at Layer 2 of the OSI model, or they can run at higher layers, relying on lower layers as essentially dumb transport. The most common protocols for client-to-PEP communication are the *Point-to-Point Protocol* (PPP)[1], *PPP over Ethernet* (PPPoE)[2], IEEE 802.1X[3], *IP Security* (IPsec), *Secure Sockets Layer* (SSL) VPN, and *Hypertext Transfer Protocol* (HTTP), each of which is discussed in this article.

PPP, the standard protocol for communicating across point-to-point links, includes an optional authentication step—the point at which the AAA element is introduced. During this authentication phase, protocols such as the *Challenge Handshake Authentication Protocol* (CHAP) can be used to identify the client to the PEP. (These protocols were discussed in the credential section of Part One of this article.) PPP is extensively used in dialup access but is otherwise not found in modern AAA. PPPoE, an adaptation of PPP to run over Ethernet, is used by many service providers rolling out broadband services.

PPPoE allows the broadband endpoint to authenticate itself to the service provider's network when making the initial connection. Because many broadband networks use shared Ethernet mediums, PPPoE allows *Internet Service Providers* (ISPs) to maintain the per-user accounting they were familiar with from dialup. The 802.1X protocol is an IEEE standard specifying a way to provide network access control at the port level for wired and wireless networks. The 802.1X standard specifies a way for the client to communicate with the PDP using the *Extensible Authentication Protocol* (EAP)[4], which is discussed in more detail later in this section. The 802.1X standard requires that the endpoint support 802.1X through a "supplicant" or client sign-on application. This application authenticates the client to the network through the PEP. (See the EAP section later in this article for an explanation showing how EAP and 802.1X can work together.)

For wireless networks, 802.1X has become the standard way of authenticating clients because it supports communicating unique key material to the client to secure its use of the wireless infrastructure. In wired Ethernet networks, 802.1X is rising in popularity as a way to authenticate clients as well. These applications are more fully described in the "AAA Applications" section, later in this article.

At a more generic level, the IPsec protocol has established a standard for securing IP communications, and this approach has become another common method of communicating from a client to a PEP (referred to as a *VPN Gateway* from an IPsec perspective). The initial authentication for IPsec communications uses the *Internet Key Exchange* (IKE) protocol. Version 1[5] of the IKE protocol had no built-in method for authenticating users with credentials such as passwords, so an extension to IKE called *XAUTH*[6] was proposed.

XAUTH never became an official standard (though it certainly was a de facto one) because the IETF IPsec working group created a second version of IKE[7] that used EAP as a transport for credentials such as passwords. Finally, in the areas of HTTP and VPN communications, the SSL and *Transport Layer Security* (TLS)[28] standards are two closely related protocols for securing, among other things, Web communications. SSL/TLS VPNs use these protocols to create a secure session from the client to the PEP (VPN Gateway). Client authentication with SSL and TLS can be done with client-side certificates, but more commonly they use passwords or *One-Time Passwords* (OTPs).

PEP to PDP

The three main protocols for communicating between a PEP and a PDP are *TACACS+*[9], *RADIUS,* and *Diameter*[10]. First, consider TACACS+: Developed by Cisco, TACACS+ is a proprietary protocol that is used primarily in communicating administrator authorizations for network devices. TACACS+ uses TCP port 49 and features payload encryption for the entire TACACS+ message. Though developed by Cisco, TACACS+ is supported by other companies as well, including Juniper.

Although TACACS+ excels at command-level authorizations and accounting for administrator control, another protocol has become far more common for client AAA: RADIUS. Thanks to nearly ubiquitous support for this protocol in network hardware, RADIUS is the primary protocol for communication between a PEP and a PDP in most environments. RADIUS uses the *User Datagram Protocol* (UDP) port 1812 for authentication and authorization and UDP port 1813 for accounting[8] (early deployments used ports 1645 and 1646, which are still used sometimes today). RADIUS supports numerous different attributes for communicating information back and forth from the PEP to the PDP, such as client MAC address, username, filter information for enforcement, and so on. It also supports an extensible framework for *Vendor-Specific Attributes* (VSAs), which allow extensions of the functions of RADIUS to support whatever elements a given PEP might need to best serve its role on the network. For example, a PEP manufacturer might support VSAs that allow the assignment of a user to a particular enforcement profile. RADIUS in its default implementation encrypts only the Password field of RADIUS messages, making the RADIUS protocol more prone to leaking information that could be used by an adversary. Both RADIUS and TACACS+ are secured by only a shared secret that is configured on both the PEP and the PDP.

Finally, consider the Diameter protocol. Diameter (the name is a play on words from RADIUS) is the next-generation, de jure standard for AAA. It supports stronger security through either IPsec or TLS and greater extensibility than RADIUS. It uses port 3868 for either TCP or the *Stream Control Transmission Protocol* (SCTP)[11]. The strongest use of Diameter to date is in the carrier space, where it provides AAA for call processing and *third-generation* (3G) mobile networks.

However, the corporate market has been fairly reluctant to embrace Diameter, and that reluctance has translated into a lack of support for Diameter in corporate network infrastructure equipment.

At this point in the discussion, it makes sense to compare RADIUS and Diameter. Although Diameter is an obvious alternative, RADIUS continues to be used in both new and existing deployments, so the IETF has a working group specifically formed to extend RADIUS in the future. The relationship between RADIUS and Diameter is a little like the relationship between IPv4 and IPv6. IPv6 had IPsec as a standard feature, IPv4 integrated IPsec as well, and today, by a large margin, most IPsec deployments are on IPv4 networks. The situation is similar with AAA. RADIUS certainly had limitations, but since Diameter entered the picture, RADIUS has been extended to address some of those shortcomings, particularly with both protocols using EAP as a transport. The result is that RADIUS today does what most people want. Therefore, given the significant added complexity of Diameter, many organizations have elected not to migrate to Diameter. Both RADIUS and Diameter will be around for many years to come.

Client to PDP

Although most of the protocols in this article handle communication from one component to the next component in the AAA chain (that is, client to PEP, PEP to PDP, etc.), there is one protocol that deals with communication from the client to the PDP directly: the *Extensible Authentication Protocol* (EAP). As mentioned earlier, EAP is a flexible mechanism for communicating almost any kind of credential over almost any lower-layer transport. Each technique for authenticating a client is referred to as an *EAP Method*. Originally conceived as an extension to PPP, EAP can now use many transports, including IKEv2 and 802.1X. Cisco's proprietary *Network Admission Control* (NAC) solution offers a deployment option that puts EAP inside UDP. When using 802.1X, for example, EAP uses LAN transport, referred to as *EAPoL* (EAP over LAN). This transport is only for the connection between the client and the PEP though. From the PEP to the PDP, EAP rides inside RADIUS[12, 13]. The actual conversation, however, takes place between the client and the PDP, with the PEP acting as a relay.

The major benefit of this approach is that the PEP does not need to understand the specifics of the EAP method selected—only the client and the PDP do. The EAP specification in the IETF specifies several different EAP methods, including *EAP Message Digest Algorithm 5* (EAP-MD5, very similar in security to CHAP), *EAP-OTP* (which supports an IETF-defined OTP solution[14]), and *EAP Generic Token Card* (EAP-GTC). Of the methods explicitly called out in the EAP standard, EAP-GTC is the only one in much use today in production networks. EAP-GTC allows the use of OTP token cards within an EAP context.

Beyond the methods defined in the EAP standard, EAP by its nature can be extended to support additional methods. EAP *Subscriber Identity Module* (EAP-SIM)[15] specifies a method for authentication using SIM elements in the *Global System for Mobile Communications* (GSM). EAP-SIM was developed by the *Third Generation Partnership Project* (3GPP) as a solution for these second-generation (GSM) mobile networks. EAP-AKA[16] is the 3GPP's EAP authentication technique for third-generation (*Universal Mobile Telecommunications Service* [UMTS] or *Code Division Multiple Access 2000* [CDMA2000]) mobile networks. Both EAP-SIM and EAP-AKA support authenticating a mobile phone to a Wi-Fi network without using passwords. The problem is that without some sort of user identity federation solution in place, SIM-based authentication can work only with the mobile provider's network that supplied the SIM card. EAP-TLS[17] specifies a technique for mutual certificate authentication. Although it is widely supported, EAP-TLS is not commonly deployed because of its requirement for client-side certificates.

Though none of the following EAP methods are standards, they—somewhat confusingly—represent the vast majority of EAP deployments. Each of them is referred to as a *Tunneled EAP Method* because it establishes one outer EAP method as a base secure channel and then runs another method (one that may be less secure) over that secure channel. *Protected EAP* (PEAP)[18], well supported in Microsoft's Windows operating system, has become a de facto standard for EAP methods. Most clients and PDPs support PEAP today. PEAP works by establishing a TLS session authenticated by the server certificate, and then an inner authentication method rides inside that TLS session. The inner method is almost always *Microsoft CHAP Version 2* (MS-CHAPv2), but other methods can be used as well. Another popular tunneled protocol is *EAP Tunneled TLS* (EAP-TTLS)[19]. This protocol is similar to PEAP except it supports a more arbitrary exchange of information inside the TLS tunnel. For example, one of the primary uses for EAP-TTLS is using the *Password Authentication Protocol* (PAP) as the inner authentication method, allowing an EAP-TTLS-capable PDP to authenticate clients against older password stores (such as those that support only PAP authentication).

Finally, in settings that use primarily Cisco equipment, a common tunneled protocol is *EAP Flexible Authentication via Secure Tunneling* (EAP-FAST)[20]. This protocol uses TLS to authenticate the PDP, and then a shared key is distributed to allow faster subsequent authentication. An inner EAP method such as MS-CHAPv2 can then be used to authenticate the client to the server. EAP-FAST is used extensively in Cisco products for wireless deployments.

PDP to PIP

The final set of AAA protocols we consider are the ones that govern the communication between the PDP and the PIP. The primary protocol of interest is the *Lightweight Directory Access Protocol* (LDAP)[21]. From a AAA context, LDAP allows a PDP to query a PIP (typically an X.500 directory[22]) for information about a client. This information is exposed through a series of group and attribute identifiers, which can include information about a client's home location, organizational role, job title (if referring to a user), and so on. LDAP includes several different authentication options[23]. This client information learned from the PIP enables the PDP to better make its policy decision. Also useful in the PDP-PIP communications context is the RADIUS protocol. Some large organizations or inter-organization federations use a hierarchy of RADIUS-speaking PDPs where one RADIUS PDP can act as a PIP for another RADIUS PDP further down the AAA hierarchy.

Finally, Microsoft *Active Directory* (AD) uses the LDAP protocol when acting as a PDP but also has its own extension, called *Netlogon,* for validating Microsoft credentials such as MS-CHAPv2. This means that integrating a PDP with Microsoft AD generally involves using LDAP to find information about the client and using Netlogon to validate the client's credential. Other options for PDP-to-PIP interaction—though less often used—include *Structured Query Language* (SQL) databases, *Network Information Service* (NIS), and Kerberos.

AAA Applications

This section surveys the different applications of AAA technology throughout networking. It is divided into three sections covering consumer, enterprise, and carrier applications, with a final section covering emerging applications of AAA technology.

Consumer-Managed Applications

Most consumer network deployments do not perform any advanced AAA beyond a shared key for authentication to a wireless network. In this example, the client is the consumer's host and the wireless access point acts as PEP, PDP, and PIP by validating that any client connecting to the access point presents the correct shared key.

Enterprise-Managed Applications

AAA has numerous enterprise applications, including remote access, wireless security, *Voice over IP* (VoIP), guest access, *Role-Based Access Control* (RBAC), and endpoint posture validation (also known as NAC). This section discusses each of these applications. Remote-access security is the original enterprise AAA application. In the remote-access scenario, remote users connect over a dialup connection or a VPN and authenticate themselves (and optionally their hosts) to the organization's network.

The client's credential is almost always a password, expressed in one of the forms discussed in the credential section of Part One of this article. The main purpose of AAA in the remote-access case is to validate that the client is a valid user of the organization's network.

Wireless security is similar in some respects to remote-access security. The goals of AAA in wireless security are twofold: first it must validate that the wireless client is an authorized user, and second, it must provide the client with a session key for cryptographic protection of the client's traffic. Given these goals, 802.1X using EAP are the ideal protocols to use because they support both client authentication and dynamic keying. Older wireless security approaches relied on an open wireless network and a VPN Gateway separating that network from the rest of the organization's network. In that example, the wireless-security approach mimics the remote-access application just discussed. Other types of networking require different applications of AAA. For example, VoIP deployments have authentication requirements as well. The *Session Initiation Protocol* (SIP)[24] is used extensively for, well, session initiation in VoIP networks (for example, authenticating the calling parties prior to initiating a new call). Authentication can be handled natively within SIP using HTTP digest authentication, or the same request can be sent to a PDP using RADIUS. AAA for VoIP allows handsets to authenticate themselves to the network and gain access to call-processing services.

Another, very popular application of AAA is guest-access management for networks. This application has grown quickly with the recent growth of wireless networks. Guest access is a method by which guests can be granted temporary access to a network with a full audit trail[27]. Guest access generally involves creating a distinct PIP, which houses short-term user accounts, and a technique for creating and, after a configurable period of time, automatically deactivating those user accounts. The PIP is often co-resident with the PDP and allows this temporary access without having to provision these users into the organization's more permanent directory. The guest can communicate with the PEP using any of the client-PEP protocols discussed earlier, though HTTP is the most common. The PEP is told by the PDP that the client (because it is a guest) should have restricted access—typically access only to the Internet at large and not any communication with an organization's internal network.

Also growing in popularity as a AAA application is RBAC, an application of AAA that allows customization of the network session based on the role of the client. In fact, guest access is a simple form of RBAC whereby two classes of clients are created: guest and permanent. However, RBAC can be extended to include more levels of delineation, including guest, contractor, and specific classes of permanent users such as sales, human resources, and engineering.

This classification can be done with all forms of AAA-enabled network infrastructure, including wired, wireless, and remote access. Current scalability limitations of VLAN technology and *Access Control Lists* (ACLs) make creating large quantities of roles difficult, but a significant business benefit in audit and regulatory compliance can be realized with usually fewer than five roles.

To implement RBAC, most organizations choose a mix of 802.1X and HTTP authentication for wired and wireless access, combined with VPN technology for remote access. This approach is the most common one to RBAC, though others are used.

Finally, another important AAA application is *Endpoint Posture Validation,* also referred to as *Network Access Control* (NAC). Unfortunately NAC is an inappropriate name because of its almost complete overlap with the more general AAA term—leading to a fair amount of confusion in the market. Endpoint posture validation refers to many different parameters in the industry as it is an emerging technology. These parameters range from very narrow device-centric posture checking to a more identity-centric approach for secure mobile computing. Because this entire article is concerned with the latter, we will consider NAC in its narrow context of endpoint posture checking. With this label, NAC simply acts as another PIP for the PDP to use.

This time, though, instead of checking the client's credential, NAC checks the client's software configuration. This checking generally focuses on security-sensitive configuration details of the endpoint security software and the operating system itself, such as the revision, configuration, and current operating status. This client configuration data is gathered by a host agent on the client and then sent to the PDP or PIP for evaluation. The host agent is either permanent on the client or downloaded dynamically to acquire the information. Some NAC applications rely exclusively on external scanning of the client, although this scanning generally yields far less granular information than an agent would.

The challenge with NAC today is deploying a system built on standards. The IETF and the *Trusted Computing Group* (TCG) are both pursuing standards in this space. Meanwhile Cisco, Microsoft, and a host of smaller companies have offerings not currently based on any standard. Recent announcements from the TCG and Microsoft are changing this. The TCG recently standardized the as-implemented NAC protocol used by Microsoft's NAC approach. Though there is much more work to do, this should allow the beginnings of standards-based interoperability in NAC solutions since a core protocol in Microsoft's NAC is now a standard from the TCG. There is a great base in standards at a low enough layer in all the NAC approaches though, as the emerging standards use the protocols discussed in this article including 802.1X, IPsec, RADIUS, and LDAP.

Carrier-Managed Applications

Some carrier-managed AAA applications are similar to those for the enterprise and others are different. The common distinctions for almost all carrier applications are their large scale and their emphasis on accounting. Carrier applications include dialup, DSL or cable PPPoE, mobile or 3G, wireless hotspot, and metro wireless. Dialup is similar to the remote-access application in the enterprise section, but on a massive scale. *Network Access Servers* (NASs) for a large ISP are geographically dispersed, as are the PDP and PIP systems that support them. Clients communicate with the PEP (NAS) with PPP using one of the password credential techniques discussed in Part One of this article, and the PEP communicates with the PDP using RADIUS or Diameter.

Now consider DSL or cable PPPoE. Though PPPoE-based broadband access seems to be on the decline, many ISPs are still using PPPoE for the enhanced audit trail it provides compared with an unauthenticated connection. In the realm of mobile telephone networks, service providers are increasingly providing data services in mobile phones, and these services require AAA for security and billing. Such data services come in several varieties on both the second- and third-generation mobile networks. Additionally, smartphones are increasingly supporting 802.11-based wireless access as well, creating a complex relationship between the smartphone, mobile voice network, mobile data network, 802.11 data network, and VoIP-based voice services. Previously discussed standards such as EAP-SIM and EAP-AKA are trying to bridge some of these worlds, but there is much work to be done. Ideally, any smartphone should take advantage of the network with the fastest and richest set of services, and callers trying to reach a smartphone user as well as the user himself, should be shielded from this discovery and association process. Business motivators and detractors within the carrier space may affect this convergence.

The next carrier-managed AAA application to discuss is the *wireless hotspot*. Hotspots work much like dialup providers in that regular users get a password-based credential that lets them authenticate to the hotspot. In this context, the 802.1X protocol is less commonly used because the required client software is not yet ubiquitous in the client install base. More common is Web-based authentication much like that used to access broadband in a hotel. A critical security consideration for a hotspot operator is the ability to ensure that a given client is not connected to two hotspots at the same time—a situation that would indicate an account was shared between two or more users. This stipulation places an increasing burden on the accounting aspect of AAA, as with any carrier-based AAA application.

Finally, the last AAA application we examine is the metropolitan wireless network, known as "metro wireless." In metro wireless, an 802.11 network is deployed throughout a metropolitan area, and access is provided free of charge or for a fee. I live in Mountain View, California, which is home to Google's headquarters, and is where Google has installed its free, citywide metro wireless network.

Although the service is free, AAA is still required: to sign on to the wireless network, you must authenticate to Google using an ID. This step, much like signing on to a wireless hotspot, allows Google to trace network use to an individual (if necessary) and switch to a fee-based model later on if desired. HTTP authentication is most common in metro wireless environments, and, because of the on/off nature of access, little sophistication in policy decision is required other than validating the client's credential.

Emerging Applications

Several interesting applications of network AAA are emerging. The first is in building just-in-time networks, such as when establishing an on-scene emergency operations center after a disaster. In this situation, emergency workers often need to communicate in a protected environment, and the press that covers the disaster needs network access to send in its reports. The AAA application required here is a cross between wireless security, guest access, and RBAC.

Another emerging application is what we call "granular RBAC." As opposed to RBAC, which associates users into coarse-grained classes of users, granular RBAC knows much more about the users and makes a more sophisticated access decision.

One example of the use of granular RBAC is for classroom control in higher education. Increasingly, classrooms are wireless-enabled as a convenience feature for faculty and students. However, during exam time it is often useful to disable this access to the students taking an exam. Without a granular understanding of which clients are connecting to the network, this setup is very difficult to achieve without physically disabling large portions of the wireless network during exam time. By using AAA, a school could put class schedules inside an LDAP store along with the rest of the students' information. Professors could also register exam times by time and location. AAA could then prevent students from getting on the network inside the classroom during their exam period, while still letting them connect to the network when inside their dorm room.

Finally, the last application we consider is what I call "punitive access restrictions." As networks become more and more an integral part of our lives, it is natural to want as fast a network connection as we can find, creating the situation where denying access to the network based on past behavior (network related or not) can be used as a punitive action. Today, your driver's license can be revoked based on your behavior while on the road. Punitive access restrictions on the network could mirror the same technique (for example, punishing people who propagate a virus by restricting their network access for a time) or could be used even if the infraction is not related to the network. Imagine a university that has trouble getting students to return overdue library books. Fines are one way to get the books back quickly, but if the student's parents are paying the bill, this consequence may not be as effective as the university desires.

However, imagine if the student's account record (in the PIP) had a directory attribute containing a count of the student's overdue library books. The network could then use RBAC or *Quality-of-Service* (QoS) techniques to provide degraded access to the student until the books were returned.

The Future of AAA

AAA as a concept has remained relatively unchanged since its inception. However, as this article has demonstrated, the techniques and applications of AAA continue to evolve. This section discusses some of the ways AAA may change more fundamentally in the future.

Security and Identity Convergence

Today the security and identity services provided by physical building access, network access, and application access are completely distinct. Security can be improved by communicating among these layers. Imagine a user executing a $10 million purchase order in a financial application. The chance of fraud would be reduced if the application could know that the user was coming from an authorized client with an up-to-date antivirus configuration. The chance of fraud could be further reduced by checking that the same user had accessed the badge access system of the building that day, and that the point of badge-access entry was consistent with the location where the application request originated. Within computer security, the notion of *defense-in-depth* has been around for a long time and is considered a best practice. Security and identity convergence adds new layers to this defense, and can potentially make all the layers more intelligent in their interaction.

User-Centric AAA

In the Web application world, the notion of user-centric identity is gaining ground. Kim Cameron's "Laws of Identity"[25] makes a compelling case that identity information housed in silos to be used by one organization is problematic. Several circles in the Web and e-commerce communities are beginning to look at identity differently. One change, consistent with the notion of user-centric identity, is that users should own their own identity information and should control how that information is used. The simplest example I can offer is shopping preferences at an online store. Most online stores make suggestions to you based on prior purchases. This data is owned by the online store, though, and not you, the consumer. If you wanted to take your purchasing profile from Amazon.com and transfer it to Barnes and Noble, it would not work. With user-centric identity, this kind of process is possible.

Another example is asserting a user's age. Depending on what you are trying to do on the Internet, you may need to validate that you are above a certain age. To do that, you are often asked to enter your date of birth, but that is more information than the site really needs.

If you could assert, with an identity you control but that is validated by a trusted party, that you are over the required age, it would not be necessary to disclose your date of birth (a process that is sometimes used as an authentication factor when you call places such as your credit card company).

The idea here is that you control your own information and limit what you need to share with others. This is very beneficial for privacy. One user-centric identity approach is included in Windows Vista through an application called "Card Space." Other approaches include OpenID and the Higgins Project. All of these approaches are somewhat consumer-focused, but if they take hold, it seems natural that there will be pressure for similar identity approaches in the enterprise and carrier space.

Federation

One of the natural evolutions of AAA infrastructure is to start federating access between multiple organizations. Imagine if visiting professors at another university's campus could access the network as guests using their password from their home location? Federation promises to make this possible, but the most challenging hurdles are political and logistical rather than technological. Protocols such as the *Security Assertion Markup Language* (SAML)[26] combined with RADIUS and LDAP can overcome this hurdle. The challenge is how to set up the trust relationships between the organizations to make it work. Eduroam based on RADIUS is an early effort delivering federation in Europe today.

Summary

This article, with its companion piece, has explored all aspects of AAA. Part One described the overall approach of AAA, how it works, and the elements that provide authentication, authorization, and accounting. Part Two has explored all the protocols used in the communication between the various AAA elements, the applications of AAA, and some thoughts about the future of AAA. AAA is a giant topic, and each of these sections, protocol descriptions, and applications could be expanded into a paper all by itself. The information in this article, combined with the references provided, should be a good starting point for your own examination of the specific aspects of AAA that are of interest to you.

References

[0] Convery, S., "Network Authentication, Authorization, and Accounting —Part One: Concepts, Elements, and Approaches," *The Internet Protocol Journal,* Volume 10, No. 1, March 2007.

[1] Simpson, W., "The Point-to-Point Protocol (PPP)," RFC 1661, July 1994.

[2] Mamakos, L., "A Method for Transmitting PPP Over Ethernet (PPPoE)," RFC 2516, February 1999.

[3] Jeffree et al., "Port-Based Network Access Control," IEEE Std 802.1X-2004, November 2004.

[4] Aboba et al., "Extensible Authentication Protocol," RFC 3748, June 2004.

[5] Harkins et al., "The Internet Key Exchange (IKE)," RFC 2409, November 1998.

[6] Beaulieu et al., "Extended Authentication within IKE (XAUTH)," Internet Draft, Work in Progress, October 2001. **draft-beaulieu-ike-xauth-02.txt**

[7] Kaufman C., ed., "Internet Key Exchange (IKEv2) Protocol," RFC 4306, December 2005.

[8] Rigney C., "RADIUS Accounting," RFC 2866, June 2000.

[9] Carrel et al., "The TACACS+ Protocol Version 1.78," Internet Draft, Work in Progress, January 1997. **draft-grant-tacacs-02.txt**

[10] Calhoun et al., "Diameter Base Protocol," RFC 3588, September 2003.

[11] Stewart et al., "Stream Control Transmission Protocol," RFC 2960, October 2000.

[12] Aboba et al., "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)," RFC 3579, September 2003.

[13] Congdon et al., "IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines," RFC 3580, September 2003.

[14] Haller et al., "A One-Time Password System," RFC 2289, February 1998.

[15] Haverinen et al., "Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)," RFC 4186, January 2006.

[16] Arkko et al., "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)," RFC 4187, January 2006.

[17] Aboba et al., "PPP EAP TLS Authentication Protocol," RFC 2716, October 1999.

[18] Palekar et al., "Protected EAP Protocol (PEAP) Version 2," Internet Draft, Work in Progress, October 2004. **draft-josefsson-pppext-eap-tls-eap-10.txt**

[19] Funk et al., "EAP Tunneled TLS Authentication Protocol Version 1 (EAP-TTLSv1)," Internet Draft, Work in Progress, March 2006. **draft-funk-eap-ttls-v1-01.txt**

[20] Cam-Winget et al., "The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST)," Internet Draft, Work in Progress, January 2007. **draft-cam-winget-eap-fast-06.txt**

[21] Zeilenga K., "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map," RFC 4510, June 2006.

[22] Zeilenga K., "Lightweight Directory Access Protocol (LDAP): Directory Information Models," RFC 4512, June 2006.

[23] Harrison R., "Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms," RFC 4513, June 2006.

[24] Rosenberg et al., "SIP: Session Initiation Protocol," RFC 3261, June 2002.

[25] Cameron, "The Laws of Identity," May 2005.

[26] OASIS, "Security Assertion Markup Language 2.0," March 2005.

[27] Dory Leifer, "Visitor Networks," *The Internet Protocol Journal,* Volume 5, No. 3, September 2002.

[28] William Stallings, "SSL: Foundation for Web Security," *The Internet Protocol Journal,* Volume 1, No. 1, June 1998.

SEAN CONVERY is CTO at Identity Engines, a venture-backed startup developing innovative identity management solutions for enterprise networks. Prior to Identity Engines, Sean (CCIE® no. 4232) worked for seven years at Cisco Systems, most recently in the office of the security CTO. Sean is best known as the principal architect of the SAFE Blueprint from Cisco and the author of *Network Security Architectures* (Cisco Press, 2004). Sean has presented to or consulted with thousands of enterprise customers around the world on designing secure networks. Before Cisco, Sean held various positions in IT and security consulting during his 14 years in networking. E-mail: **sconvery@idengines.com**

IPv6 Network Mobility

by Carlos J. Bernardos, Ignacio Soto, and María Calderón, Universidad Carlos III de Madrid

The *Internet Protocol* (IP) is currently accelerating the integration of voice and data communications. The Mobile IP protocol enables host mobility support, but several scenarios exist today, such as the provision of Internet access from mobile platforms (for example, planes, trains, cars, etc.), making it necessary to also support the mobility of complete networks. In response to this demand, the *Internet Engineering Task Force* (IETF) has developed the *Network Mobility* (NEMO) *Basic Support Protocol*[1], enabling IPv6 network mobility.

This article explains the Network Mobility Basic Support Protocol, by first providing a general overview and then examining the details.

Why Network Mobility?

Accelerated by the success of cellular technologies, mobility has changed the way people communicate. As Internet access becomes more and more ubiquitous, demands for mobility are not restricted to single terminals anymore. It is also needed to support the movement of a complete network that changes its point of attachment to the fixed infrastructure, maintaining the sessions of every device of the network: what is known as *network mobility* in IP networks. In this scenario, the mobile network has at least a (mobile) router that connects to the fixed infrastructure, and the devices of the mobile network connect to the exterior through this mobile router.

Support of the roaming of networks that move as a whole is required in order to enable the transparent provision of Internet access in mobile platforms, such as the following:

- *Public transportation systems:* These systems would let passengers in trains, planes, ships, etc. access the Internet from terminals onboard (for example, laptops, cellular phones, *Personal Digital Assistants* [PDAs], and so on) through a mobile router located at the transport vehicle that connects to the fixed infrastructure.
- *Personal networks:* Electronic devices carried by people, such as PDAs, photo cameras, etc. would connect through a cellular phone acting as the mobile router of the personal network.
- *Vehicular scenarios:* Future cars will benefit from having Internet connectivity, not only to enhance safety (for example, by using sensors that could control multiple aspects of the vehicle operation, interacting with the environment and communicating with the Internet), but also to provide personal communication, entertainment, and Internet-based services to passengers.

However, IP networks were not designed for mobile environments. In both IPv4[2] and IPv6[3, 4], IP addresses play two different roles. On the one hand, they are *locators* that specify, based on a routing system, how to reach the node that is using that address. The routing system keeps information about how to reach different sets of addresses that have a common network prefix. This address aggregation in the routing system satisfies scalability requirements. On the other hand, IP addresses are also part of the *endpoint identifiers* of a communication, and upper layers use the identifiers of the peers of a communication to identify them. For example, the *Transmission Control Protocol* (TCP), which is used to support most of the Internet applications, uses the IP address as part of the TCP connection identifier.

This dual role played by IP addresses imposes some restrictions on mobility, because when a terminal moves from one network (IP subnet) to another, we would like to *maintain* the IP address of the node that moves (associated to one of its network interfaces) in order not to change the identifier that upper layers are using in their ongoing sessions. However, we also would like to *change* the IP address to make it topologically correct in the new location of the terminal, allowing in this way the routing system to reach the terminal.

Protocols such as the *Dynamic Host Configuration Protocol* (DHCP)[5, 6] facilitated the portability of terminals by enabling the dynamic acquisition of IP configuration information without involving manual intervention. However, this automation is not enough to achieve real and transparent mobility because it requires the restarting of ongoing transport sessions after the point of attachment changes. The IETF has studied the problem of terminal mobility in IP networks for a long time, and IP-layer solutions exist for both IPv4 (Mobile IPv4[7, 8]) and IPv6 (Mobile IPv6[9]) that enable the movement of terminals without stopping their ongoing sessions.

If we focus on IPv6[3] networks, Mobile IPv6 does not support, as it is now defined, the movement of complete networks. One way of achieving the transparent mobility of all the nodes of a network moving together (for example, in a plane) could be enabling host mobility support in all of them, so they independently manage their mobility. However, this approach has the following drawbacks:

- Host mobility support (for example Mobile IP[7, 8, 9]) is required in *all* the nodes of the network. This support might not be possible, for example, because of the limited capacities of the nodes (such as in sensors or embedded devices) or because it is not possible to update the software in some older devices. By having a single entity (the mobile router) that manages the mobility of the complete network, nodes of the network do not require any special mobility software to benefit from the transparent mobility support provided by the (mobile) router.

- The signaling exchanged because of the roaming of the network is limited to a single node sending only one message (avoiding "storms" of signaling messages every time the network moves).
- Nodes of the network must be able to attach to the access technology available to connect to the Internet. This requirement might mean that all the nodes of the network should have *Universal Mobile Telecommunications Service* (UMTS) or WiMAX interfaces, for example. On the other hand, by putting this requirement on a single node (the mobile router), nodes of the network can gain access to the Internet through the mobile router, using cheaper and widely available access technologies (for example, *wireless LAN* [WLAN] or Bluetooth).

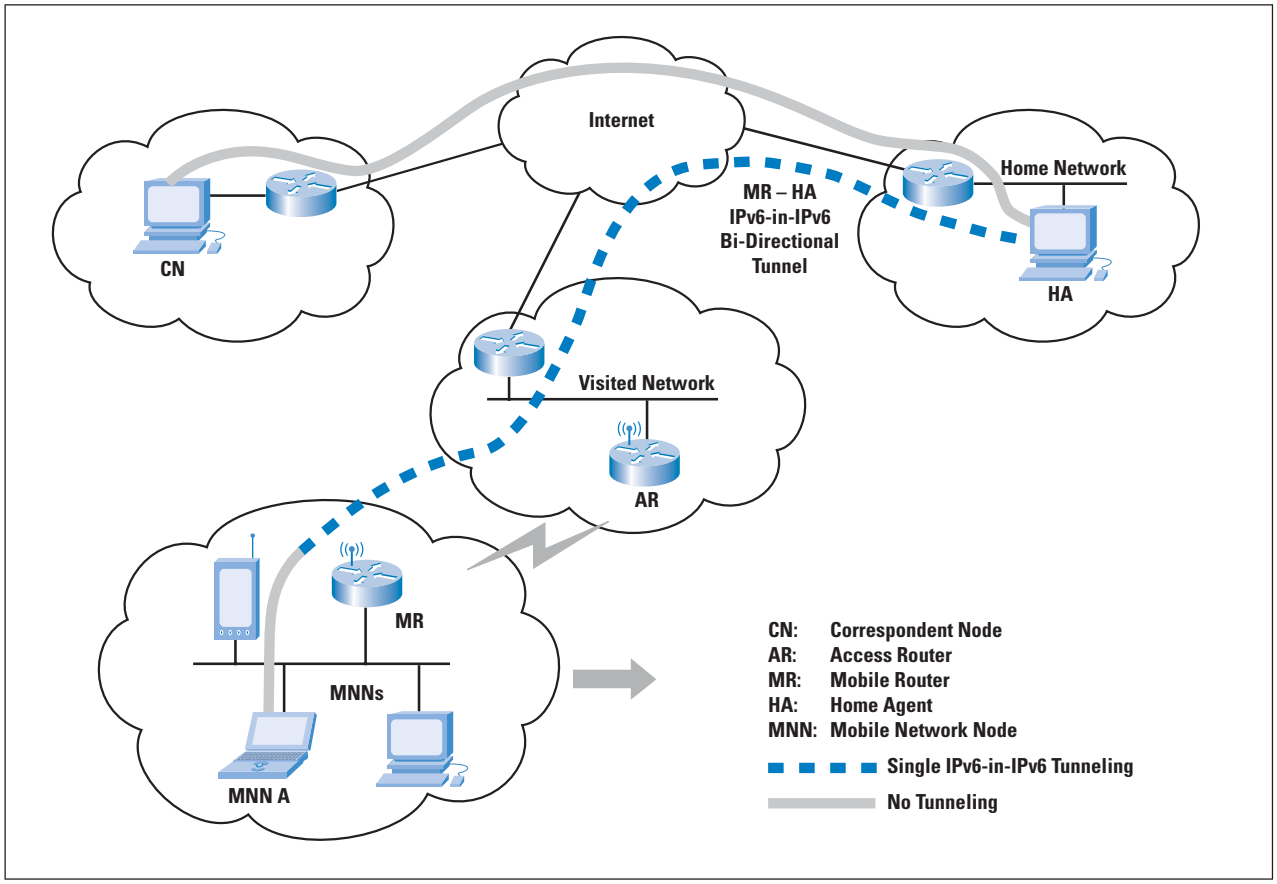
Because of these problems, the IETF *NEMO Working Group* was created to standardize a solution enabling network mobility at the IPv6 layer. The current solution, called the Network Mobility Basic Support Protocol, is defined in RFC 3963[1].

Operation of the NEMO Basic Support Protocol

A mobile network (known also as a "network that moves," or *NEMO*) is defined as a network whose attachment point to the Internet varies with time. Figure 1 depicts an example of a network-mobility scenario. The router within the NEMO that connects to the Internet is called the *Mobile Router* (MR). It is assumed that the NEMO is assigned to a particular network, known as its *Home Network,* where it resides when it is not moving. Because the NEMO is part of the home network, the mobile network has configured addresses belonging to one or more address blocks assigned to the home network: the *Mobile Network Prefixes* (MNPs). These addresses remain assigned to the NEMO when it is away from home. Of course, these addresses have topological meaning only when the NEMO is at home. When the NEMO is away from home, packets addressed to the nodes of the NEMO, known as *Mobile Network Nodes* (MNNs), are still routed to the home network. Additionally, when the NEMO is away from home, the mobile router acquires an address from the visited network, called the *Care-of Address* (CoA), where the routing architecture can deliver packets without additional mechanisms.

When any node located at the Internet, known as a *Correspondent Node* (CN), exchanges IP datagrams with a *Mobile Network Node* (MNN; A in Figure 1), the following operations are involved in the communication:

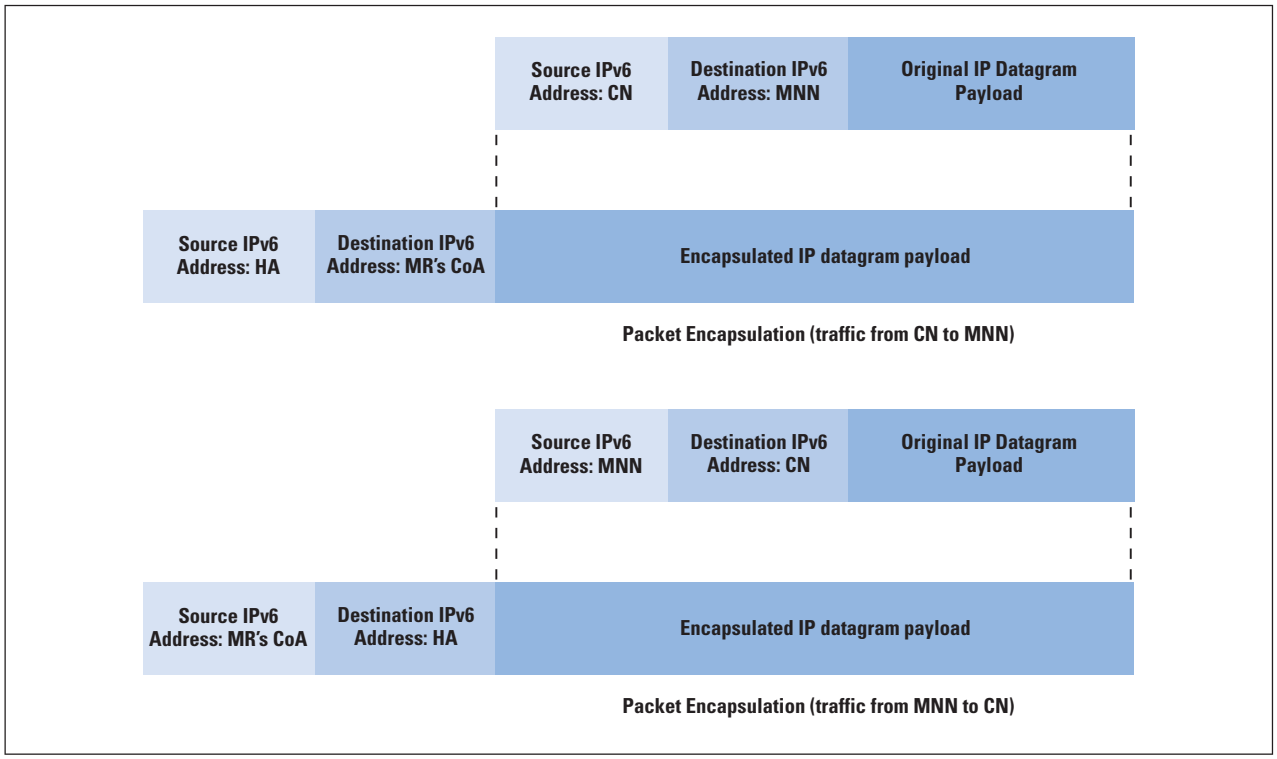
Figure 1: Example of NEMO Basic Support Protocol Operation



1. The correspondent node transmits an IP datagram destined for MNN A. This datagram carries as its destination address the IPv6 address of MNN A, which belongs to the MNP of the NEMO.
2. This IP datagram is routed to the home network of the NEMO, where it is encapsulated inside a new IP datagram by a special node located on the home network of the NEMO, called the *Home Agent* (HA). The new datagram is sent to the CoA of the mobile router, with the IP address of the home agent as source address. This encapsulation (as shown in Figure 2) preserves mobility transparency (that is, neither MNN A nor the correspondent node are aware of the mobility of the NEMO) while maintaining the established Internet connections of the MNN.
3. The mobile router receives the encapsulated IP datagram, removes the outer IPv6 header, and delivers the original datagram to MNN A.

4. In the opposite direction, the operation is analogous. The mobile router encapsulates the IP datagrams sent by MNN A toward its home agent, which then forwards the original datagram toward its destination (that is, the correspondent node). This encapsulation is required to avoid problems with ingress filtering, because many routers implement security policies that do not allow the forwarding of packets that have a source address that appears topologically incorrect.

Figure 2: Overview of NEMO Basic Support Protocol Encapsulation

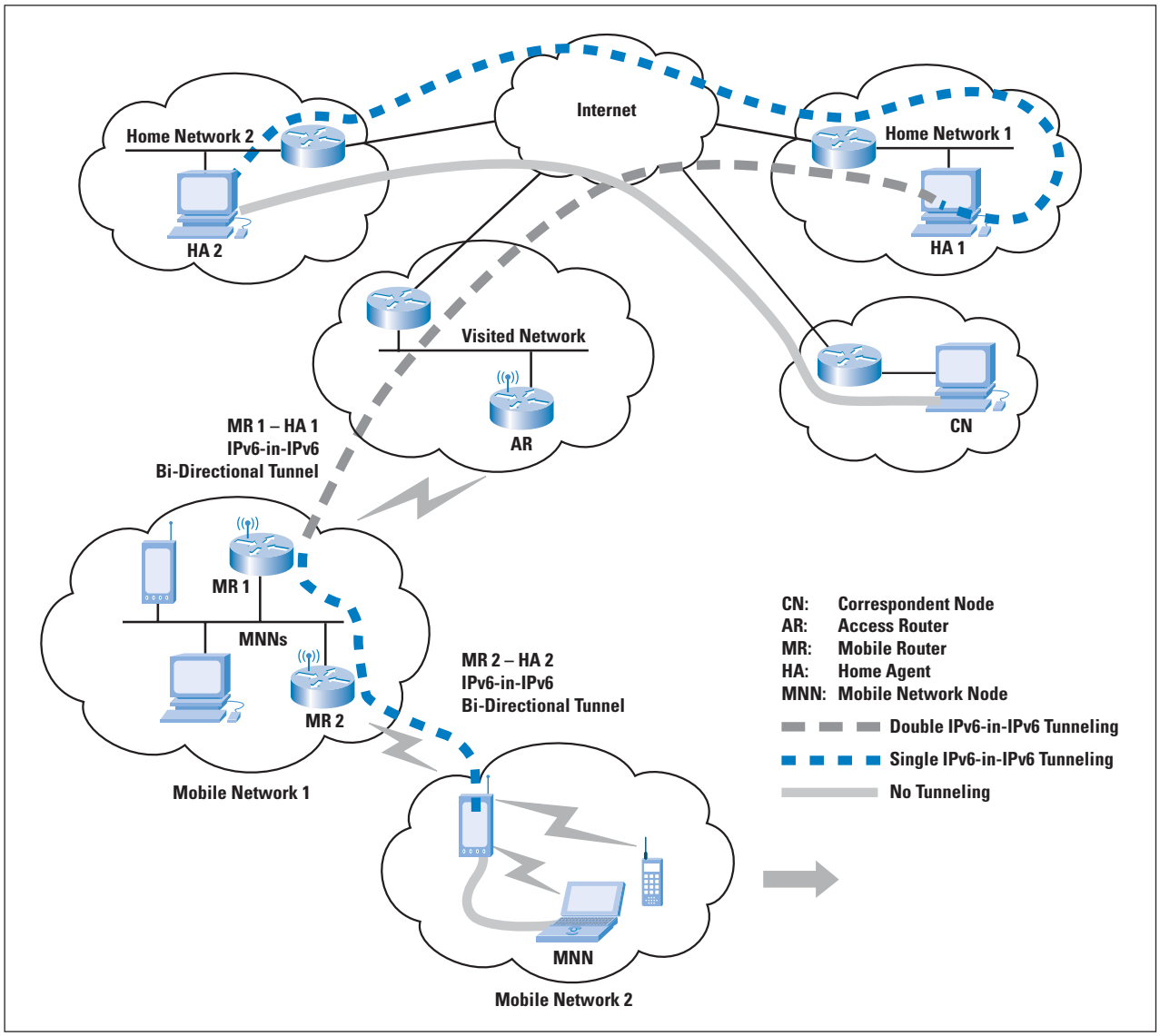


Following are different types of MNNs:

- *Local Fixed Node* (LFN): This node has no mobility-specific software and therefore cannot change its point of attachment while maintaining ongoing sessions. Its IPv6 address is taken from a MNP of the NEMO to which it is attached.
- *Local Mobile Node* (LMN): This node implements the Mobile IPv6 protocol; its home network is located in the mobile network. Its *home address* (HoA) is taken from an MNP.
- *Visiting Mobile Node* (VMN): This node implements the Mobile IP protocol (and therefore, it can change its point of attachment while maintaining ongoing sessions), has its home network outside the mobile network, and it is visiting the mobile network. A VMN that is temporarily attached to a mobile subnet (used as a foreign link) obtains an address on that subnet (that is, its CoA is taken from an MNP).

Additionally, mobile networks can be *nested*. A mobile network is said to be nested when it attaches to another mobile network and obtains connectivity through it (refer to Figure 3). An example is a user who enters a vehicle with his personal area network (mobile network 2) and connects, through a mobile router—like a Wi-Fi enabled PDA—to the network of the car (mobile network 1), which is connected to the fixed infrastructure.

Figure 3: Nested Mobile Network: Operation of the NEMO Basic Support Protocol (multiangular routing)



Protocol Details: NEMO Versus Mobile IPv6

The NEMO Basic Support Protocol is an extension of the solution proposed for host mobility support, *Mobile IPv6* (MIPv6)[9].

In Mobile IPv6, three mechanisms support the mobility of a host: movement detection, location registration, and traffic tunneling. The NEMO Basic Support Protocol extends some of these mechanisms to support the movement of complete networks. These mechanisms are described next, with those parts that are different from the Mobile IPv6 protocol highlighted.

Movement Detection

In Mobile IPv6, the host needs to discover its own movement, so it can proceed with the required signaling and operations that allow its transparent mobility. Mobile IPv6 defines a generic movement-detection mechanism based on the *Neighbor Discovery Protocol*[10], which basically consists of listening to *Router Advertisements* (RAs). Routers send these router-advertisement messages, both periodically and in response to a *Router Solicitation* message issued by a host. By looking at the information contained in the router advertisements, a host can determine whether or not it has moved to a new link.

The NEMO Basic Support Protocol does not introduce any change on the movement-detection mechanisms that a mobile router can use.

Location Registration

When a host moves to a new network, it has to configure a new IPv6 address on the visited link (belonging to the IPv6 address space of that visited network): the CoA, and inform the home agent of the movement. In Mobile IPv6, the mobile node (that is, a mobile host) informs its home agent of its current CoA using a mobility message called the *Binding Update* (BU). This message is carried in an IPv6 datagram using a special extension header defined by Mobile IPv6 to encapsulate all messaging related to the creation and management of mobility bindings, called the *mobility header.* The binding-update message contains information required by the home agent to create a mobility binding, such as the home address of the *Mobile Node* (MN) and its CoA, where the home agent should encapsulate all the traffic destined to the mobile node. The home agent replies to the mobile node by returning a *Binding Acknowledgement* (BA) message.

The NEMO Basic Support Protocol extends the binding-update message to convey the following additional information:

- *Mobile Router Flag* (R): The mobile router flag is set to indicate to the home agent that the binding update is from a mobile router. A mobile router can behave as a mobile host: by setting this flag to 0, the home agent does not forward packets destined for the mobile network to the mobile router, but forwards only those packets destined to the home address of the mobile router.

- *Mobile Network Prefix Option:* This option is in the binding update to indicate the prefix information for the mobile network to the home agent. There could be multiple mobile network prefix options if the mobile router has more than one IPv6 prefix in the mobile network and wants the home agent to forward packets for each of these prefixes to the current location of the mobile router.

When the NEMO Basic Support Protocol is used to provide mobility to a complete network, only one binding-update or binding-acknowledgement signaling messages exchange is performed, whereas if the Mobile IP protocol were used by all the nodes of an *N*-node network, *N* × (Binding-update or Binding-acknowledgement) signaling messages synchronized exchanges would be required—usually referred to as a "binding-update signaling storm."

Mobile IPv6 defines a route-optimization mechanism that enables direct path communication between the mobile node and a correspondent node (avoiding traversal of the home agent). This route optimization is achieved by allowing the mobile node to send binding-update messages also to the correspondent nodes. In this way the correspondent node is also aware of the CoA, where the home address of the mobile node is currently reachable. A special mechanism—called the *Return Routability* (RR) procedure—is defined to prove that the mobile node has been assigned (that is, "owns") both the home address and the CoA at a particular moment in time[11], and therefore provides the correspondent node with some security guarantees.

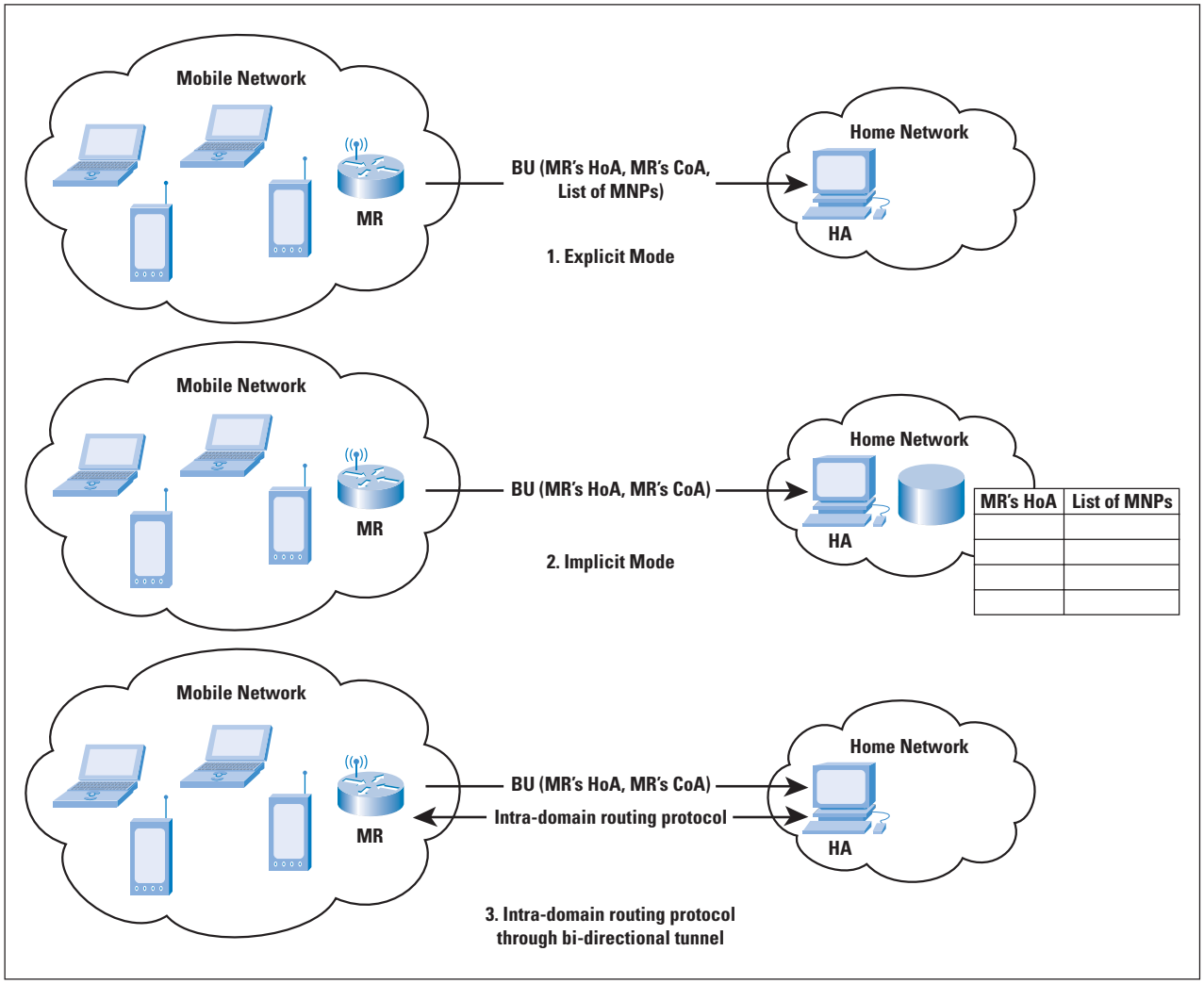
Because of the nature of the network-mobility scenario, the task of providing mobile networks with route-optimization support becomes more complex. The IETF is currently working on this topic[12, 13, 14].

Traffic Tunneling

In Mobile IPv6, after the mobile node has successfully registered its current location, the home agent starts encapsulating the data traffic destined to the mobile node toward its CoA.

In a NEMO scenario, the home agent forwards not only those IP datagrams arriving at the home network that are destined to the home address of the mobile router, but also all the traffic addressed to any of the mobile-network prefixes managed by the mobile router. The home agent can determine which prefixes belong to the mobile router in three different ways (refer to Figure 4):

Figure 4: NEMO Basic Support Modes of Operation



- *Explicit mode:* The mobile router includes one or more mobile network prefix options in the binding-update message that it sends to the home agent. These options contain information about the mobile-network prefix(es) configured on the mobile network.
- *Implicit mode:* The mobile router does not include prefix information in the binding-update message it sends to the home agent. The home agent determines the mobile-network prefix(es) owned by the mobile router by using any other mechanism (the NEMO Basic Support Protocol does not define any, leaving this prefix determination open to be implementation-specific).

  One example would be manual configuration at the home agent mapping the home address of the mobile router to the information required for setting up forwarding for the mobile network.

- *Intradomain Dynamic Routing Protocol through the bidirectional tunnel:* Alternatively to the previous two modes of operation, the home agent and the mobile router can run an intradomain routing protocol (for example, *Routing Information Protocol next generation* [RIPng] or *Open Shortest Path First* [OSPF]) through the bidirectional tunnel. The mobile router can continue running the same routing protocol that it ran when attached to the home link.

Fragmentation may be needed to forward packets through the tunnel between the mobile router and the home agent. In this case, the other end of the tunnel (the home agent of the mobile router) must reassemble the packet before forwarding it to the final destination. This requirement does not contradict the fact that *intermediate* IPv6 routers do not fragment (as opposed to IPv4), because the mobile router and home agent are the actual *ends* of the tunnel.

Performance of the NEMO Basic Support Protocol

The NEMO Basic Support Protocol relies on the creation of a bidirectional tunnel between the mobile router and the home agent to provide transparent mobility support to a complete network. The use of this tunnel causes an additional overhead of 40 bytes per packet, because of the extra IPv6 header added by the encapsulation. The effect of this overhead might be relevant for applications that generate small packets, such as *voice-over-IP* (VoIP) packets, because the 40-byte added overhead may be even bigger than the actual VoIP payload.

The end of the bidirectional tunnel at the side of the mobile router needs to be updated each time the mobile network moves (and also periodically to refresh the binding at the home agent), to reflect the current location of the mobile router. This updating is achieved by the binding-update or binding-acknowledgement signaling exchange between the mobile router and the home agent. As stated previously, only one exchange (two packets, one in each direction) is required per movement, regardless of the number of MNNs that are attached to the mobile router—one of the main advantages of using the NEMO Basic Support Protocol on the mobile router instead of Mobile IPv6 on every node of the mobile network, because the signaling generated by a complete moving network (composed of numerous nodes) is the same as the one generated by a single moving node.

Conclusions

The NEMO Basic Support Protocol[1] extends the functions of Mobile IPv6 to support the mobility of complete networks. The current specification supports basic mobility, and the IETF is currently working on new enhancements and extensions to provide route-optimization support, multihoming capabilities, and IPv4 support.

Some implementations of the NEMO Basic Support Protocol are already available. For example, the latest Cisco IOS® Software releases provide network mobility support. Open-source implementations also exist, such as the *NEMO Platform for Linux* (NEPL) (**http://www.mobile-ipv6.org/**) and SHISA (**http://www.mobileip.jp/**), for Linux and *Berkeley Software Distribution* (BSD) operating systems, respectively.

References

[1] Vijay Devarapalli, Ryuji Wakikawa, Alexandru Petrescu, and Pascal Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963, January 2005.

[2] Jon Postel, "Internet Protocol," RFC 791, September 1981.

[3] Iljitsch van Beijnum, "IPv6 Internals," *The Internet Protocol Journal*, Volume 9, No. 3, September 2006.

[4] Stephen E. Deering and Robert M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1998.

[5] Ralph Droms, "Dynamic Host Configuration Protocol," RFC 2131, March 1997.

[6] Ralph Droms, Jim Bound, Bernie Volz, Ted Lemon, Charles E. Perkins, and Mike Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3315, July 2003.

[7] William Stallings, "Mobile IP," *The Internet Protocol Journal*, Volume 4, No. 2, June 2001.

[8] Charles E. Perkins, "IP Mobility Support for IPv4," RFC 3344, August 2002.

[9] David B. Johnson, Charles E. Perkins, and Jari Arkko, "Mobility Support in IPv6," RFC 3775, June 2004.

[10] Thomas Narten, Erik Nordmark, and William A. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)," RFC 2461, December 1998.

[11] Pekka Nikander, Jari Arkko, Tuomas Aura, Gabriel Montenegro, and Erik Nordmark, "Mobile IP Version 6 Route Optimization Security Design Background," RFC 4225, December 2005.

[12] Chan-Wah Ng, Pascal Thubert, Masafumi Watari, and Fan Zhao, "Network Mobility Route Optimization Problem Statement," Internet Draft, Work in Progress, September 2006. **draft-ietf-nemo-ro-problem-statement-03.txt**

[13] Chan-Wah Ng, Fan Zhao, Masafumi Watari, and Pascal Thubert, "Network Mobility Route Optimization Solution Space Analysis," Internet Draft, Work in Progress, September 2006. **draft-ietf-nemo-ro-space-analysis-03.txt**

[14] María Calderón, Carlos J. Bernardos, Marcelo Bagnulo, Ignacio Soto, and Antonio de la Oliva, "Design and Experimental Evaluation of a Route Optimisation Solution for NEMO," *IEEE Journal on Selected Areas in Communications* (J-SAC), Issue on Mobile Routers and Network Mobility, Volume 24, Number 9, pages 1702–1716, September 2006.

CARLOS J. BERNARDOS received a telecommunication engineering degree in 2003, and a Ph.D. in telematics in 2006, both from University Carlos III of Madrid. His Ph.D. thesis focused on Route Optimisation for Mobile Networks in IPv6 Heterogeneous Environments. He has been working as a research and teaching assistant in Telematics Engineering since 2003. His current work focuses on IP-based mobile communication protocols. E-mail: **cjbc@it.uc3m.es**

IGNACIO SOTO received a telecommunication engineering degree in 1993, and a Ph.D. in telecommunications in 2000, both from the University of Vigo, Spain. He was a research and teaching assistant in telematics engineering at the University of Valladolid from 1993 to 1999. In 1999 he joined University Carlos III of Madrid, where he has been an associate professor since 2001. His research activities focus on mobility support in packet networks and heterogeneous wireless access networks. E-mail: **isoto@it.uc3m.es**

MARÍA CALDERÓN is an associate professor at the Telematics Engineering Department of University Carlos III of Madrid. She received a computer science engineering degree in 1991 and a Ph.D. degree in computer science in 1996, both from the Technical University of Madrid. She has published more than 20 papers in the fields of advanced communications, reliable multicast protocols, programmable networks, and IPv6 mobility. E-mail: **maria@it.uc3m.es**

More ROAP: Routing and Addressing at IETF68

by Geoff Huston, APNIC

Over the past year or so we have seen a heightened level of interest in Internet routing and addressing. Speculation regarding the future role of the Internet raises the possibility of the Internet supporting as many as hundreds of billions of chattering devices. What does such a future imply in terms of the core technologies of the Internet? Consideration of this topic has prompted a critical examination of the architecture of the Internet, including the scaling properties of routing systems, the forms of interdependence between addressing plans and routing, and the roles of addresses within the architecture.

The March 2007 meeting of the IETF, IETF68, saw some further steps in analysing these topics, and many sessions addressed aspects of routing and addressing. This article reports on these sessions, and includes some conjecture as to what lies ahead.

Plenary ROAP – The Plenary Session on Routing and Addressing

The plenary session presented an overview of the topic, looking at the previous initiatives in routing and addressing, as well as providing some perspectives on the current status of work in this area. There are concerns that the technology platform cannot scale by further orders of magnitude without some changes. Also of concern are the scalability of routing, the "transparency" of the network, renumbering questions, provider-based addressing, and service and traffic engineering and routing capabilities—and these concerns are potentially even more relevant and challenging for tomorrow's Internet.

Our routing technology does not localize the external effects of local configuration choices. Far from being a protocol that damps instability, the *Border Gateway Protocol* (BGP) is a highly effective amplifier of noise components of routing events. So although it is a remarkably useful information-dissemination protocol, the properties of BGP in an ever-more connected world with ever-finer granularity of information raise some questions about its scaling properties. Will the imposed "noise" of the behaviour of the protocol completely swamp the underlying information content? Will we need to deploy disproportionately larger routers to support a larger network? The prospect here is that routing may become far less efficient because as we simultaneously increase the degree of interconnection and the information load, the inability to effectively localize information creates a far greater load on network routing.

In addition to these observations about routing, there is the continuing suspicion that the semantic load of addresses in the Internet architecture, where an address simultaneously conveys the concepts of "who," "where," and "how," contributes to routing load.

To what extent the semantic intent of endpoint identity (or "id") can be separated from the semantic intent of network location and forwarding lookup token (or "loc") is a question of considerable interest. Although the current IP address semantics removes the need to support an explicit mapping operation between identity and location, the cost lies in the inability to support an address plan that is cleanly aligned to network topology, and the inability to cleanly support functions associated with device or network mobility. In the end it is the routing system that carries the consequent load. The questions in this area include an evaluation of the extent to which identity can be separated from location, and the effect of such a measure on the operation of applications. How much of today's Internet architecture would be affected by such a change, and what would be the resultant benefits if this measure were deployed? Are we necessarily looking at a single model of such an id/loc split, or should we think about this scenario in a more general manner with numerous potential id/loc splits?

Obviously this study of routing and addressing, and the related aspects of name space attributes and mapping and binding properties, has a very broad scope. The larger question posed here is whether we can defer resolution of this problem to a comfortably distant future, or whether its effect on the present network is imminent. Are we accelerating toward some form of near-term technical limit that will cause a significant disruptive event within the deployed Internet, and will volume-based networks economics hold or will bigger networks start to experience disproportionate cost bloat—or worse? Is it time to be alarmed?

The unallocated IPv4 address pool will certainly be exhausted in the coming years, but this sense of alarm over routing and addressing is more about whether there are real limits in the near future in the capability to continue to route the Internet within the deployed platform, using the current technologies, and working within current cost-performance relationships irrespective of whether the addresses in the packet headers are 32 or 128 bits in size. There was a strong sense of "Don't panic!" in the plenary presentation, with the relatively confident expectation that BGP will be able to carry the routing load of the Internet over the next 3 to 5 years without the need for major protocol "surgery," and that Moore's Law will continue to ensure that the capacity and speed of hardware will track the anticipated growth rates. Expectations are that the current technologies and cost-performance parameters will continue to prevail in this time frame.

The *Internet Engineering Steering Group* (IESG) has followed the *Internet Architecture Board's* (IAB's) initiative and has begun working with a focus group, the *Routing and Addressing Problem Directorate* (ROAP), to refine the broad space into many more specific work areas, and has assumed a role of coordination and communication across the related IETF activities.

In addition, because a relatively significant research agenda is posed by such long-term questions, the *Routing Research Group* of the *Internet Research Task Force* (IRTF) has been rechartered and, judging by the participation at its most recent meeting, effectively reinvigorated to investigate various approaches to routing that take us well beyond tweaking the existing routing toolset.

Internet ROAP – The Internet Area Meeting

The Internet Area meeting concentrated on aspects of this approach of supporting an identifier/locator split within the architecture of the Internet, and gathering some understanding as to whether this approach would assist with routing scaling. One of the important considerations in this area is working through what could be called boundary conditions of the study. For example, is this matter purely one for protocol stacks within an endpoint, or should distributed approaches that have active elements within the network also be considered? To what extent should a study consider mobility, traffic engineering, *Network Address Translation* (NAT), and *Maximum Transmission Unit* (MTU) behaviour? What appears to be clear at the outset is that this network is not a "clean-slate" network, and any approach should be deployable on the existing infrastructure, should use capability negotiation to trigger behaviours so that deployment can be incremental and piecemeal, should allow existing applications and their identity referential models to operate with no changes, and, hopefully, should have a direct benefit to those parties who decide to deploy the technology.

From the routing perspective, the overall desire is to reduce the growth rates of the interdomain routing space. The desired intent is to reduce the amount of information associated with locators so that locators reflect primarily network topology in such a way that the locators can be efficiently aggregated within the routing system that attempts to maintain a highly stable view of the network topology.

More detailed consideration of the implications of disambiguating aspects of identity from those of network location involves many dimensions—including the structure of the spaces—the mapping functions, and the practicalities of any form of deployment of such a technology.

A critical topic appears to be how an identity-mapping function relates to the forwarding-mapping function. Assuming that the existing name spaces remain unaltered, then the resultant framework appears to require distinct "name-to-identifier" and "identifier-to-locator" mappings and a "locator-to-forwarding" mapping. Where these mapping functions should be performed, who should perform them, when they should be performed, the duration of the validity of the outcomes, whether the mapping function outcomes are relative or universal, the scope and level of granularity in time and space of the map elements, the security of these mapping functions, and whether there is a simple operation in each mapping function or multiple operations all remain undefined at this point.

Other questions include whether the mapping is explicit or implicit, what evidence of a previous mapping operation is held in a packet in a visible manner, and what is occluded from further inspection after the mapping operation has been performed. In addition, what level of state is required in each host, and is there true end-to-end transparency—at what level?

It is likely, at least at this stage of the study, that such a split can have a variety of approaches, both in the intended roles of identifier and location tokens and in their binding. The expectation at this stage of the study is that further ideas will surface, and such ideas will be helpful rather than distracting. It is unclear if a single solution can emerge from this activity, or whether different actors have a sufficiently different set of relative priorities that multiple approaches—each of which expresses different prioritization of functions—are viable longer-term outcomes.

The critical consideration here is that it is unlikely that scaling routing over the longer term to a much larger network is simply a matter of just changing the operation of the routing system itself. Real improvement in this area appears to also require an understanding of the meaning of the objects, or "addresses," that are being passed within the routing system. The motivation for opening up the identifier or locator space within the Internet area appears to be strongly tied to the notion that if you can unburden some of the roles of the addresses used in routing, and treat these routed tokens as unadorned network locality tokens, then you can gain some additional capability in routing.

Routing ROAP – The Routing Area Meeting

The first part of the Routing ROAP session looked at the trends in the routing system over 2005 and 2006. The overall trend appears to be a system that is increasingly densely interconnected, carrying more information elements, each of which expresses finer levels of granularity in reachability. There appears to be two forms of dynamic BGP load: the BGP "supernova" that burst with an intense BGP update load over some weeks and then disappear, and "background radiation" generators that appear to be unstable at a steady update rate for months or even the entire year.

In looking at scaling the BGP routing environment, one response is that of behavioural changes in local instances of BGP that reduce the potential for unnecessary updates to be propagated beyond a "need-to-know-now" radius. Another response is to consider changes to BGP in terms of additional attributes to BGP updates—such as a "withdrawal-at-origin" flag, or selective advertisement of "next best path"—both of which are intended to limit the span of advertised intermediate transitions while the BGP distance vector algorithm converges to a stable state.

It appears that we could improve our understanding of the operational profile of the routing space, looking particularly at the various forms of pathological routing behaviours and comparing these behaviours against the observations of known control points. Such a study may also lead to some more effective models of projections of the size of the routing space in the near- and medium-term future, and allow some level of quantification as to what "scaling of the routing space" actually implies.

The second part of the Routing ROAP session considered the current status of the routing world, updating some of the observations made at the IAB Routing Workshop and outlining some further perspectives on this space. One critical perspective on BGP is the behaviour of BGP under load. It was noted that most BGP implementations use adaptive responses to peer load, so that BGP attempts to ensure that its peer receives only the most current state information when the peer signals that it is not keeping pace with the update rate.

Another critical factor is the nature of "convergence" in BGP. The claim was made that this problem was the biggest, yet least important, problem with BGP. Convergence delays can be mitigated by *Graceful Restart, Nonstop Routing,* and *Fast Reroute*. One of the measures that exacerbates convergence is the use of *Route Reflectors*. The model of information hiding or Route Reflectors is intended to reduce the number of BGP peer sessions and the update load, but the benefits they do achieve are at the cost of slower convergence with a higher message rate during the intermediate-state transitions. Perhaps it is appropriate to consider small-scale changes to BGP behaviour to mitigate the transient BGP update bursts caused by path hunting, including those already mentioned of "withdrawal-at-origin" notification and propagation of backup paths.

The approach advocated here is based on the perspective that BGP is not in danger of imminent collapse, and there is still considerable "headroom" for BGP operation in today's Internet.

More ROAP?

The routing space is a classic example of the commons, where each party can use routing to solve a multitude of business problems. This includes, for example, using routing to perform load balancing of traffic over a set of transit providers, using a "spot market" in Internet transit services, creating differentiated transit offerings using more specific routes and selective advertisements. The ultimate cost of these local efforts in optimising local business outcomes lies in the increasing bloat in the routing system and the consequent escalation in costs across the entire network in supporting the routing system. There is no way to impose administrative controls on the global routing system, nor have we been able to devise an economic model of routing where the incremental costs of local routing decisions are visible to the originator as true economic costs for the business, and the benefit of a conservative and prudent use of the routing system reaps economic dividends in terms of relatively lower costs for the business.

Like the commons, there are no effective feedback mechanisms to impose constraint on actors in the routing space. Also, like the commons, there is the distinct risk that the cumulative effect of local actions in routing creates a situation that pushes the routing system, either as a whole or in various locales, into a nonfunctioning state.

Whether it needs a sense of urgency to motivate the work, or a sense that there can and should be a better way to plan a future than crude crisis management, the underlying observation is that the routing and address world is fundamental to tomorrow's Internet. Unless we make a concerted effort to understand the various interdependencies and feedback systems that exist in the current environment, and understand the interdependences that exist between network behaviours and routing and addressing models, then I'm afraid that the true potential of the Internet will always lie within our vision—but frustratingly just beyond our grasp.

Further Reading

Following are references to further material on this topic, as presented at IETF68:

- **http://tools.ietf.org/html/draft-iab-raws-report-01**
- **http://submission.apricot.net/chatter07/slides/future_of_routing/apia-future-routing-john-scudder.pdf**
- **http://submission.apricot.net/chatter07/slides/future_of_routing/apia-future-routing-jari-arkko.pdf**
- **http://www3.ietf.org/proceedings/07mar/slides/plenaryw-3.pdf**
- **http://www3.ietf.org/proceedings/07mar/agenda/intarea.txt**
- **http://www3.ietf.org/proceedings/07mar/agenda/rtgarea.txt**
- **http://www1.tools.ietf.org/group/irtf/trac/wiki/RRG**
- **http://www.ietf.org/IESG/content/radir.html**

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. The author of numerous Internet-related books, he is currently the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of the Internet Society from 1992 until 2001. E-mail: **gih@apnic.net**

Opinion: Is It Time to Replace SMTP?

by Dave Crocker, Brandenburg InternetWorking

The first Internet (ARPANET) e-mail, sent 35 years ago, was remarkably similar to a basic text e-mail of today: From, To, CC, Subject, Date, followed by lines of text, and the familiar @-sign in addresses. The right side of the address changed from a simple string into the multilevel domain name that we now use. The body can now be a set of multimedia attachments rather than just lines of text, but it can still be in its original, simpler form. The means of moving mail was the *File Transfer Protocol* (FTP) in the early 1970s. The current mechanism, the *Simple Mail Transfer Protocol* (SMTP)[1a, 1b], was not created until 10 years later, but a mere 25 years of use is not bad, either.

All of the technical specifications for e-mail have undergone many changes over the years, but a core requirement has been to protect the installed base of users and operators by incrementally adding features as options, rather than by performing wholesale replacement of any infrastructure service component. E-mail has changed the way we communicate, yet it is also now viewed as having a serious problem: As the Internet grew, it acquired the full mixture of participants, some of whom do not make nice neighbors.

Frustration with the effect of abusive users is often expressed as a belief that the solution lies in replacing some or all of the core technology of the e-mail service, or even by moving to an entirely different paradigm, such as querying Webpages using *Really Simple Syndication* (RSS)[2]. Although different paradigms make sense for some forms of human communications, what is forgotten in these pleas for massive change is the power of the classic mail model, whether by paper or by electrons: Spontaneous or occasional communication requires the ability to "push" the message to the recipient, without prior arrangement. This ability is, of course, also what leaves the door open for abuse—anyone may walk in, uninvited and unwanted.

The alternative proposals might work well enough for ongoing, regular communication among people who already know each other. And for most of us, that is probably 80 percent of our exchanges, or more. Unfortunately, as soon as anyone starts worrying about the remaining 20 percent, these alternative approaches require cascading hacks, producing a design that looks no better that what we have today, except that it based strictly on theory rather than decades of practice. It is easy for a paper proposal to beat a deployed system; making it work as promised is, of course, more difficult.

Mantra

I have developed a simple mantra, in response to calls for replacing today's Internet mail:

0. The basic problems we are experiencing with e-mail are really based on undesirable social behaviors, long popular outside the Internet. The Internet enables broader reach, to more victims, and in much shorter time spans, but the core misbehaviors have existed for all of recorded human history. We should not assume that there are technical solutions to social problems.
1. The beginning of changing a human service is to gain community consensus about the change that is needed, because a mechanism will not be successful unless it is perceived as needed. Only then can the engineers work on designing the change.
2. When there is community consensus about the way that e-mail needs to be changed, the folks who are currently contributing to its 35-year evolution need to try to find a way to add the desired features to the existing service. Given the record of accomplishment of e-mail, the odds seem favorable that any new requirement can also be satisfied without disrupting the installed base.
3. When that effort fails, it will be time to create a replacement infrastructure.

Alas, as those who track e-mail abuse technical discussions are well aware, we have not completed Step 1. As soon as we try to formulate community consensus about basic messaging communication policies, discussion devolves into cacophony or marginalized community fragments. It is certain that there will eventually be a change required for e-mail, which we cannot fit into the current service, but we do not yet have any evidence that e-mail abuse is going to produce that requirement.

Trust Models

One hopeful sign is that we do have a solid set of efforts to evolve e-mail to support mechanisms that are based on trust. This evolution begins with the ability to associate a validated identity to a message and then requires assessing the "safety" of that identity's owner. Until recently, only the IP address of the last-hop sending SMTP server could be used as an identifier. Using addresses as identifiers sounds reasonable at first glance, but turns out to have long-term scaling and administrative problems. As a result, there has been a broad effort to find ways to use domain names, which are more stable, and they align better with organizational boundaries. This process is well under way, with the recent IETF standardization of the *Domain Keys Identified Mail* (DKIM)[3] message-signing specification, as well as path-based registration schemes, such as Sender-ID[4] and SPF[5].

That took about 5 years. And now comes the hard part: developing a range of *assessment mechanisms*—sometimes generically called *reputation services*—that satisfy requirements for quality, strength, convenience, and stability. Assessment services tell recipients whether the author of the message, or the service that sent it, can be trusted. Some mechanisms need to work for small groups, others need to work for mass-market business-to-consumer mailings, and others need to work among business partners. A few startup companies have recently joined the few, surviving volunteer services, to satisfy this need. It is too early to tell whether they will suffice, or whether additional services will be needed. What is important is that these services are generally regarded as producing good results.

For the long term it seems likely that this capability will result in an Internet mail service that is logically split into two types of traffic. One has substantial trust associated with its messages, so that they can be delivered with a reasonable degree of comfort. The other is the current, open-to-all service that requires heavy filtering and the use of various heuristics, to reduce the effect of abuse mail. If the first traffic flow is sufficiently successful, filters for the second can become much more stringent. The aggregate effects of these changes will be that wanted mail is likely to be received and identified much more reliably, and unwanted mail is more likely to be rejected.[8, 9]

So the current Internet mail technical infrastructure is safe, right? Well, maybe.

Enhancements?

What gets less attention, but perhaps should worry us more, is the general lack of user-level functional enhancement for e-mail. What users can do with e-mail, today, is pretty much the same as they could do 25 years ago. The evolution of Internet mail has been primarily in support of performance, reliability, and scaling. Although important, they have not produced functional changes that are apparent to end users. Human communication is a very rich space, yet most e-mail is limited to a narrow range of styles: person-to-person informal communications, and informal, unstructured group communications. Toss in some very basic, one-way "transactional" mail, such as order confirmations from businesses to their customers, and that about covers it.

Instead, new functions for human collaboration have tended to appear in new services. *Instant Messaging* (IM), blogging, and wikis are the most popular examples. In each case, they rely on a centralized service, rather than the highly distributed model that e-mail uses. Users must all go to a single, centralized address to obtain a given service. Most of the IM world does not even know that there are two (!) Internet standards for distributed IM—*Extensible Messaging and Presence Protocol* (XMPP)[6] and SIMPLE[7]. Even for these standards, most of their production use tends to be within noninteroperable, centralized services.

Is there something about e-mail that is a barrier to functional enhancements for end users?

For these new services, the interservice relaying that is at the core of e-mail is absent. Indeed, centralized services are easier to create and operate than are distributed services, but they also carry scaling, administration, and control challenges. So the issue is not so much what is easier, but who will do the work—and when? With a centralized service, all the interesting work is done by the single provider. For a distributed model, like e-mail, the work is shared across participating organizations. The Internet was designed to avoid single points of failure (and failure), so it is ironic that these new services risk exactly these problems.

For a distributed model, like e-mail, to add end-user functions, useful adoption is required by all user software that participates, and possibly by all the intermediate, relaying services. The adoption is in three parts: agreeing on the enhancement, modifying existing software, and making it available to users. These are daunting barriers, so the appeal of centralized services is clear: a single organization decides what to change, changes it, and makes it available to end users with, at most, a natural software upgrade.

Interorganization partnerships provide the best argument for adoption of distributed services, because they do not naturally permit agreement on a central point of control. The counterforce is, again, the simplification (for the partners) that comes from agreeing to use independent third-party services. The scaling problem here is with end users having to juggle a large number of independent services. Note the emergence of IM clients that support a variety of independent IM services.

Perhaps the real danger to e-mail is not its wholesale and traumatic replacement, stemming from frustration about abuses, but a gradual attrition, as portions of its traffic move to services that evolve more quickly, but leave end users with a complicated array of narrow, specialized, and noninteroperable venues.

References

[1a] Postel, J. B., "Simple Mail Transfer Protocol," RFC 821, August 1982.

[1b] Klensin, J., "Simple Mail Transfer Protocol," RFC 2821, April 2001.

[2] Really Simple Syndication Specifications, **http://www.rss-specifications.com/rss-specifications.htm**

[3] Allman, E., et al., "DomainKeys Identified Mail (DKIM) Signatures," February 2007. *(RFC publication pending.)* **http://dkim.org/specs/draft-ietf-dkim-base-10.html**

[4] Lyon, J. and Wong, M., "Sender ID: Authenticating E-Mail," RFC 4406, April 2006.

[5] Wong, M. and Schlitt, W., "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1," RFC 4408, April 2006.

[6] Saint-Andre, P. (ed.), "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence," RFC 3921, October 2004.

[7] Campbell, B. (ed.), Rosenberg, J., Schulzrinne, H., Huitema, C., and Gurle, D., "Session Initiation Protocol (SIP) Extension for Instant Messaging," RFC 3428, December 2002.

[8] Crocker, D., "Challenges in Anti-Spam Efforts," *The Internet Protocol Journal,* Volume 8, No. 4, December 2005.

[9] Klensin, J., "Taking Another Look at the Spam Problem," *The Internet Protocol Journal,* Volume 8, No. 4, December 2005.

DAVE CROCKER is a principal with Brandenburg InternetWorking. He has authored or contributed to most Internet mail standards, and an assortment of e-mail products and businesses, as well as working on facsimile, security, e-commerce, and EDI. He received the 2004 *IEEE Internet Award* for his work on e-mail. Dave is a contributor to the development efforts for DKIM, CSV, and BATV, motivated by a strong desire to protect more than 30 years of professional investment that is being threatened by spamming. E-mail: **dcrocker@bbiw.net**

Fragments

ARIN Board Advises Internet Community on Migration to IPv6

The *American Registry for Internet Numbers* (ARIN) and the other *Regional Internet Registries* (RIRs) have distributed Internet Protocol version 6, IPv6, alongside IPv4 since 1999. To date, ARIN has issued both protocol versions in tandem and has not advocated one over the other. ARIN has closely monitored trends in demand and distribution for both protocol versions with the understanding that the IPv4 available resource pool would continue to diminish.

The available IPv4 resource pool has now been reduced to the point that ARIN is compelled to advise the Internet community that migration to IPv6 is necessary for any applications that require ongoing availability from ARIN of contiguous IP number resources. On 7 May 2007, the ARIN Board of Trustees passed the following resolution:

"Whereas, community access to *Internet Protocol* (IP) numbering resources has proved essential to the successful growth of the Internet; and,

Whereas, ongoing community access to *Internet Protocol version 4* (IPv4) numbering resources can not be assured indefinitely; and,

Whereas, *Internet Protocol version 6* (IPv6) numbering resources are available and suitable for many Internet applications,

Be it Resolved, that this Board of Trustees hereby advises the Internet community that migration to IPv6 numbering resources is necessary for any applications which require ongoing availability from ARIN of contiguous IP numbering resources; and,

Be it Ordered, that this Board of Trustees hereby directs ARIN staff to take any and all measures necessary to assure veracity of applications to ARIN for IPv4 numbering resources; and,

Be it Resolved, that this Board of Trustees hereby requests the ARIN Advisory Council to consider Internet Numbering Resource Policy changes advisable to encourage migration to IPv6 numbering resources where possible."

Implementation of this resolution will include both internal and external components. Internally, ARIN will review its resource request procedures and continue to provide policy experience reports to the Advisory Council. Externally, ARIN will send progress announcements to the ARIN community as well as the wider technical audience, government agencies, and media outlets. ARIN will produce new documentation, from basic introductory fact sheets to FAQs on how this resolution will affect users in the region. ARIN will focus on IPv6 in many of its general outreach activities, such as speaking engagements, trade shows, and technical community meetings. For more information, visit ARIN's IPv6 Information Center at:

`http://www.arin.net/v6/v6-info.html`

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

September 2007 Volume 10, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

For the last 10 or so years I have been involved with the organization of APRICOT, the *Asia Pacific Regional Internet Conference on Operational Technologies*. APRICOT has at its core a set of workshops featuring expert instructors with years of operational network experience. A recent addition to the APRICOT workshop program is a course focusing on Internet security in a multivendor environment. Our first article, written by Kunjal Trivedi from Cisco Systems, Inc., and Damien Holloway from Juniper Networks, is based on this workshop. It's not every day that you see an article co-authored by instructors from competing companies, but this is exactly the type of cooperation that is needed in order to deploy security in a multivendor network.

The rest of this issue is mostly devoted to IPv4 depletion and the transition to IPv6. The first article, by Geoff Huston, summarizes many of the concerns related to IPv4 depletion and IPv6 transition, and gives numerous pointers to further articles and documents of interest. Our second addressing-related article, by Iljitsch van Beijnum, looks more closely at the numbers relating to address allocation by the *Regional Internet Registries* (RIRs). The final article concerns some address blocks that are currently unassigned but actually in use. Leo Vegoda explains the potential problems that may arise when these blocks eventually become part of the RIR assignment pool.

We are pleased to announce a new online addition to this journal. *The Internet Protocol Forum* (IPF) available at **www.ipjforum.org** is designed to allow discussion of any article published in the printed edition of IPJ. In addition to article discussions, the forum will be used to provide updates and corrections, downloads, expanded versions of some articles, configuration and programming examples, and news and analysis that does not fall into our quarterly publication schedule. The IPF's editor and moderator is Geoff Huston, long-time contributor to this journal and chief scientist at APNIC. I am confident that IPF will become an important addition to IPJ, and I hope you will take the time to participate in the online discussions. Of course, you can always contact us at the usual e-mail address: **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

A Standards-Based Approach for Offering a Managed Security Service in a Multivendor Network Environment

By Kunjal Trivedi, Cisco Systems and Damien Holloway, Juniper Networks

As transport becomes a commodity, service providers are seeking new revenue sources and new ways to differentiate themselves. Managed security services address a growing market because business customers are struggling to comply with regulatory requirements such as the *Payment Card Industry-Data Storage Standards* (PCI-DSS), the *Sarbanes-Oxley Act,* the *Gramm-Leach Bliley Act, Health Insurance Portability and Accountability Act* (HIPAA), *Directive 2002/58/EC,* and the *Asia-Pacific Economic Cooperation-Organization for Economic Cooperation and Development* (APEC-OECD) initiative on regulatory reform. Increasingly, business customers recognize that outsourcing network security is less costly than staffing with highly specialized security personnel who can provide 24-hour incident detection and response. Another incentive for outsourcing is to free existing IT resources to focus on the core business.

A standards-based approach helps service providers take best advantage of the managed security service opportunity because it increases the potential breadth and depth of the service offering. Multivendor solutions are becoming the norm when deploying services on an integrated backbone. Therefore, standards simplify deployment and management, helping control operational costs and accelerating time to market.

Service providers are experiencing a growing need for skilled engineers who understand multivendor environments—the motivation for conducting a multivendor security workshop at the 2006 *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT 2006)[15], held in Perth, Australia, in February 2006. During the workshop (which was repeated again at APRICOT 2007 in Bali), participants successfully deployed and tested a multivendor service environment using *IP Security* (IPsec)-based Layer 3 *Virtual Private Networks* (VPNs)[1, 2, 3] over a *Border Gateway Protocol/ Multiprotocol Label Switching* (BGP/MPLS) core[4].

Technical Challenges

To offer managed security services, service providers need the following:

- A secure network infrastructure, including tools and techniques for risk mitigation
- Technical solutions for the customer's business needs, such as VPNs based on BGP/MPLS, IPsec, or both

- Web-based reporting tools that business customers can use to monitor the security service in accordance with *Service-Level Agreements* (SLAs). Service providers can scale cost-effectively by offering customers a secure, Web-based portal that shows open trouble tickets, security incident-handling detail, SLAs, and access reports that customers need to comply with regulations.

An effective managed security service requires tools and techniques to address the following challenges:

- *More sophisticated threats, and less time between vulnerability and exploitation:* In addition to worms and viruses, the service provider needs to protect its own and its customers' networks against *Denial-of-Service* (DoS) attacks. Today's botnets can launch thousands or even a million bots that carry out outbound DoS attacks. New varieties of worms have side effects similar to those of DoS attacks. These threats can take down the service provider infrastructure, thereby violating SLAs and eroding revenue.
- *A need for proactive rather than reactive threat response:* Many service provider security groups are stuck in reactive mode. Every network device and security system produces voluminous event logs every day, and vendors use different formats. Therefore, identifying security incidents in order to react to them can take hours or days—or not happen at all. The connection between two separate events in different parts of the network can easily escape human detection, especially when the clues are buried among tens of thousands of harmless events that took place around the same time.
- *Multivendor networks:* Network security and reporting are easier to achieve in single-vendor networks. Realistically, however, many service providers and business customers have multivendor networks, sometimes because of mergers and acquisitions. Even if the service provider itself has a single-vendor network, some of its customers will use other vendors' equipment.
- *Slow progress toward adopting IP Next-Generation Networks (IP NGNs):* When service providers complete the migration to IP NGN, they will achieve greater control, visibility, and operational efficiency. Until then, service providers will incur higher costs and labor requirements for support and migration.
- *A need to comply with industry standards from IETF and ITU:* Standards facilitate security in multivendor networks. MPLS helps ensure infrastructure security, whereas IPsec provides secure connectivity among the customer's branches and remote offices. By using industry standards, the service provider can select best-of-class products based on performance, features, or cost.

- *Scalability challenges:* The security operations center for a managed services provider cannot cost-effectively scale to process several million events for each customer. However, it can scale to process a few security-incident trouble tickets. Scalability hinges on the ability to minimize false positives. Products such as Cisco *Security Monitoring, Analysis and Response System* (MARS), IBM Micromuse, and NetIQ provide analysis and correlation of events from multiple elements in the IT infrastructure. They process events using consolidation, filtering, normalization, enrichment, correlation, and analysis techniques, and also notify IT staff about critical events.

Infrastructure Security in Multivendor Environments

Securing the service provider infrastructure requires the following common best practices:

- Point protection
- Edge protection
- Remote-triggered black-hole protection
- Source-address validation on all customer traffic
- Control-plane protection
- Total visibility into network activity

Point Protection

Before offering a managed security service, providers need to protect the backbone; security operations center or network operations center; *Authentication, Authorization, and Accounting* (AAA)[10, 11] server; and remote-access networks. Securing individual network devices requires enforcing AAA, controlling the type of packets destined to network devices, and performing regular configuration audits to ensure that no unauthorized changes have been made. Best common practices include:

- *Protect the backbone by locking down the vty and console ports:* This protection helps prevent unauthorized access to network devices.
- *Encrypt management commands that staff send to devices:* Use of the *Secure Shell* (SSH) protocol helps prevent hackers from obtaining passwords that they could later use to compromise the network. Service providers that use out-of-band management for device configuration should also encrypt this management traffic and restrict access to authorized personnel.
- *Deploy a AAA server:* Using a AAA server is preferable to relying on local authorization on the devices themselves because it enables centralized policy control. The AAA server controls a user's access to the device, or even the specific commands that the user is authorized to execute.

It is strongly recommended that service providers use TACACS+[13] authentication rather than *Remote Authentication Dial-In User Service* (RADIUS)[12] authentication. With RADIUS, traffic is sent in the clear between the AAA servers and network devices using the *User Datagram Protocol* (UDP), which defeats the use of SSH to encrypt logins and passwords. Open-source implementations of TACACS+ are available.

- *Use one-time passwords (OTPs):* To distribute one-time passwords, service providers can provide authorized users with a token card, soft token, or soft key. One-time passwords ensure that the user was authorized at the time of login, and was not an attacker who used a packet-sniffer program to intercept a password.
- *Protect the AAA infrastructure from DoS attacks:* Some service providers set up local accounts on routers and switches so that staff can log in if the AAA infrastructure is down, creating vulnerability. If the service provider does not secure management-plane access to the device, hackers can use SSH or Telnet and attempt a brute-force attack to crack the local account. The local account is often not as secure as an OTP because it is changed only once every 30 days, providing a longer window of opportunity for hackers to gain device access. It is strongly advised to not use default or easy-to-guess passwords. To prevent attacks against the AAA infrastructure, service providers should harden the infrastructure and consider placing the server behind a firewall with stateful inspection. Use *Access Control Lists* (ACLs), which are packet filters, on the firewall to restrict traffic between the AAA server and network devices only. Also be sure to distribute the AAA servers so that they do not create a single point of failure.
- *Regularly audit device configurations:* Frequently, the first indications of an attack, often unnoticed, are unauthorized commands executed on routers that change the configuration. An easy way to monitor configurations is using RANCID *(Really Awesome New Cisco Config Differ)*[14], a UNIX or Linux freeware tool that logs into each of the devices in the device table file, runs various show commands, processes the output, and sends e-mail messages reporting any differences from the previous collection to staff. RANCID works with routers from Cisco and other vendors. Another tool for auditing device configurations, the *Router Audit Toolkit* (RAT) assigns security scores to ACLs and other security best practices to show the relative security of routers.

Traditionally, service providers enforced policy at the process level, using vty ACLs, *Simple Network Management Protocol* (SNMP) ACLs, and others. Some service providers used ingress ACLs when possible. Today, it is far preferable to stop DoS traffic at ingress points: the peer edge, downstream and upstream routers, colocated network devices, and the customer access edge, enabling central policy enforcement and more granular protection schemes.

In addition, many network devices at the network edge have hardware acceleration, which provides far more robust resistance to attack than the process level.

Edge Protection

In many service provider networks, each core router is individually secured but still accessible to outsiders using SNMP or Telnet. Now service providers can supplement individual router protection with infrastructure protection that prevents undesired traffic from ever touching the infrastructure.

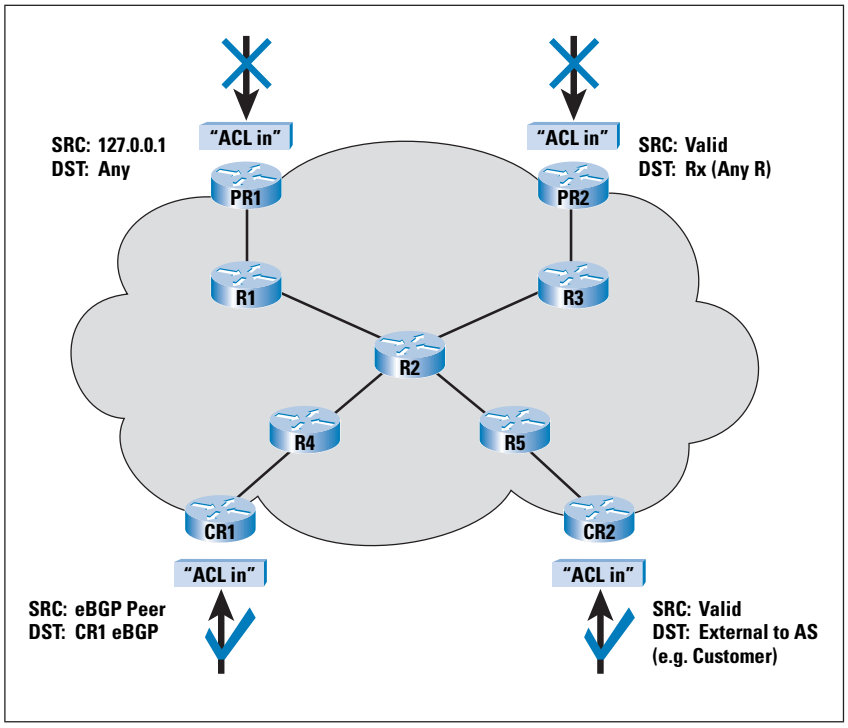


Figure 1: Protecting the Network Edge

The following steps help protect the network edge (Figure 1):

1. Classify the required protocols that are sourced from outside the *Autonomous System* (AS) access core routers, such as *external BGP* (eBGP) peering, *Generic Routing Encapsulation* (GRE)[5], and IPsec. (Examples of nonrequired protocols are SNMP and Telnet.) Classification can be performed using a classification packet filter or Cisco *NetFlow* telemetry. The classification packet filter comprises a series of permit statements that provide insight into required protocols. Gradually narrow down the list, keeping in mind that very few protocols need access to infrastructure equipment, and even fewer are sourced from outside the autonomous system. Summarize the IP address space as much as possible, for simpler and shorter ACLs. Be cautious: just because certain types of traffic appear in a classification packet filter or NetFlow telemetry data does not mean they should be permitted to pass through to the routers.

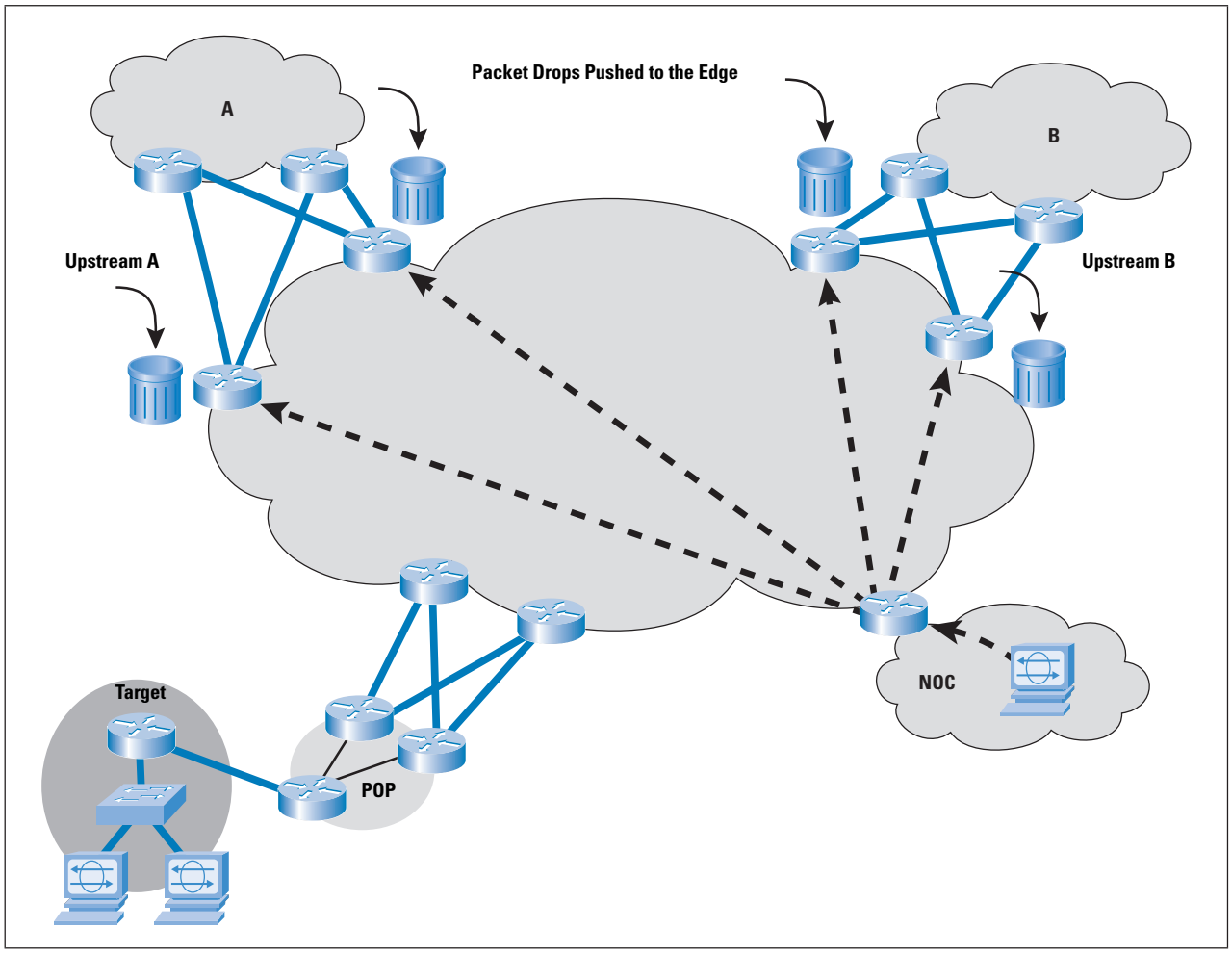
2. Begin filtering. Use an infrastructure packet filter to permit only the required protocols to access infrastructure-only address blocks, denying all other protocols. It is important to monitor the packet filter entry counters, because a high volume of hits, whether or not a protocol has been identified as required, might signal an attack. To permit transit traffic, use the following as the final line of the *Infrastructure ACL* (iACL): **`permit ip any any`**, protecting the core network with a basic iACL that admits only the required protocols. Note that iACLs also provide antispoof filtering by denying access to the space from external sources, denying the RFC 1918 space[6], and denying multicast source addresses. RFC 3330[7] defines special-use IPv4 addressing.
3. Further protect the core by identifying legitimate source addresses for the required protocols, such as external BGP peers and tunnel endpoints.
4. Deploy destination filters when possible.

Infrastructure packet filters at the edge of the network protecting the infrastructure are an effective first layer of defense. Service providers need additional forms of infrastructure protection for their older routers that do not support infrastructure packet filters and for packets that cannot be filtered with infrastructure packet filters.

Remote Triggered Black Hole Filtering

Remote Triggered Black Hole Filtering (RTBH) is among the most effective reaction and mitigation tools for DoS, *Distributed DoS* (DDoS), and backscatter tracebacks. It enables service providers to quickly drop DoS traffic at the network edge (Figure 2). Rather than sending commands to every router to drop DoS or other problem traffic, the service provider can deploy a trigger router that uses BGP to signal all other routers—just as fast as iBGP can update the network. In destination-based RTBH, all traffic headed to the destination under attack is dropped—the good traffic as well as the bad. In source-based RTBH, traffic from all or certain sources are blocked. The advantage of sourced-based RTBH is that service providers can whitelist certain addresses, such as the *Network Operations Center* (NOC) or route-name servers, so that they can continue providing services.

Figure 2: DoS Packets Dropped at the Network Edge



Source Address Validation on all Customer Traffic

Source address validation, defined in *Best Current Practices* (BCP) 38[8], prevents service provider customers from spoofing traffic—that is, sending IP packets out to the Internet with a source address other then the address allocated to them by the service provider. Best practices from BCP 38 are to filter as close to the edge as possible, filter precisely, and filter both the source and destination address when possible.

Every access technology has antispoofing mechanisms derived from BCP 38:

- Packet filters
- Dynamic packet filters that are provisioned to be AAA profiles; when a customer signs in with RADIUS, a packet filter is set up for the customer
- *Unicast Reverse Path Forwarding* (URPF)
- Cable-Source Verify and packet cable multimedia (cable)
- IP Source Verify and DHCP Snooping (Metro Ethernet)

To gain operational confidence in BCP 38, service providers can take a phased approach—for example, implementing it first on one port, then on a line card, then on an entire router, and then on multiple routers.

Control-Plane Protection

Protecting the infrastructure control plane helps prevent an attacker from taking down a BGP session and thereby causing denial of service. The exploits a service provider needs to prevent include saturating the receive-path queues so that BGP times out, saturating the link so that the link protocols time out, dropping the *Transmission Control Protocol* (TCP) session, and dropping the *Interior Gateway Protocol* (IGP), which causes a recursive loop-up failure.

Following are techniques for control-plane protection.

- *Generalized Time-to-Live (TTL) Security Mechanism (GTSM):* This technique protects BGP peers from multihop attacks. Routers are configured to transmit their packets with a TTL of 255, and to reject all packets with a TTL lower than 254 or 253. Therefore, a device that is not connected between the routers cannot generate packets that either router will accept.
- *Configuring routing authentication:* The *Message Digest Algorithm 5* (MD5) peer authentication feature instructs the router to certify the authenticity of its neighbors and the integrity of route updates. MD5 peer authentication can also prevent malformed packets from tearing down a peering session, and unauthorized devices from transmitting routing information. Be aware that MD5 peer authentication does not protect the router if an attacker compromises the router and begins generating bogus routing updates. Although it is not a panacea, MD5 peer authentication does raise the level of protection.
- *Customer ingress prefix filtering:* Prefix hijacking is an exploit in which a service provider customer announces an address space that belongs to another customer. The remedy is customer ingress prefix filtering, which enables service providers to accept only those customer prefixes that have been assigned or allocated to their downstream customers. For example, if a downstream customer has a **`220.50.0.0/20`** block, customers can announce this block only to their peers, and upstream peers accept this prefix only. Service providers can apply ingress prefix filtering to and from customers, peers, and upstream routers.

Visibility into Network Activity

To gain visibility into the network for early detection of security incidents, service providers can use open-source tools to analyze flow-based telemetry data, which is retrieved from routers and switches. Open-source tools for visibility into security incidents include RRDTool, FlowScan, Stager, and NTOP *Remote Monitoring* (RMON).

These tools provide information such as packets per second, bits per second, and traffic types. For example, RRDTool shows the number of *Domain Name System* (DNS) queries per second, according to record type. A spike in *Mail Exchange* (MX) Record queries might indicate that a customer's router has been compromised and is being used as a spam proxy. Similarly, a sharp increase in round-trip-time latency might indicate a DoS attack.

MPLS Security in a Multivendor Environment

In addition to securing the infrastructure, managed security service providers need to secure packets as they travel from one customer-edge router to another—regardless of the equipment the customer uses at the edge. Layer 3 VPNs meet this need. RFC 4364, which replaced RFC 2547bis, defines a BGP/MPLS IP VPN that creates multiple virtual routers on a single physical router: one virtual router for each customer.

In BGP/MPLS VPNs, *Customer Edge* (CE) routers send their routes to the *Service Provider Edge* (PE) routers. Customer edge routers at different sites do not peer with each other, and the customer's routing algorithms are not aware of the overlay. Data packets are tunneled through the backbone so that the core routers do not need to know the VPN routes. BGP/MPLS IP VPNs support either full mesh or partial mesh, although full mesh is more cost-effective.

A unique advantage of BGP/MPLS VPNs is that two service provider customers with overlapping IP addresses can connect across the service provider backbone. The router distinguishes between traffic from different companies by examining the label at the beginning of the packet, and then instantly forwards the traffic based on the *Label Switching Path* (LSP) that has been established for each customer's VPN. Eliminating the need to look at the packet in depth enables faster forwarding. That is, the service provider core does not impose any latency as packets pass between the provider edge routers.

IPsec Security in a Multivendor Environment

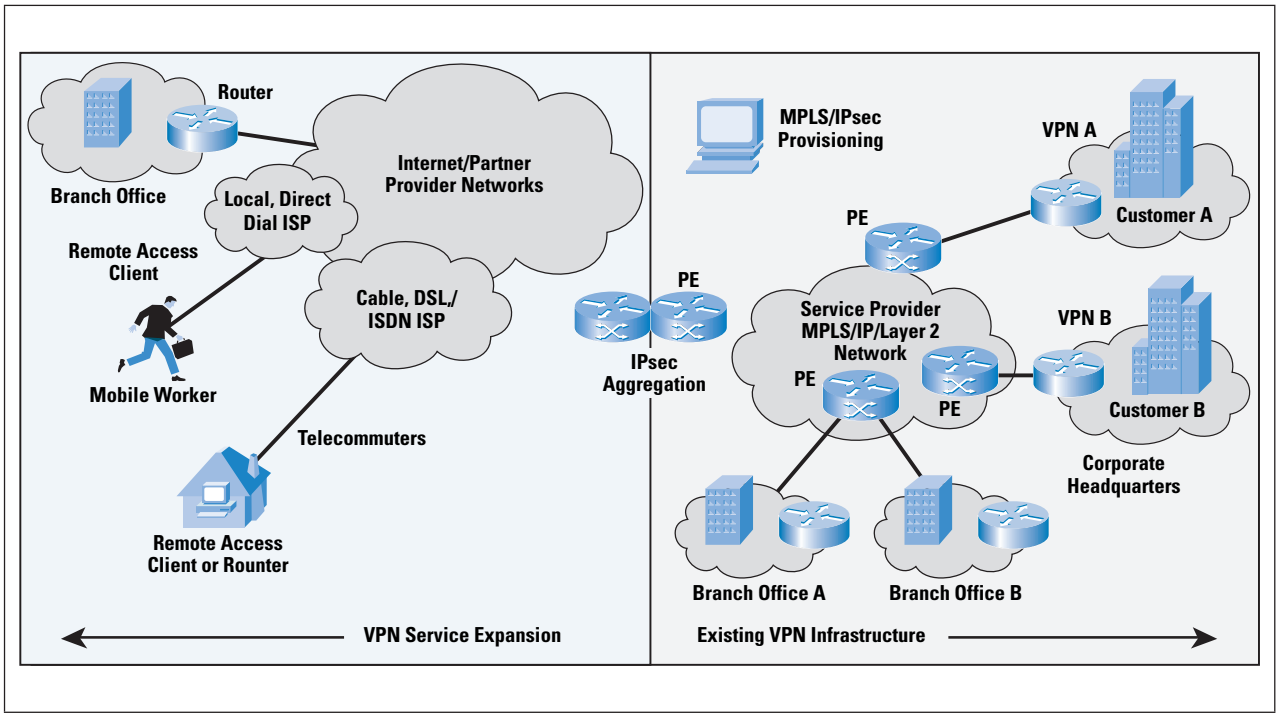
In addition to or instead of deploying a BGP/MPLS IP VPN, the service provider can extend its service to other partner provider networks using IPsec. The options are to use MPLS alone, IPsec alone, or a combination (Figure 3 on page 12). A retail customer that needs to comply with PCI-DSS, for example, needs IPsec or *Secure Sockets Layer* (SSL) encryption for payment card transaction data as part of its managed security service.

Table 1 summarizes the process based on the option the service provider selects. In the table, VPNA refers to one customer's VPN on a router that hosts VPNs for multiple customers.

Table 1: Comparing Packet Flow in IPsec VPNs, BGP/MPLS VPNs, and Combination VPNs

IPsec	BGP/MPLS VPN	BGP/MPLS VPN and IPsec
1. Host A in site 1 of VPNA sends packets to host B in site 2 of VPNA. 2. Routers A and B negotiate an Internet Key Exchange (IKE) [9] phase-one session in aggressive or main mode to establish a secure and authenticated channel between peers. 3. Routers A and B negotiate an IKE phase-two session to establish security associations on behalf of IPsec services. 4. Information is exchanged securely through an IPsec tunnel. 5. The tunnel is terminated.	1. Host A in site 1 of VPNA sends packets to host B in site 2 of VPNA. 2. Packet arrives on a VPN Route-Forwarding (VRF) VPNA interface on the PE1 router. 3. The PE1 router performs an IP lookup, determines the label stack and the outgoing core-facing interface, and forwards the packet to the MPLS core. 4. The packet is label-switched at each hop in the core until it reaches the penultimate hop router. At this point, the top label is popped before the packet is forwarded to the egress provider edge router. 5. The egress PE2 router performs a MPLS lookup and determines that it should remove the label before forwarding the packet to host B in site 2. 6. Router B in site 2 receives a regular IP packet and forwards it to host B.	1. Router A in site 1 and the associated PE1 router negotiate an IKE phase-one session in aggressive or main mode to negotiate a secure and authenticated channel between peers. 2. Router A and the PE1 router negotiate an IKE phase-two session to establish security associations on behalf of IPsec services so that information is exchanged securely through an IPsec tunnel. 3. Host A in site 1 of VPNA sends packets to host B in site 2 of VPNA. 4. The PE1 router, which is enabled with VRF-aware IPsec, creates a direct association through the IPsec tunnel that connects site 1 and the corresponding VRF ID (VPNA) on the provider edge router over the Internet. 5. Encrypted traffic arrives on an Internet-facing interface on the provider edge router A, which terminates the IPsec tunnel, decrypts the incoming packet, and forwards the plaintext packet to the VRF VPNA for further processing. 6. The PE1 router performs an IP lookup, determines the label stack and the outgoing core-facing interface, and forwards the packet to the MPLS core. 7. The packet is label-switched at each hop in the core until it reaches the penultimate hop router. At this point, the top label is popped before the packet is forwarded to the egress provider edge router. 8. The egress PE2 router performs a MPLS lookup and determines that it should remove the label before forwarding the packet to host B in site 2. Router B in site 2 receives a regular IP packet and forwards it to host B. 9. If site 2 is also reachable over the Internet and the egress PE2 router is enabled with VRF-aware IPsec, the packet is encrypted and sent to site 2 across the Internet over an IPsec tunnel. 10. Router B in site 2 terminates the IPsec tunnel, performs a regular IP lookup, and forwards the packet to host B.

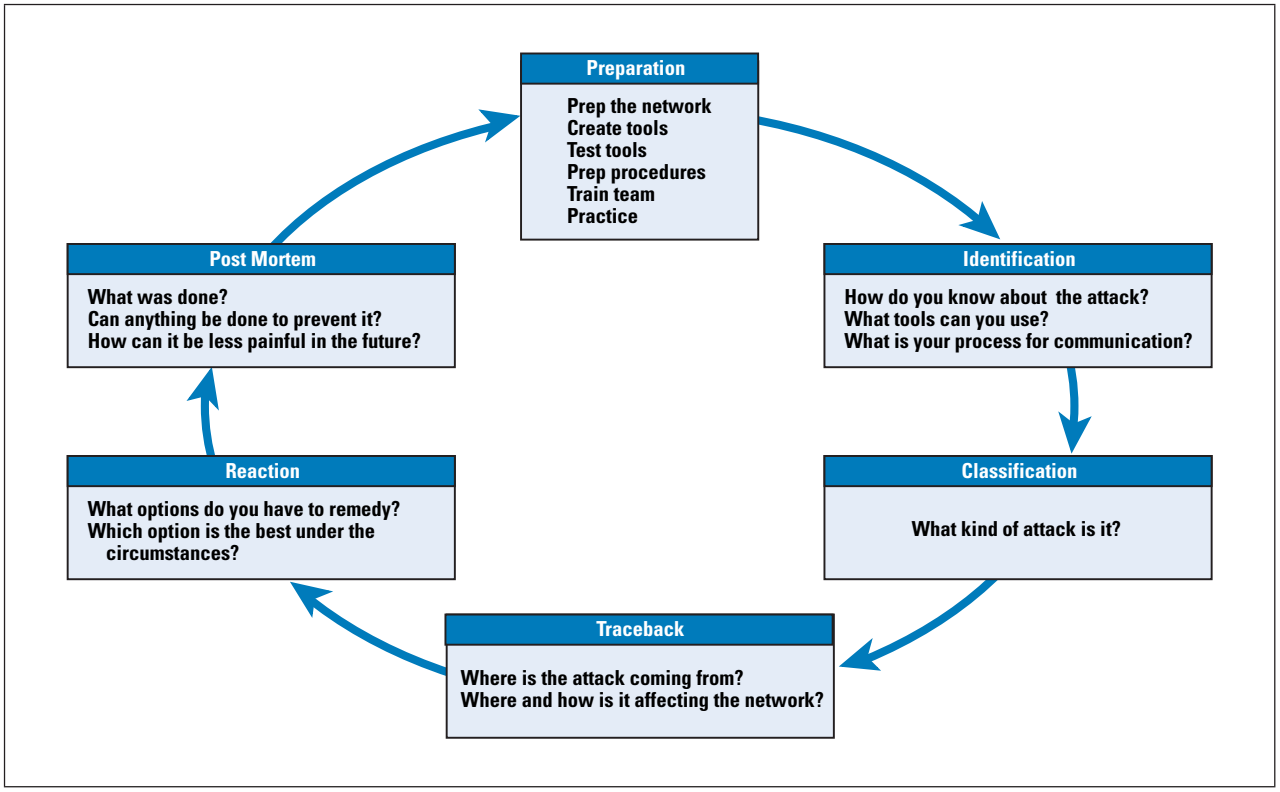
Figure 3: Managed IP VPN Security Services: IP MPLS, IPsec, or MPLS plus IPsec



Six-Step Methodology

Service providers can detect and mitigate attacks on the infrastructure using a six-step incident-response methodology (Figure 4).

Figure 4: Six Phases of Incident Response



- *Preparation:* The service provider needs to prepare the network, acquire the needed tools, develop and document a security plan, implement security procedures, and train NOC staff to use tools and procedures. *It is vital that security be a practice; the first time that the NOC staff follows its incident-response procedures should not be during an actual attack.*
- *Identification:* Unfortunately, service providers sometimes learn about a security incident from their customers. It is far better to be able to identify the threat before it becomes a problem, using NetFlow telemetry data and analysis tools, for example.
- *Classification:* The service provider needs to be able to quickly assess the nature of the threat and its scope: single customer, multiple customers, or entire infrastructure.
- *Traceback:* After classifying the threat, the IT staff needs to identify the point of ingress: peer, upstream server, downstream server, or compromised network device in the data center.
- *Reaction:* Following classification and traceback, the IT team applies the tools and processes needed to mitigate the attack. Success requires visibility into the network and well-defined procedures. Adherence to standard operating procedures helps prevent the service provider from inadvertently making the problem worse.
- *Post-mortem:* After the incident, the security team should analyze the root causes and integrate new insights into the security incident-handling procedures for use during the next incident.

Real-Life Observations About Interoperability from the APRICOT Workshops

Cisco and Juniper conducted a multivendor security workshop at APRICOT 2006 in Perth, Australia, and again at APRICOT 2007 in Bali, Indonesia. The workshops were offered in response to the fact that service providers often deploy a multivendor network for reasons ranging from financial to political.

Hands-on workshops were conducted in a lab using 12 routers running the Cisco IOS Software and another 12 running JUNOS software. Topics included:

- Password protection
- Packet filtering at the network edge
- Protecting the control plane
- Securing routing protocols
- Network monitoring techniques: NetFlow, syslog, SNMP, and *Network Time Protocol* (NTP)
- BGP MPLS Layer 3 VPNs
- IPsec VPNs

The goal of the workshops was to achieve a working configuration that interoperated with JUNOS and the Cisco IOS Software, resulting in consistent technology implementation, as well as common security policy enforcement. The workshops underscored the fact that interoperability is not automatic—even among standards-based network products. The reason is that standards bodies such as IETF, ITU, IEEE, and others define some aspects of protocols but leave others to vendor discretion. Standards do define *protocol format,* which is a syntactical structure identifying bit-field definition, length, and more. They also define *protocol behavior,* which specifies when actions occur, such as sending Hello and Keepalive timer probes and handling retransmission and reset packets. For purposes of analogy, a spoken language such as English is like a protocol format, and polite conversation conventions, such as beginning with a greeting and concluding with goodbye, is like a protocol behavior.

What standards do *not* cover are vendor-specific internal implementations, such as software coding techniques, hardware acceleration for performance, *command-line interface* (CLI) structure, and so on. Therefore, even though the APRICOT workshops involved deploying standards-based technology such as BGP-based MPLS VPNs and IPsec, vendor-specific differences had to be accounted for in the workshop materials and were noticed by participants. Following are examples noted at the APRICOT workshop:

- *Label Distribution Protocol:* With BGP MPLS VPN, JUNOS and Cisco IOS Software did not interoperate in their default configurations. However, routers from the same vendor did establish *Label Distribution Protocol* (LDP) sessions. The explanation, which participants found by troubleshooting with debug commands and referring to the manual, is that Cisco IOS Software uses the *Tag Distribution Protocol* (TDP) by default, whereas JUNOS uses LDP. After the Cisco IOS Software was changed to use LDP, the BGP-based MPLS VPN configuration succeeded.
- *IPsec tunnel establishment:* To simplify IPsec configuration, the workshop employed a *Graphical User Interface* (GUI) that prompted the user to choose source and destination IP addresses for the tunnel endpoints, a shared key, and the prefixes that defined the "interesting" traffic that was to use the IPsec tunnel. On the first attempt, the IPsec tunnel was not established. Workshop participants used the CLI to determine the problem, which was that the default encryption being negotiated was incompatible. The root cause for this mismatched encryption standard was that some routers were using an export version of software and needed an upgrade to support a higher encryption standard. Furthermore, even with common encryption capabilities, the two operating systems used different criteria to identify the interesting traffic that would be encrypted. Using the GUI, JUNOS defined interesting traffic as sourced from "ANY" network and destined to **`192.168.1.0/24`**.

In contrast, the Cisco IOS Software defined interesting traffic as sourced from **10.1.1.0/24** and destined to **192.168.1.0/24**. Following a discussion about whether the JUNOS default was too permissive or the Cisco IOS Software default was too restrictive, workshop participants agreed to disallow traffic that did not require encryption in the IPsec tunnel. The consensus was that the customer's security policy would provide a more conclusive answer to how permissive the policy should be, and that it was reasonable to require use of the CLI to tweak the configuration because the GUI performed most of the more difficult parts of the configuration on both platforms.

- *Loopback interface cost with Open Shortest Path First (OSPF):* During the OSPF deployment, participants noticed that the OSPF cost associated with interfaces was the same for each vendor. The OSPF cost is based upon a reference bandwidth of 100 Mbps. However, the loopback interfaces had different values: a default OSPF cost of 1 for the Cisco IOS Software and 0 for JUNOS. It is advisable to change one of the defaults to make them the same.

Although these subtle differences in protocols are documented by the vendors, service provider operational teams often have little time to research them. Therefore, it can be valuable for them to participate in multivendor hands-on workshops. Anecdotal evidence suggests that operators who are comfortable with multiple vendors understand the protocols, helping them design networks that can support new, revenue-generating services.

It is hoped that events such as the APRICOT workshops will help build a community of professionals who can add value for their employers, each other, and the broader Internet community. The result will be a secure and trusted networking environment that people and industry can rely on and use to connect in new and innovative ways.

Summary

Managed security services represent a growing revenue opportunity for service providers. Most service providers operate in a multivendor environment, either because of mergers and acquisitions or because their customers use other vendors' equipment. Therefore, a standards-based approach positions providers to capitalize on the managed security service opportunity. Providers can secure their infrastructure in a multivendor environment by following best practices for point protection, edge protection, RTBH protection, source-address validation, control-plane protection, and total visibility into network activity.

References

[1] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, November 1998.

[2] S. Kent and R. Atkinson, "IP Authentication Header," RFC 2402, November 1998.

[3] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)," RFC 2406, November 1998.

[4] E. Rosen and Y. Rekhter, "BGP/MPLS VPNs," RFC 2547, March 1999.

[5] S. Hanks, T. Li, D. Farinacci, and P. Traina, "Generic Routing Encapsulation (GRE)," RFC 1701, October 1994.

[6] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address Allocation for Private Internets," RFC 1918, February 1996.

[7] Internet Assigned Numbers Authority (IANA), "Special-Use IPv4 Addresses," RFC 3300, September 2002.

[8] F. Baker and P. Savola, "Ingress Filtering for Multihomed Networks," RFC 3704, March 2004.

[9] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)," RFC 2409, November 1998.

[10] Convery, S., "Network Authentication, Authorization, and Accounting – Part One: Concepts, Elements, and Approaches," *The Internet Protocol Journal,* Volume 10, No. 1, March 2007.

[11] Convery, S., "Network Authentication, Authorization, and Accounting – Part Two: Protocols, Applications, and the Future of AAA," *The Internet Protocol Journal,* Volume 10, No. 2, June 2007.

[12] Rigney et. al., "Remote Authentication Dial- In User Service (RADIUS)," RFC 2865 (Obsoletes RFC 2138, and RFC 2058), June 2000.

[13] Carrel et al., "The TACACS+ Protocol Version 1.78," Internet Draft, Work in Progress, **`draft-grant-tacacs-02.txt`**, January 1997.

[14] **`http://www.shrubbery.net/rancid/`**

[15] **`http://www.apricot.net`**

KUNJAL TRIVEDI joined Cisco in 1999 as a consulting engineer initially and then worked in product management covering Cisco IOS Software infrastructure security. Currently, he is helping Cisco shape a Managed Security Services marketing vision and strategy. A widely respected networking security expert, Kunjal presents infrastructure security, IP Security, and Managed Security topics at Cisco Networkers events as well as at conferences such as APRICOT. Kunjal has a Bachelor of Engineering degree with honors in electrical and electronics engineering from University of Wales, College of Cardiff, and a Master of Science degree in Artificial Intelligence from Cranfield Institute of Technology, UK. He holds CISSP and CCIE designations in routing and switching as well as security. Recently, he published a book titled *[Read Me First]: Building or Buying VPNs;* Kunjal has been awarded Chartered Engineer status by Institute of Engineering and Technology. He can be reached at **kunjal@cisco.com**

DAMIEN HOLLOWAY joined Juniper Networks in 2004 as an Instructing Engineer. He contributes to the development of the Juniper Technical Certification Program and custom delivery of training in the Asia Pacific region. Previously he was a consulting engineer and provided design, installation, and training to providers in Australia and the United States. Damien has presented a wide variety of topics relevant to customers, including backbone design, application acceleration, and *Broadband Remote Access Server* edge design, to audiences, including APRICOT and SANOG. Damien has a Bachelor of Electrical Engineering and Bachelor of Science from University of Sydney, Australia. He is a CCIE expert in routing and switching and JNCIE-M, JNCIP-E, and CISSP. He can be reached at **holloway@juniper.net**

Kunjal Trivedi (left) and Damien Holloway (center) share a joke with workshop students at APRICOT 2007

Kunjal with APRICOT 2007 workshop attendees

IPv4 Address Depletion and Transition to IPv6

by Geoff Huston, APNIC

At the recent APNIC meeting in New Delhi, the subject of IPv4, IPv6, and transition mechanisms was highlighted in the plenary session[1]. This article briefly summarizes that session and the underlying parameters in IPv4 address depletion and the transition to IPv6.

IPv4 Status

As of September 2007 we have some 18 percent of the unallocated IPv4 address pool remaining with the *Internet Assigned Numbers Authority* (IANA), and 68 percent has already been allocated to the *Regional Internet Registries* (RIRs) and through the RIRs to *Internet Service Providers* (ISPs) and end users. The remaining 14 percent of the IPv4 address space is reserved for private use, multicast, and special purposes. Another way of looking at this situation is that we have exhausted four-fifths of the unallocated address pool in IPv4, and one-fifth remains for future use. It has taken more than two decades of Internet growth to expend this initial four-fifths of the address space, so why shouldn't it take a further decade to consume what remains?

At this point the various predictive models come into play, because the history of the Internet has not been a uniformly steady model. The Internet began in the 1980s very quietly; the first round of explosive growth in demand was in the early 1990s as the Internet was adopted by the academic and research sector. At the time, the address architecture used a model where class A networks (or a /8) were extremely large, the class B networks (/16) were also too large, and the class C networks (/24) were too small for most campuses. The general use of class B address blocks was an uncomfortable compromise between consuming too much address space and consuming too many routing slots through address fragmentation. The subsequent shift to a classless address architecture in the early 1990s significantly reduced the levels of IPv4 address consumption for the next decade. However, over the past five years the demand levels for addresses have been accelerating again. Extensive mass-market broadband deployment, the demand for public non-*Network Address Translation* (NAT) addresses for applications such as *Voice over IP* (VoIP), and continuing real cost reductions in technology that has now brought the Internet to large populations in developing economies all contribute to an accelerating IPv4 address consumption rate.

Various approaches to modeling this address consumption predict that the IANA unallocated address pool will be fully depleted sometime in 2010 or 2011[2, 3, 4, 5].

Transitioning to IPv6

The obvious question is "What then?", and the commonly assumed answer to that question is one that the *Internet Engineering Task Force* (IETF) started developing almost 15 years ago, namely a shift to use a new version of the Internet Protocol: what we now know as IP Version 6, or IPv6. But if IPv6 really is the answer to this problem of IPv4 unallocated address-pool depletion, then we appear to be leaving the transition process quite late. The uptake of IPv6 in the public Internet remains extremely small as compared to IPv4[6]. If we really have to have IPv6 universally deployed by the time we fully exhaust the unallocated IPv4 address pools, then this objective appears to be unattainable during the 24 months we have to complete this work. The more likely scenario we face is that we will not have IPv6 fully deployed in the remaining time, implying a need to be more inventive about IPv4 in the coming years, as well as inspecting more closely the reason why IPv6 has failed to excite much reaction on the part of the industry to date.

We need to consider both IPv4 and IPv6 when looking at these problems with transition because of an underlying limitation in technology: *IPv6 is not "backward-compatible" with IPv4*. An IPv6 host cannot directly communicate with an IPv4 host. The IETF worked on ways to achieve this through intermediaries, such as a protocol to translate NATs[7], but this approach has recently been declared "historic" because of technical and operational difficulties[8]. That decision leaves few alternatives. If a host wants to talk to the IPv4 world, it cannot rely on clever protocol translating intermediaries somewhere, and it needs to have a local IPv4 protocol stack, a local IPv4 address, and a local IPv4 network and IPv4 transit. And to speak to IPv6 hosts, IPv6 has the same set of prerequisites as IPv4. This approach to transition through replication of the entire network protocol infrastructure is termed "Dual Stack." The corollary of Dual Stack is continued demand for IPv4 addresses to address the entire Internet for as long as this transition takes. The apparent contradiction here is that we do not appear to have sufficient IPv4 addresses in the unallocated address pools to sustain this Dual Stack approach to transition for the extended time periods that we anticipate this process to take.

What Can We Expect?

So we can expect that IPv4 addresses will continue to be in demand well beyond any anticipated date of exhaustion of the unallocated address pool, because in the Dual Stack transition environment all new and expanding network deployments need IPv4 service access and addresses. But the address distribution process will no longer be directly managed through address allocation policies after the allocation pool is exhausted.

Ideas that have been aired in address policy forums include encouraging NAT deployment in IPv4, expanding the private use of IPv4 address space to include the last remaining "reserved-for-future-use" address block, various policies relating to rationing the remaining IPv4 address space, increased efforts of address reclamation, the recognition of address transfers, and the use of markets to support address distribution.

Of course the questions here are about how long we need to continue to rely on IPv4, how such new forms of address distribution would affect existing notions of fairness and efficiency of use, and whether this effect would imply escalation of cost or some large-scale effect on the routing system.

On the other hand, is IPv6 really ready to assume the role of the underpinning of the global Internet? One view is that although the transition to a universal deployment of IPv6 is inevitable, numerous immediate concerns have impeded IPv6 adoption, including the lack of backward compatibility and the absence of simple, useful, and scalable translation or transition mechanisms[9]. So far the business case for IPv6 has not been compelling, and it appears to be far easier for ISPs and their customers to continue along the path of IPv4 and NATs.

When we contemplate this transition, we also need to be mindful of what we need to preserve across this transition, including the functions and integrity of the Internet as a service platform, the functions of existing applications, the viability of routing, the capability to sustain continued growth, and the integrity of the network infrastructure.

It appears that what could be useful right now is clear and coherent information about the situation and current choices, and analyzing the implications of various options. When looking at such concerns of significant change, we need to appreciate both the limitations and the strengths of the Internet as a global deregulated industry and we need, above all else, to preserve a single coherent networked outcome. Perhaps this topic is far broader than purely technical, and when we examine it from a perspective that embraces economic considerations, business imperatives, and public policy objectives, we need to understand the broader context in which these processes of change are progressing[10].

It is likely that some disruptive aspects of this transition will affect the entire industry, and this transition will probably be neither transparent nor costless.

References

[1] APNIC 24 Plenary Session: "The Future of IPv4," September 2007. **http://www.apnic.net/meetings/24/program/plenaries/apnic/**

[2] Geoff Huston, "The IPv4 Report." **http://ipv4.potaroo.net**

[3] Tony Hain, "IPv4 Address Pool." **http://www.tndh.net/~tony/ietf/ipv4-pool-combined-view.pdf**

[4] Tony Hain, "A Pragmatic Report on IPv4 Address Space Consumption," *The Internet Protocol Journal,* Vol. 8, No. 3, September 2005. **http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_8-3/ipv4.html**

[5] K.C. Claffy, CAIDA, " 'Apocalypse Then': IPv4 Address Space Depletion," Presentation to ARIN XVI, October 2005. **http://www.arin.net/meetings/minutes/ARIN_XVI/PDF/wednesday/claffy_ipv4_roundtable.pdf**

[6] Geoff Huston, "IPv6 / IPv4 Comparison Metrics." **http://bgp.potaroo.net/v6/v6rpt.html**

[7] G. Tsirtsis and P. Srisuresh, "Network Address Translation – Protocol Translation (NAT-PT)," RFC 2766, February 2000.

[8] C. Aoun and E. Davies, "Reasons to Move the Network Address Translator – Protocol Translator (NAT-PT) to Historic Status." RFC 4966, July 2007.

[9] Randy Bush, "IPv6 Operational Reality," APNIC 24 Plenary Presentation, September 2007. **http://www.apnic.net/meetings/24/program/plenaries/apnic/presentations/bush-ipv6-op-reality.pdf**

[10] Geoff Huston, "IPv4 Exhaustion," APNIC 24 Plenary Presentation, September 2007. **http://www.apnic.net/meetings/24/program/plenaries/apnic/presentations/huston-ipv4-exhaustion.pdf**

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. The author of numerous Internet-related books, he is currently the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of the Internet Society from 1992 until 2001. E-mail: **gih@apnic.net**

IPv4 Address Space: 2.46 Billion Down, 1.25 Billion to Go

by Iljitsch van Beijnum

In September 2005, *The Internet Protocol Journal* published an article about the IPv4 address space consumption[1]. At that time, projections done by Geoff Huston and Tony Hain varied widely, because the number of /8 address blocks in use had gone up sharply in early 2005. So what has happened since then, and what can we expect for the not-too-distant future?

Address Assignment and Allocation

The *Internet Assigned Numbers Authority* (IANA, part of the *Internet Corporation for Assigned Names and Numbers* [ICANN]) has authority over the IPv4 address space. In the past, IANA gave out address blocks directly to end users, but now IANA distributes address space in the form of /8 blocks, each holding 24 bits worth of address space, or 16,777,216 addresses, to five *Regional Internet Registries* (RIRs). There are a few exceptions, but AfriNIC[2] gives out address space in Africa; APNIC[3] in the Asia-Pacific region; ARIN[4] in North America; LACNIC[5] in Latin America and the Caribbean; and RIPE NCC[6] in Europe, the former Soviet Union, and the Middle East. These RIRs sometimes assign address space to end users, but mostly allocate it to *Internet Service Providers* (ISPs), who then assign it to their customers, meaning that there are two pools of available address space: the global pool of /8 blocks that IANA has not delegated to anyone[7], and the address space held by the RIRs that they have not given out yet. The article in the September 2005 issue of *The Internet Protocol Journal*[1] looked at the depletion of the IANA global pool, whereas this article mostly looks at the amounts of address space given out by the RIRs, providing a more granular view. The RIRs publish daily reports of their address assignments and allocations on their respective FTP servers. According to these reports as downloaded on January 1, 2007, the amounts of address space shown in Table 1 were given out over the past seven years.

Table 1: Address Space Allocated 2000–2006 [January 2007 data]

	2000	2001	2002	2003	2004	2005	2006
AfriNIC	0.56	0.39	0.26	0.22	0.51	1.03	2.72
APNIC	20.94	28.83	27.03	33.05	42.89	53.86	51.78
ARIN	30.83	28.55	21.08	22.32	34.26	47.57	38.94
LACNIC	0.88	1.61	0.65	2.62	3.77	10.97	11.50
RIPE NCC	24.79	25.36	19.84	29.61	47.49	62.09	56.53
Total	78.00	84.73	68.87	87.82	128.92	175.52	161.48

However, if we compare these totals to the totals seen on January 1, 2006, we see some differences (Table 2).

Table 2: Address Space Allocated 2000–2006 [January 2006 data]

	2000	2001	2002	2003	2004	2005	2006
Total	78.35	88.95	68.93	87.77	128.45	165.45	–

For the years 2000 to 2002, the number of addresses registered as given out is slightly lower, as seen in the January 1, 2007 data compared to the January 1, 2006 data—a result that is to be expected because address space given out in that year that is no longer used is returned. However, for the later years, and especially for 2005, there is a retroactive *increase* in the number of addresses given out. The reason: When ARIN suspects an address space user may come back for more space relatively soon, it takes a larger block than requested, and then fulfills the request from part of that block and keeps the rest in reserve. So an organization requesting a /16 may get the first half of a /15. When that organization then requests another /16 one or two years later, ARIN gives the organization the second half of the /15. ARIN subsequently records this as a /15 given out on the date when the original /16 was requested.

For instance, ARIN's January 1, 2006, data shows that a block of 12.6 million addresses was given out within **`73.0.0.0/8`** block:

```
arin|US|ipv4|73.0.0.0|12582912|20050419|allocated
```

In the January 1, 2007, data, this number had changed to 13.6 million addresses:

```
arin|US|ipv4|73.0.0.0|13631488|20050419|allocated
```

This change means that simply looking at the registration date does not provide very good information. It also does not account for address space given out in earlier years that is returned. An alternative approach is to count the amount of address space given out based on the RIR records published on a certain date (Table 3).

Table 3: RIR Records for Address Space Allocation

	IANA (/8)		RIRs (millions)			Total	
	Delegated	Free	Received	Delegated	Free	Free	Delta
Jan. 1, 2004	133	88	1509.95	1245.63	264.32	1740.71	
Jan. 1, 2005	142	79	1660.95	1351.66	309.30	1634.69	106.02
Jan. 1, 2006	155	66	1879.05	1517.74	361.31	1468.61	166.08
Jan. 1, 2007	166	55	2063.60	1685.69	377.90	1300.65	167.96
May 1, 2007	172	49	2181.04	1754.68	426.36	1248.44	52.21

(Note that block **`7.0.0.0/8`** shows up as unused in the IANA global pool and is counted as available in the table, but this block is in fact used by the U.S. Department of Defense.)

The jump in address consumption between 2004 (106 million) and 2005 (166 million) is even more dramatic in this light, while consumption numbers of 2005 and 2006 (168 million) are now almost identical. The figure for the first four months of 2007 seems rather modest at 52 million addresses, but the reason lies in the fact that Bolt, Beranek and Newman returned **`46.0.0.0/8`** to IANA in April. So the number of addresses given out from January to April was 69 million, a rate that puts the RIRs on track to give out more than 200 million addresses in 2007.

The size of address blocks given has been increasing steadily. Table 4 shows the number of requests for a certain range of block sizes: equal or higher than the first, lower than the second value (2005 and earlier values from the January 1, 2006 data, 2006 values from the January 1, 2007 data).

Table 4: Number of Requests for Ranges of Block Sizes

	2000	2001	2002	2003	2004	2005	2006
< 1,000	326	474	547	745	1022	1309	1526
1,000 – 8,000	652	1176	897	1009	1516	1891	2338
8,000 – 64k	1440	868	822	1014	1100	1039	1133
64k – 500k	354	262	163	215	404	309	409
500k – 2M	19	39	29	46	61	60	56
> 2M	3	5	5	6	7	18	13

The number of blocks in the two smallest categories has increased rapidly, but not as fast as the number of blocks in the largest category, in relative numbers. However, the increase in large blocks has a very dramatic effect whereas the small blocks are insignificant, when looking at the millions of addresses involved (Table 5).

Table 5: Millions of Addresses Given Out

	2000	2001	2002	2003	2004	2005	2006
< 1,000	0.10	0.16	0.18	0.25	0.35	0.44	0.52
1,000 – 8,000	2.42	4.47	3.23	3.45	4.49	5.07	6.10
8,000 – 64k	18.79	12.81	11.35	14.00	15.99	15.46	17.17
64k – 500k	35.98	32.19	20.28	25.51	42.01	34.23	49.64
500k – 2M	12.68	24.64	21.30	31.98	44.63	41.63	46.64
> 2M	8.39	14.68	12.58	12.58	20.97	68.62	41.42

The increase in the 2M+ blocks was solely responsible for the high number of addresses given out in 2005. In 2006, there was growth in all categories except the 2M+ one (even the 500k – 2M category increased in number of addresses if not in number of blocks). When the 2M+ blocks are taken out of the equation, 2005 had a total of 96.83 million addresses (January 1, 2006) and 2006 had 119.06 million given out, even without fully correcting for the ARIN reporting particularities. Apparently there is still an underlying upward trend.

Figure 1 shows the amounts of address space given out by IANA and by the RIRs every year from 1994 to 2006.

Figure 1: IPv4 Address Space Given Out from 1994 to 2006

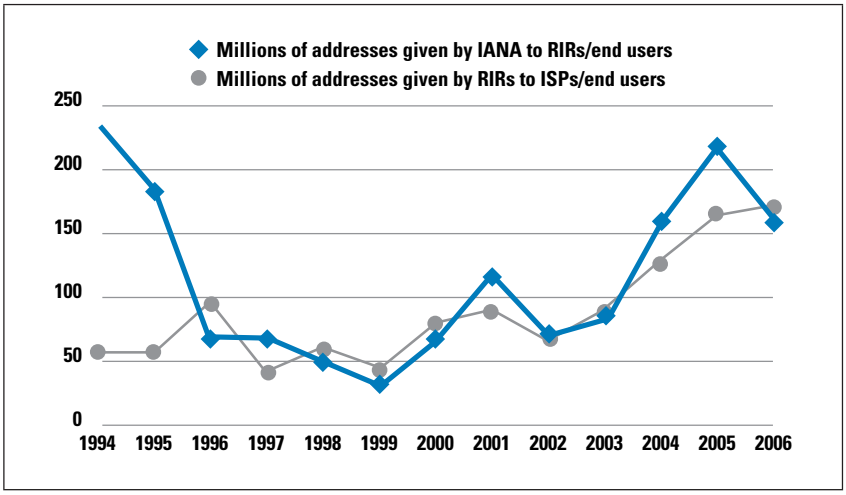
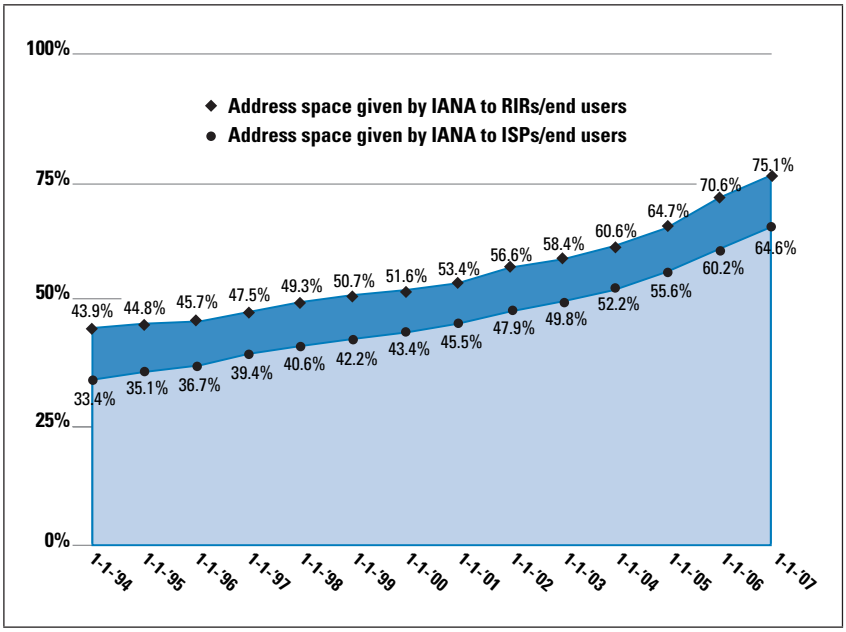


Figure 2 shows the amounts of address space marked as "in use" by IANA and by the RIRs. The difference between the two numbers is what the RIRs hold in order to satisfy day-to-day address space requests. This amount is usually two years' worth of address space.

Figure 2: IPv4 Address Space in Use from 1994 to 2006



Depletion

The exact moment when the IPv4 address space will be depleted depends on numerous factors. Since 1997, the three-year period with the largest growth in yearly address use was from 2003 to 2005 relative to the 2002 figure: a factor 2.4, or 34 percent per year. If this growth repeats itself in the next three years, we will be out of IPv4 addresses in the second half of 2010.

Interestingly, the period with the lowest growth also includes the year 2003: from 2001 to 2003 relative to 2000. In 2003, 12 percent more addresses were given out than in 2000, for an average increase in yearly use of 4 percent. If this is the new trend for the coming years, we can expect to run out of IPv4 addresses in mid-2013. There is of course no reason to assume that future IP address use will conform to patterns seen in earlier years, but we really have nothing else to base our projections upon.

So anyone expecting to obtain new IPv4 address space more than three years from now is taking a big risk. With the IPv4 reserves visibly diminishing each year, the question is: What can we, as a community, do to make the IPv4 address depletion as painless as possible? IPv4 addresses are useful only if the people who need them can obtain them, meaning that using up addresses unnecessarily fast or locking up the still-available reserves are both suboptimal solutions. It has been suggested that turning IPv4 address space into a tradable commodity would allow a free market to form, aiding the efficient distribution of address space from those who have it to those who need it.

This scenario has several problems. First, when supply is limited and demand is high, prices rise and hoarding becomes lucrative. So the effect of making address space tradable could be a reduction of available address space rather than an increase. And certainly, as trading IPv4 space becomes more likely, holders of large address blocks will be less inclined to return them. Finally, more than half of the IPv4 address space in use is held by organizations in the United States, whereas the developing world has comparatively little address space. The prospect of having to buy address space from American companies that got the space for free is not likely to be popular in the rest of the world.

Address Reclamation a Solution?

There are two large classes of potentially reclaimable address space: the class E reserved space (**`240.0.0.0 – 255.255.255.255`**) and the class A blocks given out directly to end users by IANA. The class E space has 268 million addresses and would give us in the order of 18 months worth of IPv4 address use. However, many TCP/IP stacks, such as the one in Windows, do not accept addresses from class E space and will not even communicate with correspondents holding those addresses. It is probably too late now to change this behavior on the installed base before the address space would be needed. There are currently 42 class A blocks and another two /8s from class C space listed as given out to end users—738 million addresses. The U.S. government uses about 10 of those blocks; 21 of them are not present in the *Border Gateway Protocol* (BGP) routing table.

Although harsh judgments about the need for so much address space are easily made from the outside without having all the pertinent information, it seems reasonable to try to reclaim some of this space. I would consider getting back half of this space a big success, but that would give us only 2 years worth of additional address space. There are also 645 million addresses of older class B assignments, but reclaiming those will be extremely difficult because nearly 8,000 individual assignments are involved. Reclaiming a class B block is probably not much easier than reclaiming a class A block, but the amount of address space returned is less than half a percent.

Planning for the End Game

So what should we do? In my opinion: promote predictability. The situation where we run out of IPv4 address space much faster than expected would be very harmful as organizations struggle to adjust to the new circumstances. On the other hand, if the IPv4 space unexpectedly lasts longer, people may be disinclined to believe space is really running out and then would be unprepared when it does. Artificially delaying running out of IPv4 address space also prolongs the situation in which it is difficult to get IPv4 space, but not enough people feel the pain to initiate IPv6 deployment. One solution worthy of consideration would be to impose a worldwide moratorium on the change of IPv4 address allocation and assignment policies after a certain date to aid this predictability. If some kind of encouraged or forced reclamation of older class A blocks is desired, this process should be instigated sooner rather than later, both for the sake of predictability and because it gives the address holders involved time to reorganize their networks. Another small but useful step would be to limit the size of address blocks given out. This scenario would be like the agreement between the RIRs and IANA that the RIRs will receive two /8s at a time in the future. The situation where a single /9 or /8 allocation constitutes 5 or even 10 percent of the address space given out in that year makes adequate predictions extremely difficult, and also runs the risk that a good part of the address block in question will never be used as circumstances change. Limiting individual allocations to a /11 or /12 would be better, even if it requires the requesting organization to come back for more address space several times per year.

Finally, it seems prudent for all organizations using public IPv4 address space to start planning for the moment that they themselves, or third parties that they communicate with over the public Internet, can no longer obtain additional IPv4 address space.

References

[1] Tony Hain, "A Pragmatic Report on IPv4 Address Space Consumption," *The Internet Protocol Journal,* Volume 8, No. 3, September 2005.

[2] AfriNIC, **http://www.afrinic.net**

[3] APNIC, **http://www.apnic.net**

[4] ARIN, **http://www.arin.net**

[5] LACNIC, **http://www/lacnic.net**

[6] RIPE NCC, **http://ripe.net**

[7] IANA Internet Protocol v4 Address Space **http://www.iana.org/assignments/ipv4-address-space**

ILJITSCH VAN BEIJNUM holds a Bachelor of Information and Communication Technology degree from the Haagse Hogeschool in The Hague, Netherlands. In 1995, he found himself in the emerging Internet Service Provider business. There he learned about system administration, IP networking, and especially routing. After first starting a small ISP with four others and working as a senior network engineer for UUNET Netherlands, he became a freelance consultant in 2000. Not long after that, he started contributing to the IETF Multihoming in IPv6 working group. He wrote the book *BGP: Building Reliable Networks with the Border Gateway Protocol,* ISBN 0-596-00254-8, published by O'Reilly in 2002, and *Running IPv6,* ISBN 1590595270, published by Apress in 2005. E-mail: **iljitsch@muada.com**

Used but Unallocated: Potentially Awkward /8 Assignments

by Leo Vegoda, ICANN

IPv4 has proven to be exceedingly popular, so it should be no surprise that the time is rapidly approaching when the last /8 block will be allocated and the *Internet Assigned Numbers Authority's* (IANA's) free pool of address space will be empty. At the time of writing, Geoff Huston of the *Asia Pacific Network Information Centre* (APNIC) is projecting[1] the IANA free pool will run out in mid-2010. Unfortunately, it is possible that some of these remaining /8s may cause problems for enterprise and *Internet Service Provider* (ISP) network operators when they are put back into use. These blocks are not the /8s that have been returned to IANA by the original registrants; they are previously unassigned address blocks.

Concerns

There are many concerns about the IANA free pool depletion, but one of them seems particularly straightforward to identify and fix. Many organizations have chosen to use unregistered IPv4 addresses in their internal networks and, in some cases, network equipment or software providers have chosen to use unregistered IPv4 addresses in their products or services. In many cases the choice to use these addresses was made because the network operators did not want the administrative burden of requesting a registered block of addresses from a *Regional Internet Registry* (RIR)[2, 11]. In other cases they may not have realized that RFC 1918[3] set aside three blocks of address space for private networks, so they just picked what they believed to be an unused block, or their needs exceeded the RFC 1918 set-aside blocks. Other organizations used the default address range suggested by their equipment vendor, or supplied in example documentation, when configuring *Network Address Translation* (NAT) devices. Regardless of the reason, these uses of unregistered addresses will conflict with routed addresses when the /8s in question are eventually assigned to ISPs or enterprise users.

A few examples of /8s where problems are likely to occur follow:

`1.0.0.0/8` Widely used as private address space in large organizations whose needs exceed those provided for by RFC 1918[4]

`5.0.0.0/8` Used by one of numerous zero-configuration Internet applications (including the Hamachi VPN service [5, 6])

`42.0.0.0/8` Default range used in the NAT configuration of at least one Internet appliance (the HP Procurve 700wl[7])

Organizations using these address ranges in products or services may experience problems when more specific Internet routes attract traffic that was meant for internal hosts, or alternatively find themselves unable to reach the legitimate users of those addresses because those addresses are being used internally. The users of unregistered networks may also find problems with reverse *Domain Name System* (DNS) resolution, depending on how their DNS servers are configured. These problems are likely to result in additional calls to helpdesks and security desks at both enterprises and ISPs, with unexpected behavior for end users that might be hard to diagnose. Users of unregistered address space may also experience problems with unexpected traffic being received at their site if they leak internal routes to the public Internet. Many ISPs have already had experience with this type of routing inconsistency as recent /8 allocations reach routing tables and bogon filters are updated.

Alternatives

There are several alternatives to using unregistered IPv4 address space:

- Use RFC 1918 IPv4 address space (no need to obtain this space from an RIR)
- Use IPv4 address space registered with an RIR
- Use IPv6 address space registered with an RIR
- Use IPv6 Unique Local Address[8] space (no need to obtain this space from an RIR)

Obviously, all of these efforts will involve renumbering networks, a sometimes painful and time-consuming process. Those using unregistered unique IPv4 address space should look at renumbering their networks or services before the previously unallocated /8s are allocated to avoid address clashes and routing difficulties.

Additionally, vendors and documentation writers can clean up their configurations to ensure they use RFC 1918 addresses, or make it clear to their users that they must use registered addresses to avoid routing conflicts.

All RIRs provide free telephone helpdesks that can advise you on obtaining unique IPv4 or IPv6 address space. But if you want to continue using unregistered space and can transition to IPv6, the prefix selection mechanism described in RFC 4193 makes the probability of a clash a mere 1 in 550 billion. Ultimately, transitioning to IPv6 is most likely the best solution, and this approach offers an opportunity for those having to renumber parts of their network to avoid a subsequent renumbering later into IPv6.

About IANA and ICANN

IANA allocates address space to RIRs according to the global IPv4 [9] and IPv6[10] policies. Enterprise and ISP networks need to obtain IP addresses from their upstream provider or from the appropriate RIR.

The *Internet Corporation for Assigned Names and Numbers* (ICANN) is an internationally organized, nonprofit corporation that has responsibility for *Internet Protocol* (IP) address space allocation, protocol identifier assignment, *generic* (gTLD) and *country code* (ccTLD) *Top-Level Domain* name system management, and root server system management functions. These services were originally performed under U.S. government contract by IANA and other entities. ICANN now performs the IANA function.

References

[1] **http://www.potaroo.net/tools/ipv4/**

[2] RFC 1174, para 2.2 states in part, "The term Internet Registry (IR) refers to the organization which has the responsibility for gathering and registering information about networks to which identifiers (network numbers, autonomous system numbers) have been assigned by the IR. An RIR does this function for its service area."

[3] **http://www.ietf.org/rfc/rfc1918.txt**

[4] **http://tools.ietf.org/id/draft-hain-1918bis-01.txt**

[5] **https://secure.logmein.com/products/hamachi/howitworks.asp**

[6] **http://en.wikipedia.org/wiki/Hamachi**

[7] **http://www.hp.com/rnd/support/faqs/700wl.htm**

[8] **http://www.ietf.org/rfc/rfc4193.txt**

[9] **http://www.icann.org/general/allocation-IPv4-rirs.html**

[10] **http://www.icann.org/general/allocation-IPv6-rirs.htm**

[11] Daniel Karrenberg, Gerard Ross, Paul Wilson, and Leslie Nobile, "Development of the Regional Internet Registry System," *The Internet Protocol Journal,* Volume 4, No. 4, December 2001.

LEO VEGODA holds a BA (Hons) from the University of Central England. He joined ICANN in 2006 and is the Manager, Number Resources - IANA. He has previously worked for the RIPE NCC, where he ran the Registration Services department. He can be reached at: **leo.vegoda@icann.org**

Book Review

Uncommon Sense

Uncommon Sense: Out of the Box Thinking for An In the Box World, By Peter Cochrane, ISBN 1-84112-477-x, Published by Capstone, 2004, **http://www.wileyeurope.com**

A series of articles published in **silicon.com** form the basis for this book, which looks at the effect that new technology has on business and its implications for society. In many ways it attacks conventional wisdom and forces a reevaluation of the effect of technology, often exposing flaws in the business logic that lead to many investments and decisions.

The book is aimed at technologists, managers, and professionals who are interested in change and progress, offering them a glimpse of the future. It is easy to read, with liberal use of figures and tables to aid understanding.

Organisation

Cochrane begins by looking at the communication of ideas, particularly fairly complex and novel concepts. He notes the lack of agreement on the major concerns of the future and bemoans the handling of complex business and political topics–and the lack of engineering type rigour applied to their assessment. He suggests a much more rigorous modeling of complex business problems is required, especially of business processes, which are typically complex and inter-related, so treating them as isolated "stovepipes" is inappropriate and error-prone. Cochrane emphasises the need for nonlinear thinking.

Cochrane's analysis continues with an assessment of technology markets, not surprisingly beginning with the forces behind the dot-com bubble, with particular reference to the effect that the so-called new and old economies have had on each other. He suggests that short-term approaches, with their tendency to hit high-visibility symptoms and not the underlying commercial factors, are a barrier to progress. Cochrane reflects that whilst the dot-com boom is over, it is now clear that the online world has been very successful and has dragged the old world along in its wake.

The book then looks at change: considering the adoption of new technology and the impact effect of the Internet, comparing this new technology with the adoption of television. Cochrane spends a significant amount of time on both entertainment and learning. He examines topics as varied as security, the ease of movement of information across borders, and the role of specialist and general devices.

His assessment of security considers the range and rate of spread of threats and some advanced countermeasures such as biometrics. He considers the nature of change programmes and the harmful ways insensitive micromanagement can affect their progress.

Cochrane explores the role of the consumer in deciding which technical innovations survive, as exemplified by the growth of the American cable TV (CATV) market. He notes that most consumers have a fixed level of disposable income and new innovations allow them to redirect rather than increase their level of spending. Cochrane argues that this truth is reflected in the saturation within the mobile handset market and the dynamics seen between the media companies and new innovators such as Napster.

The penultimate collection of essays considers the speed of innovation. Cochrane notes that many consumers are suffering from "technology fatigue" and many products are suffering from "feature death." Here he discusses stagnation within the mobile market and disillusionment with the *Wireless Application Protocol* (WAP), *General Packet Radio Service* (GPRS), and Bluetooth. He notes that the adoption of technology is linked to the willingness of customers to pay.

Cochrane concludes by looking at leading-edge variables, including reliability, noting that this variable goes hand-in-hand with maturity, with the *Public Switched Telephone Network* (PSTN) delivering extremely high levels of reliability and most modern IT solutions delivering considerably less. He makes this comparison a critical test of the five-nines availability claims of many new technology solutions. Cochrane looks at some more less-conventional ideas such as the replication of ant logic in IT systems and the possible future use of plasma screens and voice recognition as convenient input/output devices. He notes the increasing intelligence of devices, but also acknowledges that rapid communications and minimal hierarchy can triumph over better organised structures as demonstrated by protesters in France in 2000 and 2001.

Synopsis

Cochrane takes the reader through many contemporary technology developments and concerns and in the process invites his readers to form their own views. His mission is to "communicate the implications of what we have done, are doing and are about to do." In 50 short articles, delivered in 233 pages, it is possible for the author to cover only a small portion of a rapidly growing field, providing sufficient detail to appeal to the technologist without losing the bigger picture. He examines the implications of new technology for society and notes that the progress we are seeing means that we have to take on the new, changing the way we manage, operate, and govern our businesses as a result.

The Author

Peter Cochrane is the ex-BT Chief Technologist, who with a group of ex-Apple Computer technologists founded Concept Labs, where he advises a range of companies across the world. He has published widely, holds B.Sc., M.Sc., Ph.D., and D.Sc. degrees from Nottingham (Trent) and Essex Universities, is an Apple Master, and is a visiting professor at London, Essex, and Southampton Universities. He is best known for his incisive and often provocative views on the United Kingdom and world telecommunications industries.

—Edward Smith, BT, UK
edward.a.smith@btinternet.com

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

December 2007 Volume 10, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

Identity theft is a widely reported problem in today's world. Criminals can use numerous ways to obtain private information such as Social Security Numbers, credit card details, and other information that makes it possible for the perpetrator to successfully "pretend to be" someone else. A similar concept, albeit less personal, is so-called *IP Spoofing*, wherein fake IP datagrams can be generated and sent across the network in order to compromise remote systems in a variety of ways. Farha Ali gives an overview of IP Spoofing and explains ways in which the problem can be mitigated.

Our second article looks at numerous standards for information security management being developed by organizations such as the *International Organization for Standardization* (ISO), the *National Institute of Standards and Technology* (NIST), and others. The author of the article is William Stallings.

On November 2, 2007, Vint Cerf ended his term as chairman of the *Internet Corporation for Assigned Names and Numbers* (ICANN). At the same time he released a document entitled "Looking Toward the Future," which details ICANN's history, as well as outlining its challenges ahead. We've included the document in this issue and added some pointers for those readers who may not be familiar with the workings of ICANN.

In late October, the Internet technical community received the sad news that Dr. Junichiro Hagino, universally known as "Itojun" had passed away. Itojun played a very important role in the development of IPv6 and had many friends across the world. We asked one of them, Bob Hinden, to reflect on Itojun's life and compile some comments from those who knew him well.

We would like to remind you about our online adjunct to this journal. *The Internet Protocol Forum* (IPF) available at **http://www.ipjforum.org/** is a resource you can use to discuss articles and read additional material. Please take a moment to explore IPF.

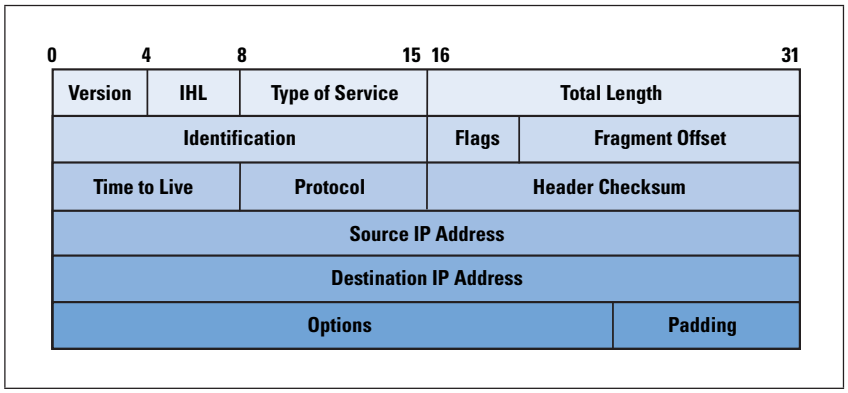
—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

IP Spoofing

by Farha Ali, Lander University

The *Internet Protocol*, or IP, is the main protocol used to route information across the Internet. The role of IP is to provide best-effort services for the delivery of information to its destination. IP depends on upper-level TCP/IP suite layers to provide accountability and reliability. The heart of IP is the IP *datagram*, a packet sent over the Internet in a connectionless manner. An IP datagram carries enough information about the network to get forwarded to its destination; it consists of a *header* followed by bytes of *data*. The header contains information about the type of IP datagram, how long the datagram should stay on the network (or how many hops it should be forwarded to), special flags indicating any special purpose the datagram is supposed to serve, the destination and source addresses, and several other fields, as shown in Figure 1.

Figure 1: The IP Header



Layers above IP use the source address in an incoming packet to identify the sender. To communicate with the sender, the receiving station sends a reply by using the source address in the datagram. Because IP makes no effort to validate whether the source address in the packet generated by a node is actually the source address of the node, you can spoof the source address and the receiver will think the packet is coming from that spoofed address. Many programs for preparing spoofed IP datagrams are available for free on the Internet; for example, *hping* lets you prepare spoofed IP datagrams with just a one-line command, and you can send them to almost anybody in the world. You can spoof at various network layers; for example, you can use *Address Resolution Protocol* (ARP) spoofing to divert the traffic intended for one station to someone else. The *Simple Mail Transfer Protocol* (SMTP) is also a target for spoofing; because SMTP does not verify the sender's address, you can send any e-mail to anybody pretending to be someone else. This article focuses on the various types of attacks that involve IP spoofing on networks, and the techniques and approaches that experts in the field suggest to contend with this problem.

Spoofing IP datagrams is a well-known problem that has been addressed in various research papers. Most spoofing is done for illegitimate purposes—attackers usually want to hide their own identity and somehow damage the IP packet destination. This article discusses ways of spoofing IP datagrams, various attacks that involve spoofed IP packets, and techniques to detect spoofed packets and trace them back to their original source; spoofing concerns for IPv6 are briefly addressed.

Spoofing an IP Datagram

IP packets are used in applications that use the Internet as their communications medium. Usually they are generated automatically for the user, behind the scenes; the user just sees the information exchange in the application. These IP packets have the proper source and destination addresses for reliable exchange of data between two applications. The IP stack in the operating system takes care of the header for the IP datagram. However, you can override this function by inserting a custom header and informing the operating system that the packet does not need any headers. You can use raw sockets in UNIX-like systems to send spoofed IP datagrams, and you can use packet drivers such as *WinPcap* on Windows. Some socket programming knowledge is enough to write a program for generating crafted IP packets. You can insert any kind of header, so, for example, you can also create *Transmission Control Protocol* (TCP) headers. If you do not want to program or have no knowledge of programming, you can use tools such as *hping, sendip,* and others that are available for free on the Internet, with very detailed documentation to craft any kind of packet. Most of the time, you can send a spoofed address IP packet with just a one-line command.

Why Spoof the IP Source Address?

What is the advantage of sending a spoofed packet? It is that the sender has some kind of malicious intention and does not want to be identified. You can use the source address in the header of an IP datagram to trace the sender's location. Most systems keep logs of Internet activity, so if attackers want to hide their identity, they need to change the source address. The host receiving the spoofed packet responds to the spoofed address, so the attacker receives no reply back from the victim host. But if the spoofed address belongs to a host on the same subnet as the attacker, then the attacker can "sniff" the reply. You can use IP spoofing for several purposes; for some scenarios an attacker might want to inspect the response from the target victim (called "nonblind spoofing"), whereas in other cases the attacker might not care (blind spoofing). Following is a discussion about reasons to spoof an IP packet.

Scanning

An attacker generally wants to connect to a host to gather information about open ports, operating systems, or applications on the host. The replies from the victim host can help the attacker in gathering information about the system.

These replies might indicate open ports, the operating system, or several applications running on open ports. For example, a response for connection at port 80 indicates the host might be running a Web server. The hacker can then try to *telnet* to this port to see the banner and determine the Web server version and type, and then try to exploit any vulnerability associated with that Web server. In the scanning case, attackers want to examine the replies coming back from the host, so they need to see the returned packet. If the spoofed address is actually an address of a host on the attacker's subnet, then the attacker can use a sniffer to see the packets.

Sequence-Number Prediction

If you establish the connection between two hosts by using TCP, the packets exchanged between the two parties carry sequence numbers for data and acknowledgments. The protocol uses these numbers to determine out-of-order and lost packets, thus ensuring the reliable delivery to the application layer as promised by TCP. These numbers are generated pseudo-randomly in a manner known to both the parties. An attacker might send several spoofed packets to a victim to determine the algorithm generating the sequence numbers and then use that knowledge to intercept an existing session. Again it is important for the attacker to be able to see the replies.

Hijacking an Authorized Session

An attacker who can generate correct sequence numbers can send a reset message to one party in a session informing that party that the session has ended. After taking one of the parties offline, the attacker can use the IP address of that party to connect to the party still online and perform a malicious act on it. The attacker can thus use a trusted communication link to exploit any system vulnerability. Keep in mind that the party that is still online will send the replies back to the legitimate host, which can send a reset to it indicating the invalid session, but by that time the attacker might have already performed the intended actions. Such actions can range from sniffing a packet to presenting a shell from the online host to the attacker's machine.

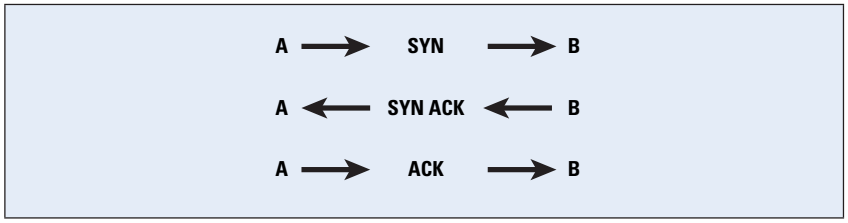
Determining the State of a Firewall

A firewall is used to protect a network from Internet intruders. Packets entering a firewall are checked against an *Access Control List* (ACL). TCP packets sent by a source are acknowledged by acknowledgment packets. If a packet seems like an acknowledgement to a request or data from the local network, then a stateful firewall also checks whether a request for which this packet is carrying the acknowledgment was sent from the network. If there is no such request, the packet is dropped, but a stateless firewall lets packets enter the network if they seem to carry an acknowledgment for a packet. Most probably the intended receiver sends some kind of response back to the spoofed address. Again, for this process to work, the attacker should be able to see the traffic returning to the host that has the spoofed address—and the attacker generally knows how to use the returned packet to advantage.

Denial of Service

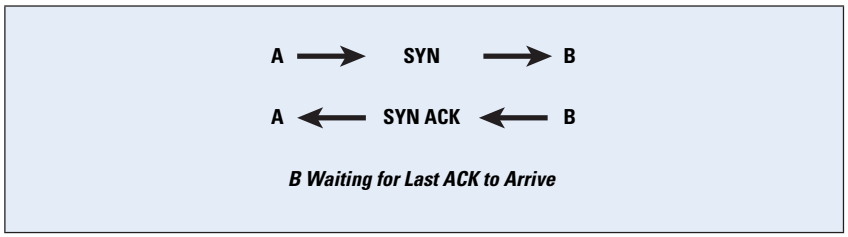
The connection setup phase in a TCP system consists of a *three-way handshake*. This handshake is done by using special bit combinations in the "flags" fields. If host A wants to establish a TCP connection with host B, it sends a packet with a SYN flag set. Host B replies with a packet that has SYN and ACK flags set in the TCP header. Host A sends back a packet with an ACK flag set, finishing the initial handshake. Then hosts A and B can communicate with each other, as shown in Figure 2.

Figure 2: A Normal TCP Connection Request from A to B



The three-way handshake must be completed in order to establish a connection. Connections that have been initiated but not finished are called *half-open connections*. A finite-size data structure is used to store the state of the half-open connections. An attacking host can send an initial SYN packet with a spoofed IP address, and then the victim sends the SYN-ACK packet and waits for a final ACK to complete the handshake. If the spoofed address does not belong to a host, then this connection stays in the half-open state indefinitely, thus occupying the data structure. If there are enough half-open connections to fill the state data structure, then the host cannot accept further requests, thus denying service to the legitimate connections (Figure 3).

Figure 3: Half-Open TCP Connection



Setting a time limit for half-open connections and then erasing them after the timeout can help with this problem, but the attacker may keep continuously sending the packets. The attacked host will not have space to accept new incoming legitimate connections, but the connection that was established before the attack will have no effect. In this type of attack, the attacker has no interest in examining the responses from the victim. When the spoofed address does belong to a connected host, that host sends a reset to indicate the end of the handshake.

Flooding

In this type of attack an attacker sends a packet with the source address of the victim to multiple hosts. Responses from other machines flood the victim. For example, if an attacker uses the IP address of source A and sends a broadcast message to all the hosts in the network, then all of them will send a reply back to A, hence flooding it. The well-known *Smurf* and *fraggle* attacks used this technique.

Countermeasures for IP Spoofing

IP spoofing countermeasures include detecting spoofed IP packets and then tracing them back to the originating source. Detection of spoofed IP packets requires support of routers, host-based methods, and administrative controls, whereas tracing of IP packets involves special traceback equipment or traceback features in routers. The following section discusses both IP spoofing detection and IP spoofing traceback techniques.

Spoofed Packet Detection

Detection of a spoofed packet can start as early as at Layer 2. Switches with the *IP Source Guard* feature[8] match the MAC address of the host with a *Dynamic Host Configuration Protocol* (DHCP)-assigned dynamic or administratively assigned static IP address. Packets that do not have the correct IP source address for that particular MAC address are dropped, thereby limiting the ability of hosts connected to such a switch to send a packet with their neighbor's address. The IP Source Guard feature works very well for interfaces with a single IP address, but one interface can be assigned multiple IP addresses, and that may cause problems. The same problems can occur with *Network Address Translation* (NAT), where hosts might get different IP addresses several times. Routers work at Layer 3 in networks, and they know which interface a network is connected to and what network addresses can be expected to come from that network. If the outgoing packet from an interface does not have the network address of that interface, then the packet is spoofed and the router can stop that packet at that point; however, if the attacker is spoofing an IP address of a host on the same network (most likely in the attacks where they will be sniffing the replies), then this technique is not really helpful. The same logic can be used for an incoming packet; if a packet destined for an interface has a source address of the same network as the interface, then it is a spoofed packet. Routers can detect spoofed packets only when the packets pass through them, and if the target and attacker are both on the same subnet then this technique does not work.

Hosts receiving a suspicious packet can also use certain techniques to determine whether or not the IP address is spoofed. The first (and easiest) one is to send a request to the address of the packet and wait for the response; most of the time the spoofed addressees do not belong to active hosts and hence no response is sent.

Another method is to check the *Time to Live* (TTL) value of the packet, and then send a request to the spoofed host. If the reply comes, you can compare the TTL of both packets. Most probably the TTL values will not match. But of course it is also possible that these TTL values are the same but the packet is coming from a different source, and conversely. Packets generated by different operating systems differ slightly in values of certain fields; for example, in *Internet Control Message Protocol* (ICMP) *ping* packets, you can examine the data payload to determine the operating system. Windows fills the packet with letters of the alphabet, whereas Linux puts numbers in the data portion. If the suspicious packet does not have the same characteristics as the legitimate packet, that is evidence it was not sent from the IP address that is in its source address field. You can also use IP identification numbers to determine whether a packet is actually coming from the said source. For legitimate packets the IP ID is close in value, but this method is not reliable because the attacker can ping the said source and determine the IP ID that it is using, and then craft packets that will seem legitimate. In all these techniques we are trying to determine only whether or not a packet is spoofed, and taking all these steps for all packets would be prohibitive from an overhead standpoint. Thus you should either randomly check packets or determine some suspicious activity that would trigger further investigation for spoofed-packet detection. The next section addresses measures you can take to trace a spoofed packet back to its real source.

Tracing Spoofed IP Packets

IP traceback technology plays an important role in discovering the source of spoofed packets. Hop-by-hop traceback and logging of suspicious packets in routers are the two main methods for tracing the spoofed IP packets back to their source.

When a node detects that it is a victim of flood attack, it can inform the *Internet Service Provider* (ISP). In flood attacks the ISP can determine the router that is sending this stream to the victim, and then it can determine the next router, and so on. It reaches either to the source of the flood attack or the end of its administrative domain; for this case it can ask the ISP for the next domain to do the same thing. This technique is useful only if the flood is ongoing.

As mentioned earlier, a router has an idea of the IP addresses that should be arriving at its interfaces. If it sees any packet that does not seem to belong to the address range for its interface, it can log the packet as suspicious. Appropriately timed broadcasts among different domains to detect spoofed packets can help administrators of different networks trace spoofed IP packets back to their source.

IP Spoofing and IPv6

IP spoofing detection, or in other words validating the source address of an IPv6 packet, is a little more complicated than the process for IPv4. A host using IPv6 may potentially have multiple addresses. Again the problem inside the Local Area Network is to associate the IPv6 address with the Layer 2 or MAC address. Among peers on the same network, you can use *Neighbor Discovery* or *Secure Neighbor Discovery* (SEND) advertisements to verify the source address in a packet. You can verify source addresses of packets arriving from nodes outside the network by using the *Authentication Header* (AH) in IPv6 datagrams. You can use agreed-upon parameters between source and destination to calculate authentication information on header fields that does not change during transit. Although this process will not prevent someone from signing a spoofed address, it does provide a means to authenticate the identity of the source.

IPv6 and IPv4 network interconnections will likely face spoofing problems. IPv6 packets are usually encapsulated in IPv4 packets to travel across the non-IPv6 supporting networks. The IPv6 interim mechanism "6to4"[10, 11] uses automatic IPv6-to-IPv4 tunneling to interconnect networks using different IP versions. This mechanism uses 6to4 routers and 6to4 *Relay Routers* that accept and decapsulate IPv4 traffic from anywhere. There are no constraints on such embedded packets. Relay routers act as bridges between IPv6 and 6to4 networks and can be tricked into sending spoofed traffic anywhere. Also, anyone can send tunneled spoofed traffic to a 6to4 router, and the router will believe that it is coming from a legitimate relay. There is no simple way to prevent such attacks, and longer-term solutions are needed in both IPv6 and IPv4 networks.

Conclusion

IP spoofing is a difficult problem to tackle, because it is related to the IP packet structure. IP packets can be exploited in several ways. Because attackers can hide their identity with IP spoofing, they can make several network attacks. Although there is no easy solution for the IP spoofing problem, you *can* apply some simple proactive and reactive methods at the nodes, and use the routers in the network to help detect a spoofed packet and trace it back to its originating source.

References

[1] Alaaeldin A. Aly, "Tracking and Tracing Spoofed IP Packets to Their Sources," Proceedings of 6th annual conference, UAEU April 2005.

[2] S.J. Templeton and K.E. Levitt, "Detecting Spoofed Packets," DARPA Information Survivability Conference and Exposition, 2003.

[3] "IP Spoofing an Introduction," **http://www.securityfocus.com/infocus/1674**

[4] **http://www.phrack.org/issues.html?issue=48&id=14#article**

[5] **http://www.hping.org**

[6] **http://www.insecure.org/nmap**

[7] **http://www.ietf.org/internet-drafts/draft-baker-sava-operational-00.txt**

[8] **http://tools.ietf.org/html/draft-baker-sava-cisco-ip-source-guard-00**

[9] **http://tools.ietf.org/id/draft-baker-sava-implementation-00.txt**

[10] **http://tools.ietf.org/html/draft-ietf-v6ops-6to4-security-04**

[11] Carpenter, B., Fink, B., and Moore, K., "Connecting IPv6 Routing Domains Over the IPv4 Internet," *The Internet Protocol Journal,* Volume 3, No. 1, March 2000.

[12] Wesley Eddy, "Defenses Against TCP SYN Flooding Attacks," *The Internet Protocol Journal,* Volume 9, No. 4, December 2006.

FARHA ALI holds a BE in Computer Engineering from NED University, Pakistan, and an MS in Computer Engineering from Clemson University, South Carolina, with a focus area in Computer Communications. She is a member of American Mensa and ACM. Her research papers (co-authored with her advisor) as a PhD student at Clemson University's Computer Science Department were published in IEEE's Conferences on Web Intelligence and Web Services. She is a Sun Certified Java Programmer and a Certified Ethical Hacker. Her main interests are Distributed Computing, Network Security, and Semantic Web. Currently she is working as a faculty member at Lander University's Department of Mathematics and Computing. She teaches mainly Networking and Programming courses.
E-mail: **fali@lander.edu**

Standards for Information Security Managment

by William Stallings

To effectively assess the security needs of an organization and to evaluate and choose various security products and policies, the manager responsible for security needs some systematic way of defining the requirements for security and characterizing the approaches to satisfy those requirements. This process is difficult enough in a centralized data processing environment; with the use of local- and wide-area networks (LANs and WANs, respectively), the problems are compounded.

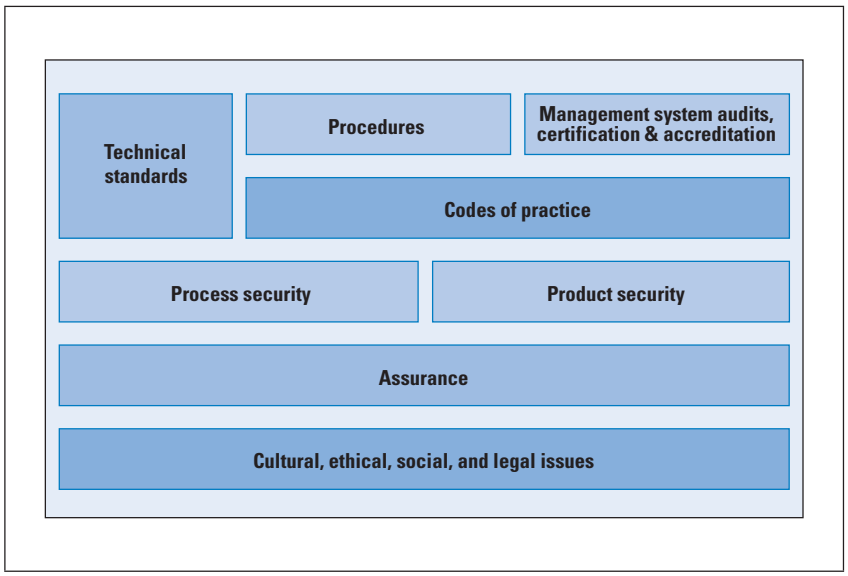
The challenges for management in providing information security are formidable. Even for relatively small organizations, information system assets are substantial, including databases and files related to personnel, company operation, financial matters, and so on. Typically, the information system environment is complex, including a variety of storage systems, servers, workstations, local networks, and Internet and other remote network connections. Managers face a range of threats always growing in sophistication and scope. And the range of consequences for security failures, both to the company and to individual managers, is substantial, including financial loss, civil liability, and even criminal liability.

Standards for providing information system security become essential in such circumstances. Standards can define the scope of security functions and features needed, policies for managing information and human assets, criteria for evaluating the effectiveness of security measures, techniques for ongoing assessment of security and for the ongoing monitoring of security breaches, and procedures for dealing with security failures.

Figure 1, based on [1], suggests the elements that, in an integrated fashion, constitute an effective approach to information security management. The focus of this approach is on two distinct aspects of providing information security: process and products. *Process security* looks at information security from the point of view of management policies, procedures, and controls. *Product security* focuses on technical aspects and is concerned with the use of certified products in the IT environment when possible. In Figure 1, the term *technical standards* refers to specifications that refer to aspects such as IT network security, digital signatures, access control, nonrepudiation, key management, and hash functions. Operational, management, and technical *procedures* encompass policies and practices that are defined and enforced by management. Examples include personnel screening policies, guidelines for classifying information, and procedures for assigning user IDs. *Management system audits, certification, and accreditation* deals with management policies and procedures for auditing and certifying information security products.

Codes of practice refer to specific policy standards that define the roles and responsibilities of various employees in maintaining information security. *Assurance* deals with product and system testing and evaluation. *Cultural, ethical, social, and legal issuers* refer to human factors aspects related to information security.

Figure 1: Information Security Management Elements



Many standards and guideline documents have been developed in recent years to aid management in the area of information security. The two most important are *ISO 17799*, which deals primarily with process security, and the *Common Criteria*, which deals primarily with product security. This article surveys these two standards, and examines some other important standards and guidelines as well.

ISO 17799

An increasingly popular standard for writing and implementing security policies is *ISO 17799* "Code of Practice for Information Security Management." (ISO 17799 will eventually be reissued as ISO 27002 in the new ISO 27000 family of security standards). ISO 17799 is a comprehensive set of controls comprising best practices in information security. It is essentially an internationally recognized generic information security standard. Table 1 summarizes the area covered by this standard and indicates the objectives for each area.

Table 1: ISO 17799 Areas and Objectives

Security Policy Provide management direction and support for information security in accordance with business requirements and relevant laws and regulations. **Organization of Information Security** Manage information security within the organization. Maintain the security of the organization's information and information processing facilities that are accessed, processed, communicated to, or managed by external parties. **Asset Management** Achieve and maintain appropriate protection of organizational assets. Ensure that information receives an appropriate level of protection. **Human Resources Security** Ensure that employees, contractors, and third-party users (1) understand their responsibilities and are suitable for the roles they are considered for; (2) are aware of information security threats and concerns; (3) exit an organization or change employment in an orderly manner. **Physical and Environmental Security** Prevent unauthorized physical access, damage, and interference to the organization's premises and information. Prevent loss, damage, theft, or compromise of assets and interruption to the organization's activities. **Communications and Operations Management** Develop controls for operational procedures, third-party service delivery management, system planning, malware protection, backup, network security management, media handling, information exchange, e-commerce services, and monitoring.	**Access Control** Develop controls for business requirements for user access, user responsibilities, network access control, OS access control, application access control, and information access control. **Information Systems Acquisition, Development, and Maintenance** Develop controls for correct processing in applications, cryptographic functions, system file security, support process security, and vulnerability management. **Information Security Incident Management** Ensure information security events and weaknesses associated with information systems are communicated in a manner that allows timely corrective action to be taken. Ensure a consistent and effective approach is applied to the management of information security incidents. **Business Continuity Management** Counteract interruptions to business activities to protect critical business processes from the effects of major failures of information systems or disasters and to ensure their timely resumption. **Compliance** Avoid breaches of any law, statutory, regulatory, or contractual obligations, and of any security requirements. Ensure compliance of systems with organizational security policies and standards. Maximize the effectiveness of and minimize interference to and from the information systems audit process.

With the increasing interest in security, ISO 17799 certification, provided by various accredited bodies, has been established as a goal for many corporations, government agencies, and other organizations around the world. ISO 17799 offers a convenient framework to help security policy writers structure their policies in accordance with an international standard.

Much of the content of ISO 17799 deals with security controls, which are defined as practices, procedures, or mechanisms that may protect against a threat, reduce a vulnerability, limit the effect of an unwanted incident, detect unwanted incidents, and facilitate recovery. Some controls deal with security management, focusing on management actions to institute and maintain security policies. Other controls are operational; they address the correct implementation and use of security policies and standards, ensuring consistency in security operations and correcting identified operational deficiencies. These controls relate to mechanisms and procedures that are primarily implemented by people rather than systems.

Finally, there are technical controls; they involve the correct use of hardware and software security capabilities in systems. These controls range from simple to complex measures that work together to secure critical and sensitive data, information, and IT systems functions. This concept of controls cuts across all the areas listed in Table 1.

To give some idea of the scope of ISO 17799, we examine several of the security areas discussed in that document. *Auditing* is a key security management function that is addressed in multiple areas within the document. First, ISO 17799 lists key data items that should, when relevant, be included in an audit log:

- User IDs
- Dates, times, and details of key events, for example, log-on and log-off
- Terminal identity or location if possible
- Records of successful and rejected system access attempts
- Records of successful and rejected data and other resource access attempts
- Changes to system configuration
- Use of privileges
- Use of system utilities and applications
- Files accessed and the kind of access
- Network addresses and protocols
- Alarms raised by the access control system
- Activation and deactivation of protection systems, such as antivirus systems and intrusion detection systems

It provides a useful set of guidelines for implementation of an auditing capability:

1. Audit requirements should be agreed upon by appropriate management.
2. The scope of the checks should be agreed upon and controlled.
3. The checks should be limited to read-only access to software and data.
4. Access other than read-only should be allowed only for isolated copies of system files, which should be erased when the audit is completed or given appropriate protection if there is an obligation to keep such files under audit documentation requirements.
5. Resources for performing the checks should be explicitly identified and made available.
6. Requirements for special or additional processing should be identified and agreed upon.

7. All access should be monitored and logged to produce a reference trail; the use of timestamped reference trails should be considered for critical data or systems.
8. All procedures, requirements, and responsibilities should be documented.
9. The person(s) carrying out the audit should be independent of the activities audited.

Under the area of communications and operations management, ISO 17799 includes *network security management*. One aspect of this management is concerned with network controls for networks owned and operated by the organization. The document provides implementation guidance for these in-house networks. An example of a control follows: Restoration procedures should be regularly checked and tested to ensure that they are effective and that they can be completed within the time allotted in the operational procedures for recovery. Similarly, the document provides guidance for security controls for network services provided by outside vendors. An example of guidance in this area follows: The ability of the network service provider to manage agreed-upon services in a secure way should be determined and regularly monitored, and the right to audit should be agreed upon.

As can be seen, some ISO 17700 specifications are detailed and specific, whereas others are quite general.

Common Criteria

The *Common Criteria for Information Technology Security Evaluation* (CC) is a joint international effort by numerous national standards organizations and government agencies[3,4,5]. U.S. participation is by the *National Institute of Standards and Technology* (NIST) and the *National Security Agency* (NSA). CC defines a set of IT requirements of known validity that can be used in establishing security requirements for prospective products and systems. The CC also defines the *Protection Profile* (PP) construct that allows prospective consumers or developers to create standardized sets of security requirements that will meet their needs.

The aim of the CC specification is to provide greater confidence in the security of IT products as a result of formal actions taken during the process of developing, evaluating, and operating these products. In the development stage, the CC defines sets of IT requirements of known validity that can be used to establish the security requirements of prospective products and systems. Then the CC details how a specific product can be evaluated against these known requirements, to provide confirmation that it does indeed meet them, with an appropriate level of confidence. Lastly, when in operation the evolving IT environment may reveal new vulnerabilities or concerns. The CC details a process for responding to such changes, and possibly reevaluating the product.

Following successful evaluation, a particular product may be listed as CC certified or validated by the appropriate national agency, such as NIST or NSA in the United States. That agency publishes lists of evaluated products, which are used by government and industry purchasers who need to use such products.

The CC defines a common set of potential *security requirements* for use in evaluation. The term *Target of Evaluation* (TOE) refers to that part of the product or system that is subject to evaluation. The requirements fall into two categories:

- *Functional requirements:* Define desired security behavior. CC documents establish a set of security functional components that provide a standard way of expressing the security functional requirements for a TOE.
- *Assurance requirements:* The basis for gaining confidence that the claimed security measures are effective and implemented correctly. CC documents establish a set of assurance components that provide a standard way of expressing the assurance requirements for a TOE.

Both functional requirements and assurance requirements are organized into classes: A *class* is a collection of requirements that share a common focus or intent. Each of these classes contains numerous families. The requirements within each *family* share security objectives but differ in emphasis or rigor. For example, the audit class contains six families dealing with various aspects of auditing (for example, audit data generation, audit analysis, and audit event storage). Each family, in turn, contains one or more components. A *component* describes a specific set of security requirements and is the smallest selectable set of security requirements for inclusion in the structures defined in the CC.

For example, the cryptographic support class of functional requirements includes two families: cryptographic key management and cryptographic operation. The cryptographic key management family has four components, which are used to specify key generation algorithm and key size; key distribution method; key access method; and key destruction method. For each component, a standard may be referenced to define the requirement. Under the cryptographic operation family, there is a single component, which specifies an algorithm and key size based on a an assigned standard.

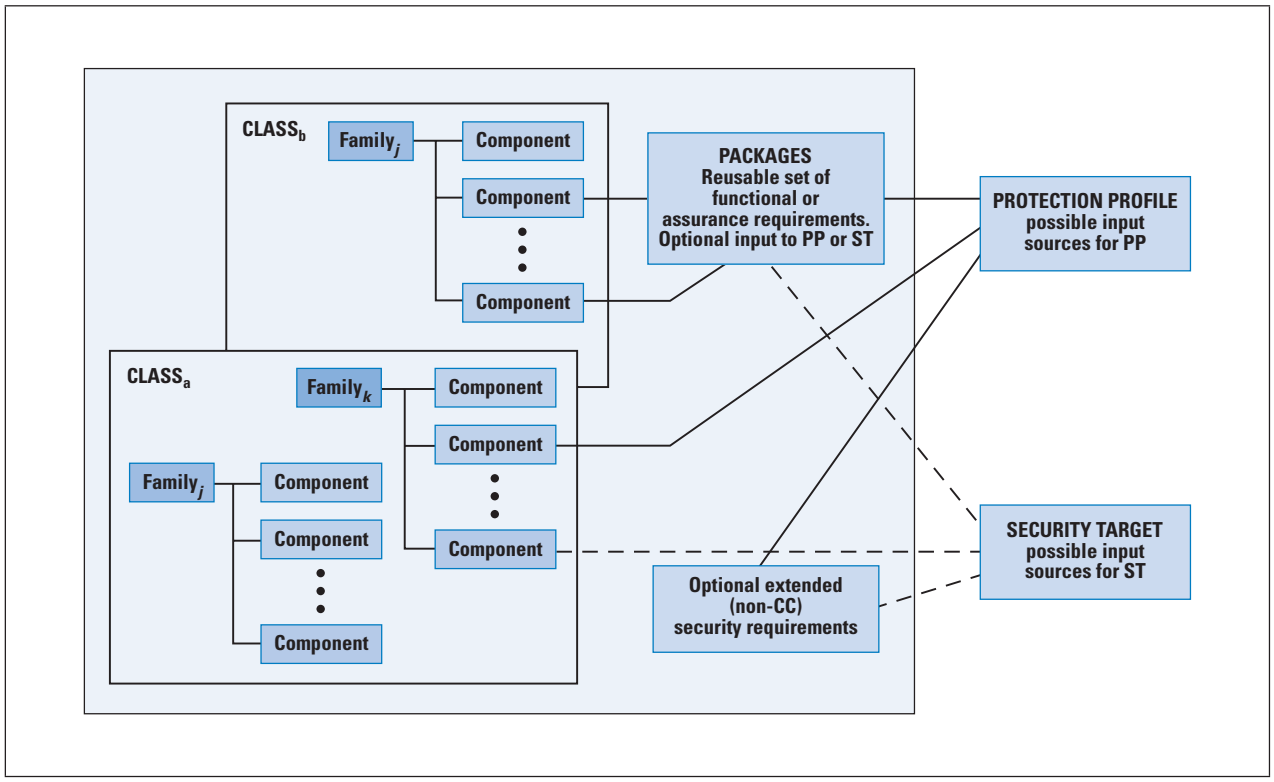
Sets of functional and assurance components may be grouped together into reusable packages, which are known to be useful in meeting identified objectives. An example of such a package would be functional components required for *Discretionary Access Controls*.

The CC also defines two kinds of documents that can be generated using the CC-defined requirements.

- *Protection profiles (PPs):* Define an implementation-independent set of security requirements and objectives for a category of products or systems that meet similar consumer needs for IT security. A PP is intended to be reusable and to define requirements that are known to be useful and effective in meeting the identified objectives. The PP concept has been developed to support the definition of functional standards and as an aid to formulating procurement specifications. The PP reflects user security requirements.
- *Security targets (STs):* Contain the IT security objectives and requirements of a specific identified TOE and define the functional and assurance measures offered by that TOE to meet stated requirements. The ST may claim conformance to one or more PPs and forms the basis for an evaluation. The ST is supplied by a vendor or developer.

Figure 2 illustrates the relationship between requirements on the one hand and profiles and targets on the other. For a PP, a user can select many components to define the requirements for the desired product. The user may also refer to predefined packages that assemble numerous requirements commonly grouped together within a product requirements document. Similarly, a vendor or designer can select numerous components and packages to define an ST.

Figure 2: Organization and Construction of Common Criteria Requirements



As an example for the use of the CC, consider the smart card. The protection profile for a smart card, developed by the *Smart Card Security User Group,* provides a simple example of a PP. This PP describes the IT security requirements for a smart card to be used in connection with sensitive applications, such as banking industry financial payment systems. The assurance level for this PP is *Evaluation Assurance Level* (EAL) 4, which is described subsequently. The PP lists *threats* that must be addressed by a product that claims to comply with this PP. The threats include the following:

- *Physical probing:* May entail reading data from the TOE through techniques commonly employed in *Integrated Circuit* (IC) failure analysis and IC reverse engineering efforts.
- *Invalid input:* Invalid input may take the form of operations that are not formatted correctly, requests for information beyond register limits, or attempts to find and execute undocumented commands. The result of such an attack may be a compromise in the security functions, generation of exploitable errors in operation, or release of protected data.
- *Linkage of multiple operations:* An attacker may observe multiple uses of resources or services and, by linking these observations, deduce information that may reveal security function data.

Following a list of threats, the PP turns to a description of *security objectives*, which reflect the stated intent to counter identified threats or comply with any organizational security policies identified. Nineteen objectives are listed, including the following:

- *Audit:* The system must provide the means of recording selected security-relevant events, so as to assist an administrator in the detection of potential attacks or misconfiguration of the system security features that would leave it susceptible to attack.
- *Fault insertion:* The system must be resistant to repeated probing through insertion of erroneous data.
- *Information leakage:* The system must provide the means of controlling and limiting the leakage of information in the system so that no useful information is revealed over the power, ground, clock, reset, or I/O lines.

Security requirements are provided to thwart specific threats and to support specific policies under specific assumptions. The PP lists specific requirements in three general areas: TOE security functional requirements, TOE security assurance requirements, and security requirements for the IT environment. In the area of *security functional requirements,* the PP defines 42 requirements from the available classes of security functional requirements.

For example, for security auditing, the PP stipulates what the system must audit; what information must be logged; what the rules are for monitoring, operating, and protecting the logs; and so on. Functional requirements are also listed from the other functional requirements classes, with specific details for the smart card operation.

The PP defines 24 *security assurance requirements* from the available classes of security assurance requirements. These requirements were chosen to demonstrate:

- The quality of the product design and configuration
- That adequate protection is provided during the design and implementation of the product
- That vendor testing of the product meets specific parameters
- That security functions are not compromised during product delivery
- That user guidance, including product manuals pertaining to installation, maintenance, and use, are of a specified quality and appropriateness

The PP also lists *Security Requirements of the IT Environment*. They cover the following topics:

- Cryptographic key distribution
- Cryptographic key destruction
- Security roles

The final section of the PP (excluding appendices) is a lengthy rationale for all the selections and definitions in the PP. The PP is an industrywide effort designed to be realistic in its ability to be met by a variety of products with a variety of internal mechanisms and implementation approaches.

The concept of *Evaluation Assurance* is a difficult one to define. Further, the degree of assurance required varies from one context and one function to another. To structure the need for assurance, the CC defines a scale for rating assurance consisting of seven EALs ranging from the least rigor and scope for assurance evidence (EAL 1) to the most (EAL 7). The levels are as follows:

- *EAL 1: Functionally tested:* For environments where security threats are not considered serious. It involves independent product testing with no input from the product developers. The intent is to provide a level of confidence in correct operation.
- *EAL 2: Structurally tested:* Includes a review of a high-level design provided by the product developer. Also, the developer must conduct a vulnerability analysis for well-known flaws. The intent is to provide a low to moderate level of independently assured security.

- *EAL 3: Methodically tested and checked:* Requires a focus on the security features, including requirements that the design separate security-related components from those that are not; that the design specifies how security is enforced; and that testing be based both on the interface and the high-level design, rather than a black box testing based only on the interface. It is applicable where the requirement is for a moderate level of independently assured security, with a thorough investigation of the TOE and its development without incurring substantial reengineering costs.
- *EAL 4: Methodically designed, tested, and reviewed:* Requires both a low-level and a high-level design specification; requires that the interface specification be complete; requires an abstract model that explicitly defines security for the product; and requires an independent vulnerability analysis. It is applicable in those circumstances where developers or users require a moderate to high level of independently assured security in conventional commodity TOEs, and there is willingness to incur some additional security-specific engineering costs.
- *EAL 5: Semiformally designed and tested:* Provides an analysis that includes all of the implementation. Assurance is supplemented by a formal model and a semiformal presentation of the functional specification and high-level design and a semiformal demonstration of correspondence. The search for vulnerabilities must ensure resistance to penetration attackers with a moderate attack potential. Covert channel analysis and modular design are also required.
- *EAL 6: Semiformally verified design and tested:* Permits a developer to gain high assurance from application of specialized security engineering techniques in a rigorous development environment, and to produce a premium TOE for protecting high-value assets against significant risks. The independent search for vulnerabilities must ensure resistance to penetration attackers with a high attack potential.
- *EAL 7: Formally verified design and tested:* The formal model is supplemented by a formal presentation of the functional specification and high-level design, showing correspondence. Evidence of developer "white box" testing of internals and complete independent confirmation of developer test results are required. Complexity of the design must be minimized.

The first four levels reflect various levels of commercial design practice. Only at the highest of these levels (EAL 4) is there a requirement for any source code analysis, and this analysis is required only for a portion of the code. The top three levels provide specific guidance for products developed using security specialists and security-specific design and engineering approaches.

National Institute of Standards and Technology

NIST has produced a large number of *Federal Information Processing Standards Publications* (FIPS PUBs) and special publications (SPs) that are enormously useful to security managers, designers, and implementers. Following are a few of the most significant and general. *FIPS PUB 200* "Minimum Security Requirements for Federal Information and Information Systems," is a standard that specifies minimum security requirements in 17 security-related areas with regard to protecting the confidentiality, integrity, and availability of federal information systems and the information processed, stored, and transmitted by those systems[6].

NIST *SP 800-100* "Information Security Handbook: A Guide for Managers," provides a broad overview of information security program elements to assist managers in understanding how to establish and implement an information security program[7]. Its topical coverage overlaps considerably with ISO 17799.

Several other NIST publications are of general interest. *SP 800-55* "Security Metrics Guide for Information Technology Systems," provides guidance on how an organization, through the use of metrics, identifies the adequacy of in-place security controls, policies, and procedures[8]. *SP 800-27* "Engineering Principles for Information Technology Security (A Baseline for Achieving Security)," presents a list of system-level security principles to be considered in the design, development, and operation of an information system[9]. *SP 800-53* "Recommended Security Controls for Federal Information Systems," lists management, operational, and technical safeguards or countermeasures prescribed for an information system to protect the confidentiality, integrity, and availability of the system and its information[10].

Other Standards and Guidelines

Another important set of standards is the *Control Objectives for Information and Related Technology* (COBIT)[11], a business-oriented set of standards for guiding management in the sound use of information technology. It has been developed as a general standard for information technology security and control practices and includes a general framework for management, users, IS audit, and security practitioners. COBIT also has a process focus and a governance flavor; that is, management's need to control and measure IT is a focus point. COBIT was developed under the auspices of a professional organization, the *Information Systems Audit and Control Association* (ISACA). The documents are quite detailed and provide a practical basis for not only defining security requirements but also implementing them and verifying compliance.

Another excellent source of information is "The Standard of Good Practice for Information Security" from the *Information Security Forum*. The standard is designed as an aid to organizations in understanding and applying best practices for information security. Because it addresses security from a business perspective, The Standard appropriately recognizes the intersection between organizational factors and security factors.

In addition to these standards, numerous informal guidelines are widely consulted by organizations in developing their own security policy. The *CERT Coordination Center* (**www.cert.org**) has an Evaluations and Practices section of its Website with a variety of documents and training aids related to information security for organizations. The *Chief Information Officers Council* (**cio.gov**) has published a collection of Best Practices and other documents related to organizational security.

References

[1] Eloff, J., and Eloff, M., "Information Security Management," *Proceedings of SAICSIT 2003,* South African Institute of Computer Scientists and Information Technologists, 2003.

[2] International Organization for Standardization, "ISO/IEC 27001 – Information technology – Security Techniques – Information security management systems – Requirements," June 2005.

[3] Common Criteria Project Sponsoring Organisations, "Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model," CCIMB-2004-01-001, January 2004.

[4] Common Criteria Project Sponsoring Organisations, "Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements," CCIMB-2004-01-002, January 2004.

[5] Common Criteria Project Sponsoring Organisations, "Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components," CCIMB-2006-09-003, September 2006.

[6] National Institute of Standards and Technology, "Minimum Security Requirements for Federal Information and Information Systems," FIPS PUB 200, March 2006.

[7] National Institute of Standards and Technology, "Information Security Handbook: A Guide for Managers," NIST Special Publication 800-100, October 2006.

[8] "Security Metrics Guide for Information Technology Systems," NIST Special Publication 800-55, July 2003.

[9] National Institute of Standards and Technology, "Engineering Principles for Information Technology Security (A Baseline for Achieving Security)," NIST Special Publication 800-27, June 2004.

[10] National Institute of Standards and Technology, "Recommended Security Controls for Federal Information Systems," NIST Special Publication 800-53, February 2005.

[11] IT Governance Institute, "COBIT 4.0.," USA, 2005.

[12] Information Security Forum, "The Standard of Good Practice for Information Security," 2005.

WILLIAM STALLINGS is a consultant, lecturer, and author of more than a dozen books on data communications and computer networking. His latest book, with Lawrie Brown, is *Computer Security: Principles and Practice* (Prentice Hall, 2007). He maintains a computer science resource site for computer science students and professionals at **`WilliamStallings.com/StudentSupport.html`** and is on the editorial board of *Cryptologia*. He has a Ph.D. in computer science from M.I.T. He can be reached at **`ws@shore.net`**.

Looking Toward the Future

by Vint Cerf, Google

The *Internet Corporation for Assigned Names and Numbers* (ICANN) was formed 9 years ago, following a period of considerable debate about the institutionalization of the basic functions performed by the *Internet Assigned Numbers Authority* (IANA)[1]. Nearly simultaneous with the inauguration of ICANN in September 1998 came the unexpected and untimely death of the man, Jonathan B. Postel[2], who had responsibility for these functions for more than a quarter century. The organization began with very limited sources of funds, a small and overworked staff, and contentious debate about its organizational structure, policy apparatus, and operational procedures. The organization underwent substantial change through its *Evolution and Reform Process* (ERP)[3]. Among the more difficult constituencies to accommodate in the organization's policy-making process was the general public. An *At-Large Advisory Committee* (ALAC)[4] emerged from the ERP and has recently formed *Regional At-Large Organizations* (RALOs) in all of ICANN's five regions.

Today, ICANN is larger, more capable, more international, and better positioned to fulfill its mandate. It stands for one global, interoperable Internet, and the model of stakeholder representation has worked. But the Internet and its vast user population have grown during the same time by a factor of more than 20 in all dimensions. The 50 million users of 1997 have become nearly 1.2 billion users today. The 22 million hosts on the network have increased to nearly 500 million today. The bandwidth of the core data circuits in the Internet have grown from 622 million bits per second to between 10 and 40 billion bits per second. This dramatic growth in physical size has been accompanied by an equally dramatic growth in the number and diversity of applications running on the Internet. All forms of media now appear on and are carried by Internet packets. Consumers of information are producing more and more of it themselves with e-mail, blogs, instant messaging, social and game-playing Websites, video uploads, and podcasts. The Internet continues to evolve and although ICANN has achieved more than most people realize, it must continue to evolve along with it.

Operational Priorities

ICANN's primary responsibility is to contribute to the security and stability of the Internet system of unique identifiers. In the most direct way, it carries out this mandate through its operation of the IANA. There is no doubt that the conduct of this function in an exemplary fashion is essential not only to ICANN's mission but also to inspiring confidence in ICANN as an organization.

But ICANN's role in the Internet goes beyond these specific IANA functions. ICANN is an experiment in the balancing of multiple stakeholder interests in policy about the implementation, operation, and use of the *Domain Name System* (DNS) and the address spaces of the Internet. Its policy choices can directly affect the business models of operating entities involved in the management of domain names and Internet addresses. The privacy and Internet-related rights of registrants and, more generally, Internet users may also be directly affected. Some policy choices raise public policy concerns in the view of governments and methods and will be needed to factor such concerns into the making of ICANN policy.

Effective, fair, and timely policy development should be a priority for ICANN. That this policy development needs to be achieved in a global setting is simply another challenge to be met. ICANN leadership and staff must seek to maintain and improve the ability of all of ICANN's many constituencies to achieve consensus or at least to prepare the ICANN Board to make choices when consensus may not be forthcoming. Because policies often have technical, economic, social, and governance implications, it is vital that ICANN's practices draw on expertise in all these domains.

Clarity in the roles and responsibilities of the many participants in the Internet arena, especially those with specific interest in ICANN policies and practices, will be helpful and should be documented. In some cases, the documentation might take the form of relatively formal relationships such as the contracts between ICANN and domain-name registries and registrars. In other cases, they may need only to characterize in plain terms the roles that each party plays.

In some areas, such as root-zone operation, excellence can be measured in such terms as responsiveness, scalability, resilience to disruption, and ability to adapt to changing needs such as *Domain Name System Security* (DNSSEC)[5], *Internationalized Domain Names* (IDNs)[6], and the addition of IPv6 records to the root zone. Many parties currently play a role in the maintenance of the root-zone file, and clear documentation of responsibility and lines of authority will be beneficial. As the technology of the Internet continues to evolve, the roles of various parties may need to change to meet the objectives of stability and security of the Internet system of unique identifiers. Managing the evolution of these roles represents another priority for policy development and implementation.

Because of the potential effect of decisions made through the ICANN policy process, it is important to implement checks and balances that make all aspects of ICANN's operation accountable and transparent. Work is still necessary in this area so that independent review of legitimate concerns arising out of policy making is possible when deemed necessary.

At the same time, it is vital that the mechanisms chosen do not have the effect of locking up the policy-making process and preventing any decisions from being made. We need to seek a balance between a potentially unfair tyranny of the majority and an equally unacceptable tyranny of the minority.

The general success of the *Uniform Dispute Resolution Process* (UDRP)[7] suggests that ICANN should seek mechanisms for resolving disputes arising in connection with implementing ICANN policies that scale, permit choice without abusive "forum shopping," and make efficient use of ICANN resources.

Outreach, transparency, and broadly participatory processes on an international basis are not inexpensive. It is vital for ICANN to continue to refine its models for sustainable operation, accounting for the economics of the various actors in the Internet arena that rely on ICANN's operation, and fairly apportioning costs of ICANN operation to appropriate sources of support. Not all of the beneficiaries of ICANN's work derive the same level of revenue from the Internet (and some, none at all). ICANN must account for this discrepancy when devising mechanisms for supporting its operation, and it should work to make transparent the need to provide services to parties who may not be able to contribute commensurate with cost. Adequate and stable funding for ICANN is necessary if ICANN is to fulfill its charter. Over the past several years, ICANN has significantly increased its ability to staff vital functions, contributing to the effectiveness of the organization. It should be a priority to assure adequate reserves to weather unanticipated expenses or periods of decreased income.

Organizational Perspectives

ICANN is a multistakeholder institution operating in the private sector but including the involvement of governments. Throughout its history, ICANN has sought to draw on international resources and to collaborate, coordinate, and cooperate with institutions whose expertise and responsibilities can assist ICANN in the achievement of its goals. ICANN should seek to establish productive relationships with these institutions, cementing its own place in the Internet universe while confining its role to its principal responsibilities.

As part of its normal operation, ICANN engages in self-examination and external review of the effectiveness of its organizational structure and processes. Improvements in all aspects of ICANN operation and structure will increase confidence in the organization and its ability to sustain long-term operation.

Finding and engaging competent participants and leaders in each of ICANN's constituent parts must be a priority. ICANN should seek to improve its ability to identify from around the world and attract highly qualified staff, executive leadership, board, and supporting organization participants. It is possible and even likely that improvements in the processes by which this process is done today will have significant payoff in the future.

Although ICANN does not bear a specific responsibility for achieving the *Millennium Development Goals* (MDGs) developed during the conduct of the *World Summit on the Information Society* (WSIS)[8], it has an opportunity to contribute to them both directly and indirectly. Its operation of its IANA functions and support for actors in the domain-name, Internet-address, and standards-development areas provides ICANN with a specific opportunity. Participation in forums dedicated to developing policies for Internet expansion and use offer indirect ways for ICANN to draw upon and provide expertise in these areas.

It has been demonstrated that the presence of ICANN staff in various regions and time zones around the world and familiarity with local languages and customs has been beneficial to parties reliant on ICANN for its services. ICANN should continue to seek ways to improve its effectiveness in this area. The introduction of the Fellowship program that supports the participation of qualified candidates in ICANN-related activities is a vital step in facilitating ICANN's outreach to the developing world. We should pursue expansion of this program through partnerships with other like-minded organizations in the interest of the globalization of ICANN.

It is possible that the present formulation of ICANN as a not-for-profit, charitable research and education entity under California law could be beneficially adapted to a more international framework. As part of its long-term strategic development, ICANN should evaluate a variety of alternatives on the possibility that a change could increase the effectiveness of its operation.

The successful creation of five Regional At-Large Organizations, one in each of ICANN's five regions, needs to be followed by a serious effort to engage these entities in the formulation of ICANN policies and in dialog with the general user community. The various constituency reviews that form part of ICANN's normal processes should address the role of these entities in the conduct of ICANN business. To the extent that civil society is not fully represented through the *Governmental Advisory Committee* (GAC)[9] and the ALAC/RALO system, an organizational home may be needed to accommodate the interests of that constituency.

The five *Regional Internet Registries* (RIRs)[10] represent a key element in the Internet and ICANN pantheon. The RIRs have responsibility for allocating IP address space to Internet service providers and sometimes individual end-user organizations. They are the means by which bottom-up global policy is developed and recommended, through the *Number Resource Organization* (NRO)[11], to ICANN. It will require substantial coordination and cooperation between the RIRs and ICANN to work through the coming years of depletion of available new IPv4 address space and the rising implementation of the new IPv6 address space.

There is little doubt that economic incentives will emerge that will distort fair and neutral IPv4 address-space allocations as the available space is depleted. Minimizing the effect of this transition will be the joint responsibility of ICANN and the RIRs.

Similarly, ICANN's cooperative relationship with the *Root Server Operators*[12] will also demand coordination and capacity building as IPv4 and IPv6 addresses are associated with old and new domain names and as the IPv6 infrastructure grows. A vital objective is to assure that the IPv6 Internet and the IPv4 Internet are, to the extent possible, completely and totally coterminous. Every termination needs to be reachable through both address spaces. In the absence of this uniformity, some IPv6 addresses may be unreachable from others, defeating the goal of a single, interoperable, and fully reachable network.

Meeting the Challenges

As ICANN approaches the close of its first decade, the operational Internet will be turning 25. In the course of its evolution, it has become a global digital canvas on which a seemingly endless array of applications has been painted. Despite the broad swath of its current applications, it is almost certain that many, many more will be invented. All of them will rely, for the foreseeable future, on the basic architecture of the system, including the global Internet address space and the DNS. But the structure will become more complex. Two parallel address spaces, IPv4 and IPv6, will be in use. ICANN needs to promote the adoption of IPv6 so as to limit the side effects of the exhaustion of the unique address space provided by IPv4.

A vast and new range of non-Latin, internationalized domain names may be registered, certainly at the second or lower levels in the domain-name hierarchy, and many will be proposed for the top level. Their diversity will create new challenges for the protection of users from confusing and potentially abusive registrations. New dispute resolution principles may be needed to deal with domain-name registrations and delegations of new top-level domains. The exposure of ASCII *punycode* strings in browsers or other applications may produce additional stresses in the intellectual property arena (for example **`xn--cocacola`**).

Digital signatures will play an increasingly important role in validating the assignment of domain names and Internet addresses, and new protocols are certain to be invented and their parameters recorded by the IANA. Infrastructure for the management of digital certificates or other authentication mechanisms will be needed to realize the value of the DNSSEC concept.

More generally, the multilayer architecture of the Internet shows vulnerabilities of various kinds that demand redress. Attacks against the DNS root servers, name resolvers, and general name servers at all levels must be mitigated.

Some of the components of the DNS are actually used to exacerbate the effects of *Denial-of-Service* (DoS) attacks. Although ICANN does not have responsibility for developing the Domain Name technology, it can use its visibility and area of responsibility to highlight the need for increased security measures for the protection of the technical infrastructure of the Internet and to facilitate its implementation where ICANN has a direct involvement in its operation.

An increasing number of mobile devices will become Internet-enabled, as will appliances of all kinds. Access speeds will increase, enabling many new applications and enhancing older ones. All of this activity will contribute to increasing reliance on the Internet for a wide range of functions by an increasingly larger user population. Electronic commerce will continue to expand, placing high priority on the stable, secure, and reliable operation of all aspects of the Internet, including those within ICANN's purview.

Although some of these aspects of the evolution of the Internet will be of direct concern to ICANN, the ICANN organization and processes will need to pay attention to additional matters as well. The business processes that sustain the management of the Internet address space and domain names will almost certainly need to adapt to account for new applications. Some of these applications will monetize various aspects of the Internet in unexpected and innovative ways that will challenge existing policy and procedures. It will be extremely important for ICANN to evolve and strengthen its implementation of multistakeholder policy development. The interests of a wide range of entities must be balanced in the process.

Although adherence to a set of technical standards has allowed millions of component networks and systems to interwork on the Internet, it is also the case that many varying business models have sustained their operation. The richness and diversity of these models is one of the reasons that the Internet has proved to be so resilient in many dimensions. ICANN's policy-development processes need to account for an informed understanding of the economics of these varying business models and the ways in which ICANN policy may affect them.

On the Domain Name side, the development of market-savvy rules of operation for operators will be essential. ICANN needs to assure compliance with policies developed through the ICANN consensus process to establish confidence in the policy processes and their execution. Clear rules for the creation of new *Top Level Domains* (TLDs) of all kinds must be adopted and enforced.

The roles of registrars, registries, wholesale registry operators, root-server operators, regional Internet address registries, governments, and standards and technical research and development bodies, among others, need to be characterized so as to set expectations and permit the establishment of practical working relationships. The documentation of best practices will be beneficial, especially where the introduction of the Internet is new.

In matters of public policy—including but not limited to public safety, security, privacy, law enforcement, conduct of electronic commerce, protection of digital property, and freedom of speech—broad and international agreements may be needed if the Internet is to serve as a useful, global infrastructure. Many of these matters lie outside the formal purview of ICANN, but some ICANN policies and resulting operational practices will contribute to the global framework for life online. ICANN must seek to contribute to public confidence in the Internet and the processes that govern its operation. It cannot accomplish this objective alone. The coordinated and cooperative efforts of many distinct entities will be essential to achieving this goal. At the same time, ICANN must protect its processes from capture or abuse by interests that are inimical to the openness and accessibility of the Internet for everyone.

A Collective Goal

As of this writing, only about 1.2 billion people around the world use the Internet. Over the course of the next decade, that number could conceivably quintuple to 6 billion, and users will be depending on ICANN, among many others, to do its part to make the Internet a productive infrastructure that invites and facilitates innovation and serves as a platform for egalitarian access to information. It should be a platform that amplifies voices that might otherwise never be heard and creates equal opportunities for increasing the wealth of nations and their citizens.

ICANN's foundation has been well and truly fashioned. It is the work of many heads and hands. It represents a long and sometimes difficult journey that has called for personal sacrifices from many colleagues and bravery from others. It has demanded long-term commitments, long hours, days, months, and years. It has called upon many to transform passion and zeal into constructive and lasting compromises. ICANN has earned its place in the Internet universe. To those who now guide its path into the future comes the challenge to fashion an enduring institution on this solid foundation. I am confident that this goal is not only attainable but now also necessary. The opportunity is there: make it so.

For Further Reading

[1] **http://www.iana.org**

[2] Vint Cerf, "I Remember IANA," *The Internet Protocol Journal*, Volume 1, No. 3, December 1998. Also published as RFC 2468, October 1998.

[3] **http://www.icann.org/committees/evol-reform/**

[4] **http://alac.icann.org/**

[5] Miek Gieben, "DNSSEC: The Protocol, Deployment, and a Bit of Development," *The Internet Protocol Journal*, Volume 7, No. 2, June 2004.

[6] **http://icann.org/topics/idn/**

[7] **http://icann.org/udrp/**

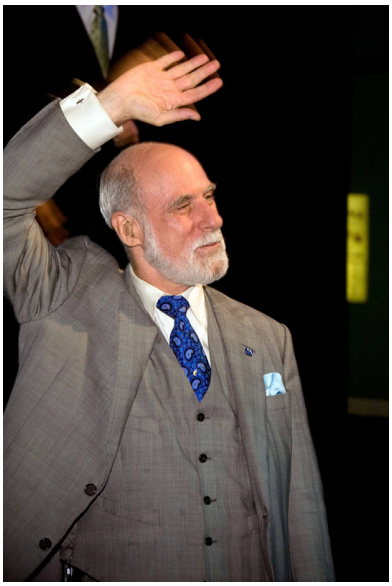
[8] **http://www.itu.int/wsis/index.html**

[9] **http://gac.icann.org/web/index.shtml**

[10] Daniel Karrenberg, Gerard Ross, Paul Wilson, and Leslie Nobile, "Development of the Regional Internet Registry System," The Internet Protocol Journal, Volume 4, No. 4, December 2001.

[11] **http://nro.org/**

[12] **http://www.root-servers.org/**



Photographer: Vanessa Stump

VINTON G. CERF is vice president and chief Internet evangelist for Google. In this role, he is responsible for identifying new enabling technologies to support the development of advanced Internet-based products and services from Google. He is also an active public face for Google in the Internet world. Cerf is the former senior vice president of Technology Strategy for MCI. In this role, he helped guide corporate strategy development from a technical perspective. Previously, he served as MCI's senior vice president of Architecture and Technology, leading a team of architects and engineers to design advanced networking frameworks, including Internet-based solutions for delivering a combination of data, information, voice, and video services for business and consumer use.

Widely known as one of the "Fathers of the Internet," Cerf is the co-designer of the TCP/IP protocols and the architecture of the Internet. In December 1997, President Clinton presented the U.S. National Medal of Technology to Cerf and his colleague, Robert E. Kahn, for founding and developing the Internet. Kahn and Cerf were named the recipients of the ACM Alan M. Turing Award, sometimes called the "Nobel Prize of Computer Science," in 2004 for their work on the Internet protocols. In November 2005, President George Bush awarded Cerf and Kahn the Presidential Medal of Freedom for their work. The medal is the highest civilian award given by the United States to its citizens.

Prior to rejoining MCI in 1994, Cerf was vice president of the Corporation for National Research Initiatives (CNRI). As vice president of MCI Digital Information Services from 1982 to 1986, he led the engineering of MCI Mail, the first commercial e-mail service to be connected to the Internet.

During his tenure from 1976 to 1982 with the U.S. Department of Defense Advanced Research Projects Agency (DARPA), Cerf played a key role leading the development of Internet and Internet-related packet-data and security technologies.

Vint was seated on the ICANN Board of Directors at the 1999 annual meeting, having been selected by the Protocol Supporting Organization. He was then selected by the nominating committee for a term on the board of directors that ran from June 2003 through the 2004 annual meeting. At the end of that term, he was selected by the 2004 nominating committee to an additional term, which ran from the end of the 2004 annual meeting through the conclusion of the ICANN annual meeting in 2007. He served as founding president of the Internet Society from 1992 to 1995, and in 1999 served a term as chairman of the board. In addition, Cerf is honorary chairman of the IPv6 Forum, dedicated to raising awareness and speeding introduction of the new Internet Protocol. Cerf served as a member of the U.S. Presidential Information Technology Advisory Committee (PITAC) from 1997 to 2001 and serves on several national, state, and industry committees focused on cyber security. Cerf sits on the board of directors for the Endowment for Excellence in Education, Avanex Corporation, and the ClearSight Systems Corporation. Cerf is a Fellow of the IEEE, ACM, and American Association for the Advancement of Science, the American Academy of Arts and Sciences, the International Engineering Consortium, the Computer History Museum, and the National Academy of Engineering.

Cerf is a recipient of numerous awards and commendations in connection with his work on the Internet, including the Marconi Fellowship, Charles Stark Draper Award of the National Academy of Engineering, the Prince of Asturias Award for science and technology, the National Medal of Science from Tunisia, the Alexander Graham Bell Award presented by the Alexander Graham Bell Association for the Deaf, the NEC Computer and Communications Prize, the Silver Medal of the International Telecommunications Union, the IEEE Alexander Graham Bell Medal, the IEEE Koji Kobayashi Award, the ACM Software and Systems Award, the ACM SIGCOMM Award, the Computer and Communications Industries Association Industry Legend Award, installation in the Inventors Hall of Fame, the Yuri Rubinsky Web Award, the Kilby Award, the Yankee Group/Interop/Network World Lifetime Achievement Award, the George R. Stibitz Award, the Werner Wolter Award, the Andrew Saks Engineering Award, the IEEE Third Millennium Medal, the Computerworld/Smithsonian Leadership Award, the J.D. Edwards Leadership Award for Collaboration, the World Institute on Disability Annual Award, and the Library of Congress Bicentennial Living Legend medal. In December 1994, People magazine identified Cerf as one of that year's "25 Most Intriguing People."

In addition to his work on behalf of MCI and the Internet, Cerf has served as a technical advisor to production for the "Gene Roddenberry's Earth: Final Conflict" television series and made a special guest appearance on the program in May 1998. Cerf has appeared on television programs NextWave with Leonard Nimoy and on World Business Review with Alexander Haig and Caspar Weinberger. He is also a distinguished visiting scientist at the Jet Propulsion Laboratory, where he is working on the design of an interplanetary Internet.

Cerf holds a Bachelor of Science degree in Mathematics from Stanford University and Master of Science and Ph.D. degrees in Computer Science from UCLA. He also holds honorary doctorate degrees from the Swiss Federal Institute of Technology (ETH), Zurich; Luleå University of Technology, Sweden; University of the Balearic Islands, Palma; Capitol College, Maryland; Gettysburg College, Pennsylvania; George Mason University, Virginia; Rovira i Virgili University, Tarragona, Spain; Rensselaer Polytechnic Institute, Troy, New York; the University of Twente, Enschede, The Netherlands; Brooklyn Polytechnic; and the Beijing University of Posts and Telecommunications.

Cerf's personal interests include fine wine, gourmet cooking, and science fiction. Cerf and his wife Sigrid were married in 1966 and have two sons, David and Bennett.

E-mail: **vint@google.com**

Remembering Itojun: The IPv6 Samurai

by Bob Hinden, Nokia

"Itojun" (Dr. Junichiro Hagino) passed away on October 29, 2007. He was 37 years old. Memorial events were held in Tokyo in November and in Vancouver at the IETF meeting in December.

Itojun was an active participant in the IETF and a member of the IAB from 2003 to 2005. He worked as a Senior Researcher at the *Internet Initiative Japan* (IIJ) and was a member of the board of the *Widely Integrated Distributed Environment* (WIDE) project. He was a strong supporter of open standards development and open software, working as a core researcher at the KAME project, a joint effort of six companies in Japan to provide a free stack of IPv6, IPsec, and Mobile IPv6 for BSD variants, from 1998 to 2006.

Itojun was totally dedicated to the development and deployment of IPv6. Most of his work was centered around building a much larger worldwide Internet based on IPv6. He was simply the "IPv6 Samurai."

Photographer: Diane Bruce

Quotes from Internet Colleagues

Steve Deering: "Those of us who got to know Itojun through his work in the Internet Engineering Task Force have lost a dear friend and much-admired colleague. From the day he arrived at his first IETF meeting, he won the respect of all in the way most honored by Internet engineers: by helping to build consensus based on running code. Moreover, he provided the best possible example of collaboration, generosity, and leadership, making not only extraordinary technological contributions but also many friends and a better world. His untimely passing is a huge loss to all who knew him, and to all those who will never have that chance."

Randy Bush: "An open heart, a big soul, and very kind and patient. A very special person. He wrote a lot of great code and got great joy from doing so."

Marc Blanchet: "Itojun adopted the Samurai's philosophy in his life: *Bushido,* which consists of values such as Honesty, Justice, Courtesy, Heroic Courage, Honor, Compassion, Sincerity, Duty, and Loyalty. Very difficult to achieve, he encompassed all these. Moreover, he was always available to help, anyone, without judging. His intelligence, his competency, and his dedication has inspired a generation of network engineers for the project he took as a mission: IPv6. Many computers in the world now run his code. My family always enjoyed meeting Itojun. He was always interested in sharing his knowledge with my children, even with the French-to-Japanese-through-English language barrier. Itojun, it was an honor to know you and to meet you. You will always be a source of inspiration to me, to my family, and to many network engineers in the world. We miss you."

Rod Van Meter: "I didn't know Itojun very well; I met him for the first time about five years ago at an IPv6 meeting in the Silicon Valley, once or twice in between, and then spent three days at the WIDE Camp this past September co-supervising (with Bill Manning, Brad Huffaker, and Kenji Saito) a group of students trying to establish long-term goals for WIDE in the area of naming. Itojun was gentle but insistent with students, a good mentor. That was the last time I saw him. Go in peace, Itojun."

Joel Jaeggli: "He cared more for the people who were going to use the code and the product of his and our labor than anyone would have had a right to expect. The Itojun that I know, our friend, has been taken from us, but we'll be the beneficiary of the fact that he cared, for decades."

Itojun IPv6 Fund

Itojun's family has expressed sincere appreciation to all who attended the memorial and funeral services. His family has set up a memorial fund in Itojun's name under the directorship of the IETF/Internet Society. The fund will be used to award an R&D grant to a person who has contributed to the deployment and further advancement of IPv6. ISOC has set up an e-mail address to accept commitments for the *Itojun IPv6 Fund*. The address is: **itojun-fund@isoc.org**

The procedure for making contributions is being developed; if you wish to contribute now, please send a note to the e-mail address describing the amount you want to contribute (and in what currency), and ISOC will collect the funds.

ROBERT HINDEN is a Nokia Fellow at Nokia and is located in Mountain View, California, USA. He has been involved in the Internet since it was a research project at ARPA. He developed one of the first TCP/IP implementations and his team at Bolt, Beranek, and Newman, Inc. built and operated the routers that formed the early Internet backbone. He was co-recipient of the 2008 IEEE Internet Award "For pioneering work in the development of the first Internet routers." He has been active in the IETF since 1985 and is the author of 35 RFCs. He was recently appointed to a position on the IETF Administrative Oversight Committee (IAOC) and co-chairs the IPv6 Maintenance (6man) working group. Prior to this he served on the Internet Architecture Board (IAB), was Area Director for Routing in the Internet Engineering Steering group from 1987 to 1994, and chaired the IPv6, Virtual Router Redundancy Protocol, Simple Internet Protocol Plus, IP over ATM, and Open Routing working groups. Hinden is also a member of the RFC Editorial Board. He holds a B.S.E.E. and a M.S. in Computer Science from Union College, Schenectady, New York. E-mail: **bob.hinden@nokia.com**

Book Review

Network Routing

Network Routing: Algorithms, Protocols, and Architectures, by Deepankar Medhi and Karthikeyan Ramasamy, Morgan Kaufmann Publishers, ISBN-13:978-0120885886, 2007,
`http://www.NetworkRouting.net`

Routing is a fundamental architectural component of any network, and in this book the authors examine in detail the routing technologies of the Internet and the *Public Switched Telephone Network* (PSTN).

Organization

The book is divided into five parts, with an additional advanced section provided on CDROM. The first part examines the fundamentals of routing technology, looking in detail at the basic approaches of distance-vector and link-state routing. The second part looks at the routing protocols used in the Internet today, as well as Traffic Engineering. The third part addresses routing in the PSTN, examining the SS7 signaling protocol and the overall architecture of the PSTN. The next part explores the internal architecture of routers, address-lookup algorithms, and packet-classification techniques. Finally, the authors consider topics encompassed in the so-called "Next-Generation Network," including *Quality-of-Service Routing, Multiprotocol Label Switching* (MPLS), and *Voice over IP* (VoIP). The advanced-topic section includes a more detailed examination of packet-switching approaches, scheduling, and conditioning. This book is positioned as a graduate-level text, and each chapter is accompanied by exercises that review the material.

The book covers a broad range of material: each topic has been the subject of entire books. The level of detail in the book varies considerably. In some instances, such as in the area of IP Traffic Engineering, it presents a highly detailed mathematical analysis of aspects of the topic, whereas in other instances, such as in the treatment of the *Border Gateway Protocol* (BGP), the material appears to be obviously condensed. I was expecting a little more use of algorithms to illustrate routing concepts, and found at times the mathematical analysis to be unhelpful in terms of understanding the underlying problem space being described.

Comparison

In this area of Internet routing, any publication is inevitably compared to Radia Perlmann's book *Interconnections: Bridges, Routers, Switches and Internetworking Protocols,* and this book is no exception. To my mind it falls a little short of this rather demanding standard. Radia spends some time discussing the underlying rationale as to why a particular technology was devised for a given problem space, and also discusses the strengths and limitations of the technology in various areas of application.

In *Network Routing* the authors limit their approach to a description of the technology by looking at packet payloads and protocol interactions and numerous deployment scenarios that illustrate the features of this particular routing technology. The consideration of choices made in the development of the protocol, and the consequent implications of such design choices, are missing in such a treatment, and the reader is often left wondering why a routing protocol has chosen to support certain functions but not others.

I found this to be a very ambitious book, because it appears to position itself both as a reference publication on routing technologies and architecture and also on the description of routing protocols, while at the same time wanting to encompass the role of a course text. This goal could have been attainable if the book had chosen a tighter focus, but the all-encompassing approach that led to the inclusion of considerations of the PSTN topics makes the outcome less than fully satisfying.

Recommended

However, the book manages to bring together the basic topics in routing in both the Internet and the PSTN, and it not only includes a good description of the routing technologies in use today, but also looks at some of the advanced topics in routing today. I found the major strength of the book in its role as a graduate-course text, where there is sufficient description of the topic to lead into further reading of current research papers and more-detailed technical material. Although the book has some shortcomings, I'd certainly recommend it as a suitable addition to the shelves of any professional in the area of Internet routing technologies and architecture.

—Geoff Huston
gih@apnic.net

The Author Responds:

I thank Geoff Huston for writing a well-thought-out review; in general, this review is fair. This book was certainly an ambitious project. I wanted to do it as I've investigated various routing protocols for almost two decades—and many people I talked to thought that it would be useful to have such a book. In fact, Dave Clark, when he read the original book proposal, wrote "It is ambitious—there may be issues of how much depth they can get on all these topics in one book," but felt that "...the approach is distinctive and very valuable. So I support the idea." As can be observed from the book, the depth on different topics remained a major trade-off we pondered without making the book go over 1000 pages (with 140 pages on CD-ROM it came pretty close).

There were a few "design" decisions I deliberately made in organizing and writing this book. One of them was based on years of teaching and interacting with industry folks: I decided to divide materials broadly on "how and why" away from "what;" this approach is somewhat surprising, but people's learning style seems to fall into these two categories (certainly there are overlaps). Therefore, details on "how and why" of different protocols went into Chapter 3 (and for algorithms into Chapter 2), while details on "what" went with chapters on specific protocols such as OSPF or BGP. Similarly, I also separated out the topic of "how" routing in the global Internet works and is organized (such as public exchange points) from the chapter on BGP. Secondly, we separated math parts from non-math parts—this way, those who are interested, for example, in detailed Traffic Engineering modeling can read the relevant chapters. Others may skip them and read just the first couple of overview sections; it should be noted that math-oriented chapters are generally organized from simple concepts to difficult concepts. Thirdly, we covered address lookup, packet filtering and classification, and router architectures separately because they can be read independently; Karthik brought his wealth of experience in writing these chapters.

I want to take this opportunity to respond to a few of Geoff's comments:

1. "...expecting a little more use of algorithms to illustrate routing concepts." I suspect that Geoff didn't think that Chapters 2 and 3 covered enough, although these chapters included details illustrative of distance-vector protocols, link-state protocols, path-vector protocols (and their pitfalls), and so on. As stated previously, by design of the book, illustrative examples of routing concepts were separated from specific protocols so that readers can read different portions of the material according to their interests. As an indicator to the reader, each chapter starts with a brief "reading guideline" (which is a unique feature of this book) that states how the material is organized and its relation to other chapters or sections in the rest of the book.
2. "The consideration of choices made in the development of the protocol, and the consequent implications of such design choices are missing in such a treatment." We did indeed cover these aspects in many instances. For example, the book covers why, for I-BGP scalability, the route reflector or the confederation approach are needed; why route flap damping was developed; why ROUTE-REFRESH was added; what MPLS was trying to solve that IP-only couldn't do at that time; the need for age with Sequence Number field in link-state protocols; what led to the development of dynamic call routing from hierarchical routing in the PSTN, and so on.

That said, I did not include certain discussions because some choices on protocols have been based on personality clashes and "camps;" I felt that this is not easily explainable in many instances—trying to do so would require quite a bit of discussion, and could potentially divert from the main focus of the book. For example, I explained why the route reflector or confederation approach was needed for I-BGP scalability, but I didn't discuss why both route reflector and confederation approaches were developed simultaneously when both convey the same idea conceptually.

3. "... the all-encompassing approach that led to the inclusion of considerations of the PSTN topics into this book make the outcome less than fully satisfying." I included routing in the PSTN because of its historic context, and particularly to make readers aware of the evolution from hierarchical routing to dynamic routing and recent changes in routing due to Local Number Portability—these lessons are important ones to learn for anyone interested in routing or designing future routing protocols. Secondly, many concepts in MPLS/GMPLS have parallels in the PSTN, thus certain aspects in MPLS/GMPLS are easily explainable if a reader is familiar with PSTN details. We therefore felt it was appropriate to include all this material in one place. Furthermore, control- and data-path separation in GMPLS is strikingly similar to separation of signaling in PSTN through SS7 from actual voice communication. Thus, lessons learned from failure propagation from SS7 to voice paths are relevant lessons to be aware of for anyone involved in deploying GMPLS-based networks. Lastly, to discuss VoIP routing, it is critical to tie into PSTN because in the real operational environment PSTN-Internet interworking for VoIP routing is expected to remain prevalent for years to come.

Finally, the "barrier to entry" in learning about routing is very high, especially for entry-level professionals—I've attempted to position the book as both a text and a reference for professionals. Thus, I very much appreciated Geoff's concluding comment "... as a suitable addition to the shelves of any professional in the area of Internet routing technologies and architecture."

—Deep Medhi
dmedhi@umkc.edu

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

Nii Quaynor Receives 2007 Postel Service Award

The *Internet Society* (ISOC) has awarded pioneering Internet engineer Nii Quaynor the prestigious *Jonathan B. Postel Service Award* for 2007 for his leadership in advancing Internet technology in Africa and galvanizing technologists to improve Internet access and capabilities throughout the continent. ISOC presented the award, including a $20,000 [USD] honorarium, during the 70th meeting of the *Internet Engineering Task Force* (IETF) in Vancouver, BC, Canada.

"Dr. Quaynor has selflessly pioneered Internet development and expansion throughout Africa for nearly two decades, enabling profound advances in information access, education, healthcare and commerce for African countries and their citizens," said ISOC president Lynn St. Amour. "Today, Dr. Quaynor continues to champion not just technological advances but also African involvement in Internet standards, processes and deployments, discussion on Internet policies and regulations, and ensuring African interests are well-represented globally. He has shaped a community of Africans who share his vision and reflect the dedication shown by Jon Postel."

"I am humbled by the award and what Jon Postel represents to our community in Africa. Jon Postel's efforts and the global view he maintained on the operation of the *Domain Name System* and the numbering services assured that Africa would share in the Internet growth and early. I thank the Internet Society for the recognition and am very pleased to be associated with Jon's memorial," said Dr. Nii Quaynor. "We will work to develop more African engineers to meet the fast network growth needs of the region, being a late starter, and to join the technical policy processes. Our overall objective is to strengthen education and research in network technologies in Africa."

The annual ISOC award is named after Dr. Jonathan B. Postel to commemorate his extraordinary stewardship exercised throughout his thirty-year career in networking. Between 1971 and 1998, Postel managed, nurtured and transformed the RFC series of notes, which encompasses the technical specifications and recommendations for the Internet and was created by Steve Crocker in 1969 as a part of his work on the ARPANET, the forerunner of today's Internet. Postel was a founding member of the Internet Architecture Board and the first individual member of the Internet Society, where he also served as a trustee until his untimely death.

Dr. Quaynor is chairman of *Network Computer Systems* (NCS) Ghana.COM and a professor of computer science at University of Cape-Coast, Ghana. He is also the convener of the *African Network Operators Group* (AfNOG), a network technology transfer institution since 2000 and the founding chairman of AfriNIC, the African numbers registry.

Dr. Quaynor began his pioneering Internet work in Africa in 1993 when he returned to his home country of Ghana to establish the first Internet Service operated by NCS in West Africa. At NCS, he and his team worked on the early development of the Internet in Africa. Today, there are more than 43 million Internet users in Africa.

Prior to NCS, Dr. Quaynor worked with Digital Equipment Corporation in the United States from 1977 till 1992. In 1979, he established the Computer Science department at the University of Cape Coast, Ghana. Dr. Quaynor graduated from Dartmouth College in 1972 with B.A (Engineering Science) and received a Ph.D. (Computer Science) in distributed systems in 1977 from State University of New York at Stony Brook.

The Jonathan B. Postel Service Award was established by the Internet Society to honor those who, like Postel, have made outstanding contributions in service to the data communications community. The award is focused on sustained and substantial technical contributions, service to the community, and leadership. With respect to leadership, the nominating committee places particular emphasis on candidates who have supported and enabled others in addition to their own specific actions.

Previous recipients of the Postel Award include Jon himself (posthumously and accepted by his mother), Scott Bradner, Daniel Karrenberg, Stephen Wolff, Peter Kirstein, Phill Gross, Jun Murai, Bob Braden, and Joyce K. Reynolds. The award consists of an engraved crystal globe and $20,000 [USD]. This year's award is sponsored in part by Afilias Global Registry Services. For more information about ISOC, please visit: **`www.isoc.org`**

Steps Taken for Multilingual Internet

The *Internet Corporation for Assigned Names and Numbers* (ICANN), the *International Telecommunication Union* (ITU), and the *United Nations Educational, Scientific and Cultural Organization* (UNESCO) will collaborate on global efforts to forge universal standards towards building a multilingual cyberspace. The three agencies organized a workshop on this subject during the second *Internet Governance Forum* (IGF) which took place in Rio de Janeiro, Brazil from 12 to 15 November 2007.

The Internet is a key factor in developing a more inclusive and development-oriented information society, which stresses plurality and diversity instead of global uniformity. Multilingualism is a key concept to ensure cultural diversity and participation for all linguistic groups in cyberspace. There is growing concern that hundreds of local languages may be sidestepped, albeit unintentionally in the radical expansion of Internet communication and information. The *World Summit on the Information Society* (WSIS) recognized the importance attached to linguistic diversity and local content, with UNESCO given the responsibility to coordinate implementation of the *Summit Action Line*.

"The discussions at this multilingualism workshop—combined with our current evaluation of *Internationalized Domain Names* (IDNs)—are going to help ICANN keep moving toward full implementation of Internationalized Domain Names," said Dr Paul Twomey, ICANN's President and CEO. "ICANN is in the midst of the largest ever evaluation of IDNs at the top level."

Thanks to ICANN's evaluation of Internationalized Domain Names, Internet users around the globe can now access wiki pages (see **`http://idn.icann.org/`**) with the domain name **`example.test`** in the 11 test languages—Arabic, Persian, Chinese (simplified and traditional), Russian, Hindi, Greek, Korean, Yiddish, Japanese and Tamil. The wikis will allow Internet users to establish their own sub pages with their own names in their own language; one suggestion is: **`example.test/yourname`**

Domain Names, which are currently mainly limited to characters from the Latin or Roman scripts, are seen as an important element in enabling the multilingualization of the Internet, reflecting the diverse and growing language needs of all users. "ITU is fully committed to assist its membership in promoting the diversity of language scripts for domain names," said Dr Hamadoun Touré, Secretary-General of ITU. "This workshop represents an important opportunity to strengthen the need for cooperation with relevant organizations, such as UNESCO, the *World Intellectual Property Organization* (WIPO) and ICANN among others to ensure Internet use and advancement across language barriers."

The Plenipotentiary Conference of ITU, which took place in Antalya, Turkey in November 2006, recognized the need to make Internet content available in non-Latin based scripts. Internet users are more comfortable reading or browsing through texts in their own language and a multilingual Internet is essential to make it more widely accessible. The WSIS outcomes also focused on the commitment to work towards multilingualization of the Internet as part of a multilateral, transparent and democratic process involving governments and all stakeholders.

UNESCO, joined by both ITU and ICANN, seeks to convene all major stakeholders around the world towards an agreement on universal standards regarding language issues in cyberspace. Such issues are far broader than the single issue of IDNs as they extend to standards for fonts and character sets, text encoding, language implementations within major computer operating systems, content development tools, automatic translation software, and search engines across languages. Ultimately, equitable access to information can be only achieved if we resolve language barriers at the same time we build communications infrastructures and capacity building programs.

RIPE Community Resolution on IPv4 Depletion and Deployment of IPv6

During the RIPE 55 meeting in Amsterdam in October 2007, the RIPE community agreed to issue the following statement on IPv4 depletion and the deployment of IPv6:

"Growth and innovation on the Internet depends on the continued availability of IP address space. The remaining pool of unallocated IPv4 address space is likely to be fully allocated within two to four years. IPv6 provides the necessary address space for future growth. We therefore need to facilitate the wider deployment of IPv6 addresses.

While the existing IPv4 Internet will continue to function as it currently does, the deployment of IPv6 is necessary for the development of future IP networks.

The RIPE community has well-established, open and widely supported mechanisms for Internet resource management. The RIPE community is confident that its *Policy Development Process* meets and will continue to meet the needs of all Internet stakeholders through the period of IPv4 exhaustion and IPv6 deployment.

We recommend that service providers make their services available over IPv6. We urge those who will need significant new address resources to deploy IPv6. We encourage governments to play their part in the deployment of IPv6 and in particular to ensure that all citizens will be able to participate in the future information society. We urge that the widespread deployment of IPv6 be made a high priority by all stakeholders."

For more information, see: **http://ripe.net/ripe/**

Upcoming Events

The next *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will be held in Taipei, Taiwan from February 20th to 29th, 2008. As usual, this conference is co-located with an APNIC Open Policy Meeting. For more information about these events see: **http://www.apricot2008.net/** and **http://www.apnic.net/meetings/25/index.html**

The *Internet Engineering Task Force* (IETF) will meet in Philadelphia, Pennsylvania, March 9–14 and "somewhere in Europe" July 27–August 1. (The announcement of the exact location is expected soon). The final IETF meeting in 2008 will take place in Minneapolis, Minnesota, November 16–21. For more information see: **http://www.ietf.org/meetings/0mtg-sites.txt**

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in New Delhi, India, February 10–15, and in Paris, France, June 22–27. See: **http://icann.org/meetings/**

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

March 2008 Volume 11, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

The *Domain Name System* (DNS) was not designed to support anything beyond 7-bit ASCII characters. Thus my middle name, Jørgen, or my colleague's surname, Fältström, cannot be used in a domain name. In fact, even using such strings on the left side of the @-sign—or in the body of an e-mail message—is problematic. We often find ourselves ignoring this limitation, using either "Jorgen" and "Faltstrom" or in some cases the two-letter convention "Joergen" and "Faeltstroem." As Scandinavians, Mr. Fältström and I are relatively lucky in that our languages contain only three characters in addition to those that can be represented by 7-bit ASCII. This, of course, isn't true for such languages as Arabic, Chinese, Japanese, or Korean, to name just a few. The IETF, ICANN, and others have been working hard to design and deploy a system that will allow native characters to appear in the DNS. Our first article discusses these efforts, known collectively as *Internationalized Domain Names* (IDNs). Geoff Huston gives an overview of IDNs and describes the many technical and political challenges that must be overcome in order to deploy such a system.

Recent activities have focused much attention on IPv6 deployment. Experiments have been conducted at several major Internet events (NANOG, APRICOT, and IETF) to "turn off" IPv4 for a period of time to test connectivity and interoperability to the outside world. You can read more about these experiments in our "Fragments" section on page 41. Such experiments provide valuable information about what works and what doesn't, and several more IPv4 "outages" are planned for 2008 and beyond. At the same time, researchers have been looking at ways to scale the routing system of the Internet, regardless of IP protocol version. One such approach is the *Locator/Identifier Separation Protocol* (LISP), which Dave Meyer describes in our second article.

The next issue of *The Internet Protocol Journal*, to be published sometime in June 2008, will be our Tenth Anniversary issue. We would love to hear your reflections on the last ten years of this journal and about the Internet as a whole over the same time period. Send your Letters to the Editor to **ipj@cisco.com**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Internationalizing the Domain Name System

by Geoff Huston, APNIC

Considering the global reach of the Internet, internationalizing the network sounds like a tautology. Surely the Internet is already truly "international," isn't it? The Internet reaches around the globe to every country, doesn't it? And no matter where you may travel these days, an Internet café is just around the corner. How much more "international" can you get?

But maybe I'm just being too parochial here when I call it a tautology. I use a dialect of the English language, and all the characters I need are contained in the *Western Latin* character set. Therefore, I avoid using a non-English language on the Internet; the only language I use on the Internet is English, and all the characters I need are encompassed in the ASCII character set. If I tried to use the Internet with a language that has a non-Latin character set and a different script, my experience would probably be different—and acutely frustrating. If my native language used a different script and a different text flow than English, I would probably give the Internet an extremely low score for ease of use. It is not as simple as managing glyph sets to represent the characters of the language; although it is relatively easy to present pictures of characters in a variety of fonts and scripts, using them in an intuitive and natural way in the context of the Internet becomes more challenging.

Mostly what is needed is good *localization,* or adapting the local computing environment to suit local linguistic needs. This environment may include support for additional character sets and additional language scripts, and perhaps altering the direction of text flow, or even the entire layout of the information.

For example, Japanese is traditionally written in a format called *Tategaki.* In this format, the text flows in columns going from top to bottom, with columns ordered from right to left. Modern Japanese also uses another writing format, called *Yokogaki.* This writing format is identical to that of European languages such as English, where the text flows from left to right in successive rows from top to bottom.

Today, the left-to-right direction is dominant in Japanese *Kana,* Chinese characters, and Korean *Hangul* for horizontal writing. This change is due partly to the influence of English, and partly to the increased use of computerized typesetting and word-processing software, most of which does not directly support right-to-left layout of East Asian languages. It would appear that even *Yokogaki* is an outcome of the lack of capability of IT systems to correctly cope with localization.[1]

One topic, however, does not appear to have a compellingly obvious localization solution in this multilingual environment: the *Domain Name System* (DNS). The subtle difference here is that the DNS is the "glue" that binds all users' language symbols together, and performing localized adaptations to suit local language use needs is not enough. The DNS spans the entire network, so what works for me in the DNS must also work for you. What we need is a means to allow the use of all of these language symbols within the same system, or *internationalization*.

The DNS is the most prevalent means of initiating a network transaction, whether it is a *BitTorrent* session, the Web, e-mail, or any other form of network activity. But the DNS name string is not just an arbitrary string of characters. What you find in the DNS is most often a sequence of words or their abbreviations, and the words are generally English words, using characters drawn from a subset of the Latin character set. Perhaps unsurprisingly, some implementations of the DNS also assume that all DNS names must be constructed only from this ASCII character set, and these implementations are incapable of supporting a larger character repertoire. If you want to use a larger character set in order to represent various diacritics, such as acute and grave symbols, umlauts and similar marks, then the deployed DNS can be resistant to this use, and may provide incorrect responses to queries that include such characters. And if you want to use words drawn from languages that do not use the western script for their characters, such as Japanese or Thai, for example, then the DNS is highly resistant to this form of multilingual use.

Latin and Roman Alphabets

The default Latin alphabet is the Roman[2] alphabet, supplemented with G, J, U, W, Y, Z, and lowercase variants. Additional letters may be formed:

- As *ligatures,* as W was from VV, for example Æ (*ash*) from AE, *oethel* Œ from OE, *eszett* ß from ſz (long s + z), *engma* ŋ from NG, *ou* Ȣ from OU, Ñ from NN, or ä from ae
- By *diacritics,* such as Å, Č, and Ų
- As *digraphs,* such as fi and fl
- By modification, as J was from I, G from C, Ø from O, *eth* Ð from D, *yogh* Ȝ from G, or *schwa* Ə from E
- By borrowing from another alphabet entirely, as *thorn* Þ and *wynn* Ƿ were from Futhark (Runic)

Over the years we have done a reasonable job of at least displaying non-Latin-based scripts within many applications, and although at times it appears to represent a less-than-reasonable compromise, it is possible to enter non-Latin characters on computer keyboards. So it appears to be possible to customise a local computing environment to use a language other than English in a relatively natural way.

But what happens when we extend the scope to consider multilingual support in the wider world of the Internet?

Again the overall story is not all that bad. We can use non-Latin character scripts in e-mail, in all kinds of Web documents, and in a wide variety of network applications. We can tag content with a language context to allow display of the content in the correct language using the appropriate character sets and presentation glyphs. However, until recently, one area continued to stick steadfastly to its ASCII roots: the DNS. This article addresses DNS internationalization, or *Internationalized Domain Names* (IDNs).

What do we mean when we talk of "internationalizing the DNS"? It refers to an environment where English, and the Latin character set, is just one of many languages and scripts in use, and where a communication is initiated in one locale and then the language and presentation are preserved wherever the communication is received.

Terminology

The following terms are used in this article:

Language: A language uses characters drawn from a collection of scripts.

Script: A script is a collection of characters that are related in their use by a language.

Character: A character is a unit of a script.

Glyph: The presentation of a character within the style of a font is called a glyph.

Font: A font is a collection of glyphs encompassing a script character set that share a consistent presentation style.

Multiple languages can use a common script, and any locale or country may use many languages, reflecting the diversity of its population and the evolution of local dialects within communities.

It is also useful to remember the distinction between internationalization and localization. *Internationalization* is concerned with providing a common substrate that many—preferably all—languages and all users can use, whereas *localization* is concerned with the use of a particular language within a particular locale and within a defined user population. Unsurprisingly, the two concepts are often confused, particularly when true internationalization is often far more difficult to achieve than localization.

Internationalizing the DNS

The objective is the internationalization of the DNS, such that the DNS can support the union of all character sets while preserving the absence of ambiguity and uncertainty in terms of resolution of any individual DNS name. We need to describe all possible characters in all languages and allow their use in the DNS. So the starting point is the "universal character set," and that appears to be Unicode.

One of the basic building blocks for internationalization is a character set that is the effective union of all character sets. *Unicode*[3] is intended to be such a universal encoding of characters (and symbols) in the contexts of all scripts and all languages. The current version of the *Unicode Standard,* Version 5.0, contains 98,884 distinct coded graphic characters.

A sequence of Unicode code points can be represented in multiple ways by using different character encoding schemes in a *Unicode Transformation Format* (UTF). The most commonly used schemes are UTF-8 and UTF-16.

UTF-8 is a variable-length encoding using 8-bit words, meaning that different code points require different numbers of bytes. The larger the index number of a code point, the more bytes are required to represent it using UTF-8. For example, the first 127 Unicode code points, which correspond exactly to the values used by the ASCII character set (which maps only 127 characters), can be represented using only 8 bits in UTF-8, using the same 8-bit values as in ASCII. UTF-8 can require up to 32 bits to encode certain code points. A criticism of UTF-8 is that it "penalizes" certain scripts by requiring more bytes to represent their code points. The IETF has made UTF-8 its preferred default character encoding for internationalization of Internet application protocols.

UTF-16 is a variable-length character encoding using 16-bit words. Characters in the *Basic Multilingual Plane* are mapped into a single 16-bit word, with other characters mapped into a pair of 16-bit words.

UTF-32 is a fixed-length encoding that uses 32 bits for every code point. This encoding tends to make for a highly inefficient coding that is, generally, unnecessarily large, because most language uses of Unicode draw characters from the Basic Multilingual Plane, making the average code size 16 bits in UTF-16 as compared to the fixed-length 32 bits in UTF-32. For this reason UTF-32 is far less commonly used than UTF-8 and UTF-16.

But languages, which we humans change in various ways every day, are not always definitive in their use of characters, and Unicode has some weaknesses in terms of identifying a context of a script and a language for a given character sequence. The common approach to using Unicode encodings in application software is to use an associated "tag," allowing content to be tagged with a script and an encoding scheme. For example, a content tag might read: "This text has been encoded using the KOI-8 encoding of the CYRILLIC script."

Tagging allows for decoding of the encoded characters in the context of a given script and a given language. This decoding has been useful for e-mail or Web page content, but tagging breaks down in the context of the DNS. There is no natural space in DNS names to contain language and script tags, implying that attempting to support internationalization in the DNS has to head toward a "universal" character set and a "universal" language context. Another way of looking at this situation is that the DNS must use an implicit tag of "all characters and all languages."

The contexts of the use of DNS names have numerous additional artefacts. What about domain-name label separators? This "dot" between DNS "words," or a DNS label separator, is an ASCII period character. In some languages, such as Thai, for example, there is no natural use of such a label separator. In a similar vein, are URLs intended to be visible to end users? If so, then we may have to transform the punctuation components of the URL into the script of the language. Therefore, we may need to understand how to manage protocol strings, such as "http:" and separators such as the "/" character. To complete the integrity of the linguistic environment, these elements may also require local presentation transformations.

For example, the Thai alphabet uses 44 consonants and 15 basic vowel characters, which are horizontally placed, from left to right, with no intervening space, to form syllables, words, and sentences. Vowels associated with consonants are nonsequential: they can be located before, after, above, or below their associated consonant, or in a combination of these positions. The latter in particular causes problems for computer encoding and text rendering[4].

The DNS name string reads left to right, and not right to left or top to bottom as in other script and language cultures. How much of this string you can encode in the DNS and how much must be managed by the application is part of the problem here. Is the effort to internationalize the DNS with multiple languages restricted to the "words" of the DNS, leaving the implicit left-to-right ordering and the punctuation of the DNS unaltered? If so, how much of this ordering and punctuation is a poor compromise, in that these DNS conventions in such languages are not natural translations?

The Unicode UTF-8, UTF-16, and UTF-32 encodings all require an "8-bit clean" storage and transmission medium. Because "traditional" DNS domain names are representable with 7-bit ASCII characters, not all applications that process domain names preserve the status of the eighth bit; in other words, they are not 8-bit clean. This situation stimulated significant debate in the IETF's *IDN Working Group* and influenced the direction of the standards development into the area of application assistance: the group took a very conservative view of the capabilities of the DNS as a restricted ASCII code application.

Accordingly, we now see the DNS itself as a heavily restricted "language." The prudent use of the DNS specifies, in RFC 1035[5], a sequence of "words" (or "labels"), where each label conforms to the "*L*etter, *D*igit, *H*yphen" (LDH) restriction. Each DNS label must begin with a letter, restricted to the Latin character subset of "A" through "Z" and "a" through "z", followed by a sequence of letters, digits, or hyphens, with a trailing letter or digit, and no trailing hyphen. Furthermore, the case of the letter is not important to the DNS, so, within the DNS "a" is equivalent to "A", and so on, and all characters are encoded in monocase ASCII. The DNS uses a left-to-right ordering of these labels, with the ASCII period as the label delimiter. This restriction is often referred to as the *LDH Convention.*

The challenge posed with the effort of *internationalizing* the DNS is one of attempting to create a framework that allows Internet applications—and the DNS in particular—to be set in the user's own language in an entirely natural fashion, and yet allow the DNS to operate in a consistent and deterministic manner within its restricted "language." In other words, we all should be able to use browsers and e-mail systems using our own language and scripts, yet still be able to communicate naturally with others who may be using a different language interface.

The most direct way of stating the choice set of IDN design is that IDNs either change the "prudent use" of the deployed DNS into something quite different by permitting a richer character repertoire in all parts of the DNS, or IDNs change the applications that want to support a multilingual environment such that they have to perform some form of encoding transfer to map between a language string using Unicode characters and an "equivalent" string using the restricted DNS LDH character-set repertoire. It appears that options other than these two lead us into fragmented DNS roots, and having already explored that particular concept in the past, not many of us want to return to that subject. So if we want to maintain a cohesive and unified symbol space for the DNS, then either the deployed DNS has to become 8-bit clean, or applications have to do the work and present to the DNS an encoded form of the Unicode sequences that conform to the restricted DNS character repertoire.

The IDN Framework

If you are an English language user with the ASCII character set, the DNS name you enter into the browser—or the domain part of an e-mail address—is almost the same string as the string that is passed to the DNS resolver to resolve into an address (the difference is the conversion of the characters into monocase). If you want to send a mail message, you might send it to **`user@example.com`**, for example, and the domain name part of this address, **`example.com`**, is the string used to query the DNS for an *MX Resource Record* in order to establish how to actually deliver the message.

But what if you want to use a domain name that is expressed in another language? What if the e-mail address is **`user@`**記念**`.com`**? The problem here is that this domain name cannot be "naturally" expressed in the restricted syntax of the DNS, and although this domain name may have a perfectly reasonable Unicode code sequence, this encoded sequence is not a strict LDH sequence, nor is it case-insensitive (whatever "case" may mean in an arbitrary non-Latin script). It is here that IDNs depart from the traditional view of the DNS and use a hybrid approach to the task of mapping these language strings into network addresses.

The IDN Working Group of the IETF was formed in 2000 with the goal of developing standards to internationalize domain names. The working group's charter was to specify a set of requirements and develop IETF standards-track protocols to allow use of a broader range of characters in domain names. The outcome of this effort was the *IDN in Applications* (IDNA) framework, published as RFCs 3454, 3490, 3491, and 3492.[6,7,8,9]

Rather than attempting to expand the character repertoire of the DNS itself, the IDN working group used an *ASCII Compatible Encoding* (ACE) to encode the binary data of Unicode strings that would make up IDNs into an ASCII character encoding. The concept is similar to the Base64 encoding used by the *Multipurpose Internet Mail Extension* (MIME) e-mail standards, but whereas Base64 uses 64 characters from ASCII, including uppercase and lowercase, the ACE approach requires the smaller DNS-constrained LDH subset of ASCII.

The working group examined various ACE algorithms in its efforts to converge to a single standard (because different encoding algorithms have different compression goals and yields) and encode the data using slightly different subsets of ASCII. Most proposals specified a prefix to the ACE coding to tag the fact that this string was, in fact, an encoded Unicode string. The IETF adopted *punycode* as its standard IDN ACE[9]. Punycode was chosen for its efficient encoding compression properties that produce short ACE strings. For example, the domain name of 記念**`.com`** encodes with punycode to **`xn--h7tw15g.com`**.

IDN in Applications

Although an ASCII-compatible encoding of Unicode characters allows representation of an IDN in a form that will probably not be corrupted by the deployed DNS infrastructure on the Internet, an ACE alone is not a full solution. The IDN approach also needs to specify how and where the ACE should be applied.

The overall approach to IDNs is relatively straightforward. In IDN the application has a critical role to play. The application takes a domain name that is expressed in a particular language using a particular script—and potentially in a particular character and word order that is related to that language—and produces an ASCII-compatible LDH-encoded version of this DNS name. Equally, when presenting a DNS string to the user, the application should take the LDH-encoded DNS name and transform it to a presentation sequence of glyphs that correspond to the original string in the original script.

It is critical that all applications perform this encoding and decoding function correctly, deterministically, and uniformly. In fact, this capability is critical to the entire IDN framework.

The basic shift in the DNS semantics that IDNs bring to the DNS is that the actual name itself is no longer in the DNS. An encoded version of the canonical name form sits in the DNS, and applications need to perform the canonical name transformation, as well as the mapping between the Unicode character string and the encoded DNS character string. So we need to agree on what are the "canonical" forms of name strings in every language. We also need to agree on the encoding method, and our various applications must have precise equivalents of these canonical name and encoding algorithms, or the symbolic consistency of the DNS will fail. The problem here is that the DNS does not perform approximate matches or return a set of possible answers to a query. The DNS is a deterministic system that performs a precise match on the query in order to generate a response. The implication here is that if we want the same IDN character sequence to map to the same network response in all cases and all contexts, then all applications must perform precisely the same operations on the character sequence in order to generate the ACE-equivalent label sequence.

RFC 3454[6] defines a presentation layer in IDN-aware applications that is responsible for the punycode ACE encoding and decoding. This new layer in the application architecture is responsible for encoding any internationalized input in domain names into punycode format before the corresponding LDH encoded domain name is passed to the DNS for resolution. This presentation layer is also responsible for decoding the punycode format in IDNs and rendering the appropriate glyphs for the user.

It is a matter of personal perspective whether this solution is an elegant one or it simply shifts an unresolved problem from one area of the IETF to another. The IDNA approach assumes that it is easier to upgrade applications to all behave consistently in interpreting IDNs than it is to change the underlying DNS infrastructure to be 8-bit clean in a manner that would support direct use of Unicode code points in the DNS.

The Presentation Layer Transform for IDNs

The objective here is to define a reliable and deterministic algorithm that takes a Unicode string in a given language and produces a DNS string as expressed in the LDH character repertoire. This algorithm should not provide a unique 1:1 mapping, but should group "equivalent" Unicode strings, where "equivalence" is defined in the context of the language of use, into the same DNS LDH string. Any reverse mapping from the DNS LDH string into the Unicode string should deterministically select the single "canonical" string from the group of possible IDN strings.

Stringprep

The first part of the presentation layer transform is to take the original Unicode string and apply numerous transformations to it to produce a "regular" or "canonical" form of the IDN string. This form of the string is then transformed using the punycode ACE into an encoded DNS string form. The generic name of this process is, in IDN language, "stringprep,"[6] and the particular profile of transformations used in IDNAs is termed "nameprep."[8]

This transform of a Unicode string into a canonical format is based on the observation that many languages have a variety of ways to display the same text and a variety of ways to enter the same text. Although we humans are unconcerned about this concept of expressing an idea in multiple ways, the DNS is an exact equivalence match operation and it cannot tolerate imprecision. So how can the DNS tell that two text strings are intended to be identical, even though their Unicode strings are different? The IDN approach is to transform the string so that all equivalent strings are mapped to the same canonical form, or "stringprep" the string. The stringprep specification is not a complete algorithm, and it requires a "profile" that describes the applicability of the profile, the character repertoire (at the time of writing RFC 3454, it was Unicode 3.2, although the Unicode Consortium has subsequently released Unicode Version 4.0, 4.1, and 5.0), mapping tables normalization, and prohibited output characters.

Mapping

In converting from a string to a *normal,* or canonical, form, the first step is to map each character into its *normalized* equivalent, using a mapping table. This table is conventionally used to map characters to their lowercase equivalent value to ensure that the DNS string comparison is case-insensitive.

Other characters are removed from the string by using this mapping operation because their presence or absence in the string does not affect the outcome of a string-equivalence operation, such as characters that affect glyph choice and placement, but without semantic meaning.

The mapping function will create monocase (specifically lowercase) outcomes and also will eliminate non-significant code points (such as, for example, the Unicode code point 1806; MONGOLIAN TODO SOFT HYPHEN or the Unicode code point 200B; ZERO WIDTH SPACE, if you really wanted to know what a non-significant code point was).

Normalization

Numerous languages use different character sequences for the same meaning. Characters may appear the *same* in presentation format as a glyph sequence, yet have *different* underlying code points. This may be associated with variables ways of combining diacritics, or using canonical code points, or using compatibility characters, and, in some language contexts, performing character reordering. For example, the character Ä can be represented by a single Unicode code point 00C4; LATIN CAPITAL A WITH DIARESIS. Another valid representation of this character is the code point 0041; LATIN CAPITAL LETTER A followed by the separate code point 0398; COMBINING DIARESIS.

The intent of normalization is to ensure that every class of character sequences that are equivalent in the context of a language is translated into a single canonical, consistent format. This consistency of format allows the equivalence operator to perform at the character level using direct comparison without additional language-dependent equivalence operations.

Languages in daily use are not rigid structures, and human use patterns of languages change. Normalization is no more than a best-effort process to detect equivalences in a rigid, rule-managed manner, and it may not always produce predictable outcomes. This unpredictability can be a problem with regard to namespace collisions in the DNS, because it does not increase the confidence level of the DNS as a deterministic exact-match information-retrieval system. IDNs introduce some forms of name approximation into the DNS environment, and the DNS is extremely ill-suited to the related "fuzzy-search" techniques that accompany such approximations.

Filtering Prohibited Characters

The last phase in string preparation is removal of prohibited characters, including the various Unicode white-space code points, control code points and joiners, private-use code points, and other code points used as surrogates or tags.

Right-to-Left Characters

As an option for a particular stringprep profile, you can perform a check for right-to-left displayed characters, and if any are found, make sure that the whole string satisfies the requirements for bidirectional strings. The Unicode standard has an extensive discussion of how to reorder glyphs for display when dealing with bidirectional text such as Arabic or Hebrew. All Unicode text is stored in logical order as distinct from the display order.

Nameprep: A Stringprep Profile for the DNS

The nameprep profile[8] specifies stringprep for internationalized domain names, specifying a character repertoire (in this case the specification references Unicode 3.2) and a profile of mappings, normalization (form "KC"), prohibited characters, and bidirectional character handling. The outcome is that two-character sequences can be considered equivalent in the context of IDNs if, by following the sequences of operations defined by the nameprep profile, the resultant sequences of Unicode code points are identical. These code point sequences are the "canonical" forms of names that the DNS uses.

The Punycode ASCII-Compatible Encoding

The next step in the processing of IDN names by the application is to transform this canonical form of the Unicode name string into a LDH-equivalent string using an ACE. The algorithm used, *punycode,* uses a highly efficient encoding, attempting to limit the extent to which Unicode sequences become extended-length ACE strings.

The algorithm first divides the input code points into a set of "basic" code points that require no further encoding, and the set of "extended" code points. The algorithm takes the basic code points and reproduces this sequence in the encoded string: the "literal portion" of the string. A delimiter is then added to the string. This delimiter is a basic code point that does not occur in the remainder of the string. The extended code points are then added to the string as a series of integers expressed through an encoding into the basic (LDH) code set.

These additions of the extended code points are done primarily in the order of their Unicode values, and secondarily in the order in which they occur in the string. The encoding of the code point and its insertion position is done by using a difference, or offset, encoding, so that sequences of clustered code points, such as would be found in a single language, encode efficiently.

For example, the German language string *bücher* uses basic codes for all characters except the *ü* character. The punycode algorithm copies all the basic codes, followed by a "-". The value and position of the *ü* insertion now has to follow.

The encoded form for *ü* (code 252) is at the position between the first and second basic characters. Using the punycode[10] algorithm gives a delta code of 745, a value that can be expressed in base 35 as (21 x 35) + 10. This code point and the position information are expressed in base 35 notation as (10,22,1), or in reverse notation, with the encoding **kva**. So the punycode encoding of *bücher* is **bcher-kva**. The internationalized domain-name format prepends the string **xn--** to the punycode string, resulting in the encoded IDN domain-name form of **xn--bcher-kva**.

IDNS and Our Assumptions About the DNS

At this stage it should be evident that we have the code points for characters drawn from all languages, and the means to create canonical forms of various words and express them in an encoded form that the DNS can resolve.

However, there is more to IDNs than the encoding algorithm. Although a massive number of discrete code points exist in the realm of Unicode, all these distinct characters are not necessarily displayed in unique ways. Indeed, given a relatively finite range of glyphs, the same glyph can display numerous discrete code points.

The often-quoted example with IDNs and name confusion is the name **paypal**. What is the difference between **www.paypal.com** and **www.paypal.com**? There is a subtle difference in the first "a" character, where the second domain name has replaced the Latin *a* with the Cyrillic *a*. Did you spot the difference? Of course not. These *homoglyphs* are cases where the underlying domain names are distinct, yet their appearance is indistinguishable. In the first case the domain name **www.paypal.com** is resolved in the DNS with the query string **www.paypal.com**, yet in the second case the query string **www.paypal.com** is translated by the application to the DNS query string **www.xn--pypal-4ve.com**. How can you tell one case from the other?

This example is by no means a unique case in the IDN realm. The reports "Unicode Security Considerations" (Unicode Technical Report 36) and "Unicode Security Mechanisms" (Unicode Technical Report 39) provide many more examples of postnormalization homographs.

There is no clear and unique relationship between characters and glyphs. Cyrillic, Latin, and Greek share numerous common glyphs. Glyphs may change their shape depending on the character sequence, multiple characters may produce a single glyph, such as the character pair *f l* being displayed as the single glyph *fl*, and a single character may generate multiple glyphs.

Homoglyphs extend beyond a conventional set of characters and include syntax elements as well. For example, the Unicode point 0244 FRACTION SLASH is often displayed using the slash glyph, allowing URLs of the form **http://a.com/e.com**. Despite its appearance, this is not a reference to **a.com** with a locator suffix of **e.com**, but is a reference to the domain **a.com/e.com**.

The basic response is that if you maintain IDN integrity at the application level, then the user just cannot tell. The punycode transform of **www.paypal.com** into **www.xn--pypal-4ve.com** is intended to be a secret between the application and the DNS, because this ASCII-encoded form is simply meaningless to the user. But if this encoded form remains invisible to the user, how can the user detect that the two identically presented name strings are indeed different? Sadly, the only true "security" we have in the DNS is the "look" of the DNS name that is presented to the user, and the user typically works on the principle that if the presented DNS string looks like the real thing, then it must be the real thing.

When this homoglyph problem was first exposed, the response from many browser implementations was to turn off all IDN support in their browser. The next response was to deliberately expose the punycode version of the URL in the browser address bar, so that directing the browser to **http://www.paypal.com** would display in the address bar the URL value of **http://www.xn--pypal-4ve.com**.

The distinction between the two equivalently displayed names was then visible to the user, but the downside was that we were back to displaying ASCII names again, and in this case ASCII versions of punycode-encoded names. If trying to "read" Base64 was difficult, then the displaying—and understanding—of displayed punycode names is surely equally as difficult, if not more so. The encoded names can be completely devoid of any form of useful association or meaning. Although the distinction between ASCII and Cyrillic may be evident by overt differences in their ASCII-encoded names, what happens when the homoglyph occurs across two non-Latin languages? The punycode strings are different, but which string is the "intended" one? Did you mean **http://xn--21bm4l.com** or **http://xn--q2buub.com** when you enter a Hindi script URL?

Using ASCII as the fall-back to resolve name confusion in response to the problem of ambiguities in non-ASCII script names appears to be a nonsensical solution. We appear to be back to guessing games in the DNS again, unfortunately, and particularly impossible guessing games at that.

These days most popular browsers display the glyphs, rather than the ASCII punycode, but once more we are back to the homoglyph problem.

If the intention in the IDN effort was to preserve the deterministic property of DNS resolution, such that a DNS query can be phrased deterministically and not have the query degenerate into a search term or require the application of fuzzy logic to complete the query, then we are not quite there yet.

The underlying observation is that languages are indeed human-use systems. They can be tricky, and they invariably use what appear to be rules in strange and inconsistent ways. They are also resistant to automated processing and the application of rigid rule sets. The canonical name forms that are produced by nameprep-like procedures are not comprehensive, nor does it appear that such a rigidly defined rule-driven system can produce the desired outcomes in all possible linguistic situations. And if the intention of the IDN effort was to create a completely "natural" environment using a language environment other than English and a display environment that is not reliant on ASCII and ASCII glyphs, while preserving all the other properties of the DNS, then the outcome does not appear to match our original IDN expectations.

The underlying weakness here is the implicit assumption that in the DNS "what you see is what you get," and that two DNS names that look identical are indeed references to the same name, and when resolved in the DNS produce precisely the same resolution outcome. When you broaden the repertoire of appearances of the DNS, such that the entire set of glyphs can be used in the DNS, then the mapping from glyph to underlying code point is not unique. Any effort to undertake such a mapping needs additional context in the form of a language and script context. But the DNS does not carry such a context, making the task of maintaining uniqueness and determinism of DNS name translation essentially impossible if we also want to maintain the property that it is the appearance, or presentation format, of DNS names to the user that is the foundation stone of the integrity of our trust in the DNS.

Some concerns still remain in this space, including the inclusion of various forms of character codes that are in effect invisible. In addition, homoglyphs could be better managed by using a refined definition of IDN labels that lists which Unicode code points can be used in the context of IDNs, excluding all others. It would be helpful if confusing and non-reversible character mappings were removed from the IDN space, including the consistent treatment of ligatures and diacritics, refining the treatment of right-to-left and left-to-right scripts, and removing the dependency on a particular version of the Unicode standard. This effort is under way in the IETF in the context of revisions to the IDNA specification documents.

IDNS, TLDs, and the Politics of the DNS

So why is there a very active debate, particularly within ICANN-related forums, about putting IDN codes into the root of the DNS as alternative *top-level domains* (TLDs)?

I have seen two major lines of argument here; namely the argument that favors the existence of IDNs in all parts of the DNS, including the TLDs, and the argument that favors a more restricted view of IDNs in the root of the DNS that links their use to that of an existing (ASCII-based) DNS label in the TLD zone.

Apparently, those who favor the approach of using IDNs in the top-level zone as just another DNS label see this as a natural extension of adding punycode-encoded name entries into lower levels of the DNS. Why should the root of the DNS be any different, in terms of allowing IDNs? Why should a non-Latin script user of the Internet have to enter the TLD code in its ASCII text form, while entering the remainder of the string in a local language? And in right-to-left scripts, where does this awkward ASCII appendage sit when a user attempts to enter it into an application?

Surely, goes the argument, the more natural approach is to allow any DNS name to be wholly expressible in the user's language, implying that all parts of the DNS should be able to carry native language-encoded DNS names. After all, コンピュータは予約する**.jp** looks wrong as a monolingual domain name. What is that **.jp** appendage doing there in that DNS name? Surely a Japanese user should not have to resort to an ASCII English abbreviation to enter in the country code for Japan, when 日本 is obviously more "natural" in the context of a Japanese user using Japanese script. If we had punycode TLDs then, goes the line of argument, users could enter the entire domain name in their language and have the punycode encoding happen across the entire name string, and then successfully perform a DNS lookup on the punycode equivalent. This way the user would enter the Japanese character sequence: コンピュータは予約する**.**日本 and have the application translate this entry to the DNS string **xn--88j0bve5g9-bxg1ewerdw490b930f.xn--wgv71a**. For this process to work in its entirety uniformly and consistently, the name **xn--wgv71a** needs to be a TLD name.

We can always take this thought process one step further and question the ASCII string **http** and the punctuation symbols **://** for precisely the same reason, but I have not heard (yet) calls for multilingual equivalents of protocol identifier codes. The multilingual presentation of these elements remains firmly in the provenance of the application, rather than attempting to alter the protocol identifiers in the relevant standards.

The line of argument also encompasses the implicit threat that if the root of the DNS does not embrace TLDs as expressed in the language of the Internet's users, then language communities will break away from a single DNS root and meet their linguistic community's requirements in their own DNS hierarchy. Admitting such encoded tags into the DNS root is the least problematic, including the consequence of inactivity, which is cited as being tantamount to condoning the complete fragmentation of the Internet's symbol set.

Of course having an entirely new TLD name in an IDN name format does not solve all of the potential problems with IDNs. How can a user tell what domain names are in the ASCII top level, and what are in the "equivalent" IDN-encoded TLDs? Are any two name spaces that refer to the same underlying name concept equivalent? Is **xn--88j0bve5g9bxg1ewerdw490b930f** appropriately a subdomain of **.jp**, or a subdomain of **xn--wgv71a**? Should the two domains be tightly synchronized with respect to their zone content and represent the same underlying token set, or should they be independent offerings to the marketplace, and allow registrants and the end-user base make implicit choices here? In other words, should the pair of domain names, namely **xn--88j0bve5g9bxg1ewerdw490b930f.xn--wgv71a** and **xn--88j0bve5g9bxg1ewerdw490b930f.jp**, reference precisely the same DNS zone, or should they be allowed to compete, and each find their own "natural" level of market support based on decoupled TLD names of **.jp** and **.xn--wgv71a**?

What does the term *equivalence* really imply here? Is equivalence something as loose as the relationship between **.com** and **.biz**, namely being different abbreviations of words that reflect similar concepts with different name-space populations that reflect market diversity and a competitive supply industry? Or is equivalence a much tighter binding in that equivalent names share precisely the same subdomain name set, and a registration in one of these equivalence names is in effect a name registration across the entire equivalence set?

Even this subject is not readily resolvable given our various interpretations of *equivalence*. In theory, the DNS root zone is populated by ISO two-letter country codes and numerous "generic" TLDs. Under what basis, and under what authority, is **xn--wgv71a** considered an "equivalent" of the ISO 3166 two-letter country code JP? Are we falling into the trap once again of making up the rules as we go along? Is the distinction between **.com** and **.biz** apparent only in English? And why should this distinction apply only to non-Latin character sets? Surely it makes more sense for a native German language speaker to refer to commercial entities as *kommerze,* and the abbreviated TLD name as **.kom**? When we say "multilingual" are we in fact ignoring "multilingual" and looking exclusively at "multiscript"?

Let's put aside the somewhat difficult concept of name equivalence for a second, and assume that this equivalence problem is solved. Also suppose that we want tight coupling across equivalence sets of names.

In other words, what we want is that a name registered in any of the elements of the equivalent domain-name set in all scripts is, in effect, registered in all the equivalent DNS zones. The question is: how should it be implemented in the DNS? One approach that could support tight synchronization of equivalence is to use the DNAME record[11] to create these TLD name aliases for their ASCII equivalents, thereby allowing a single name registration to be resolvable using a root name expressed in any of the linguistic equivalents of the original TLD name. The DNAME entry for all but the "canonical" element of the equivalence set effectively translates all queries to a query on the canonical name. The positive aspects of such an approach is uniformity across linguistic equivalents of the TLD name form—a single name delegation in a TLD domain becomes a name within all the linguistic equivalents of the TLD name without any further delegation or registration required.

Using DNAME as a tool to support sets of equivalent names in the DNS is still in the early stages. The limited experience so far with DNAME indicates that CNAME synthesis places load back on the name servers that would otherwise not be there, and the combination of this synthetic record and DNSSEC starts to get very unwieldy. Also, the IETF is reviewing the DNAME specification with the intention to remove the requirement to perform CNAME synthesis. All of these factors may explain why there is no immediate desire to place DNAMEs in the DNS root zone.

Different interpretations of equivalence in IDN names are possible. The use of DNAMEs as aliases for existing TLDs in effect "locks up" IDNs into the hands of the incumbent TLD name-registry operators. Part of the IDN debate, is, as usual, a debate over the generic TLD registry operators and the associated perception of incumbent monopolies. An alternative approach is to associate a single registrar with each IDN variant of the same generic TLD, allowing a form of "competition" between the various registrars. From the perspective of a coherent symbol space where the same symbol, expressed in any language script, resolves in the same fashion, such independent registries are not overly consistent with such a model of registry diversity in a multilingual environment. In this case such an artifice of IDN "competition" may well do more harm than good for Internet users.

It appears that another line of argument is that the DNS top-level name space is very conservatively managed, and new entries into this space are not made lightly. There are concerns of stability of operation, of attempting to conserve a coherent namespace, and the ever-present consideration that if we manage to "break" the DNS root zone it would be an irrevocable act.

This line of argument recognizes the very hazy nature of name equivalence in a multilingual environment and is based on the proposition that the DNS is incapable of representing such imprecision with any utility. The DNS is not a search engine, and the DNS does not handle imprecision at all well. Again, goes the argument, if this is the case then can we push this problem back to the application rather than trying to bend the DNS? If an application is capable of translating, say, 日本 into **xn--wgv71a**, and considering that the TLD name space is relatively small, it appears that having the application performing a further translation of this intermediate form punycode string into the ASCII string **jp** is not a particularly challenging form of table lookup. In such a model no new TLD aliases or equivalences are required in the root zone of the DNS. If we are prepared to pass the execution of the presentation layer of the DNS to the application layer to perform, then why not also ask this same presentation layer to perform the step of further mapping the punycode ACE equivalents of the TLDs to the actual ASCII TLDs, using some richer language context that the application may be aware of that is not viable strictly within the confines of the DNS?

So, with respect to the question of whether IDN TLDS should be loaded into the DNS at all, and, if so, whether they should represent an opportunity for further diversity in name supply or be constrained to be aligned to existing names, and precisely how name equivalence is to be interpreted in this context, then it appears that ICANN has managed to place itself in a challenging situation. In not making a decision, those with an interest in having diverse IDN TLDs appear to derive some pleasure in pointing out that the political origins of ICANN and its strong linguistic bias to English are influencing it to ignore non-English language use and non-English language users of the Internet. Where dramatic statements are called for, such statements often use terms such as "cultural imperialism" to illustrate the nature of the linguistic insult. The case has been made repeatedly, in support of IDN TLDs, that an overwhelming majority of Internet users and commercial activity of the Internet is in languages other than native English, and the imposition of ASCII labels on the DNS is an unnatural imposition on the overwhelming majority of Internet users.

On the other hand, most decisions to permit some form of entry in the DNS are generally seen as irrevocable, and building a DNS that is littered with the legacy of various non-enduring name technologies and poor ad hoc decisions to address a particular concern or problem without any context of a longer-term framework seems also to represent a step along a direction leading to a heavily littered and fragmented Internet where, ultimately, users cannot communicate with each other.

What about global interoperability and the Internet? Should we just take the easy answer and simply give up on the entire concept? Well of course not! But, taking a narrower perspective, are IDNs simply not viable in the DNS? I would suggest that not only is this question one that was overtaken by events years ago, but even if we want to reconsider it now, then the answer remains that any users using their local language and local script should have an equally "natural" experience. IDNs are a necessary and valuable component of the symbol space of any global communications system, and the Internet is no exception. However, we also should recognize that we do need combinations of both localization and globalization, and that we are voicing some pretty tough objectives. Is the IDNA approach enough? Is our assumption that an unaltered DNS with application-encoded name strings represents a rich enough platform to preserve the essential properties of the DNS while allowing true multilingual use of the DNS? On the other hand, taking a pragmatic view of the topic, is what we have with IDNA enough for us to work on, and is the alternative of reengineering the entire fabric of the DNS into an 8-bit clean system just not a viable option?

I suspect that the framework of IDNA is now the technology for IDNs for the Internet, and we simply have to move on from here and deliberately take the stance of understanding the space from users' perspectives when we look at the policy concerns of IDNs. The salient questions from such perspectives include: "What is the "natural" thing to do?" and "What causes a user the least amount of surprise?" Because in this world, what works for the user is what works for the Internet as a whole.

Further IDN News

IDNs are by no means completed work. Development continues in the Unicode forum on elaboration of character sets, and there are further proposals in the IETF to continue a complementary standards activity of refining the IDN documents.

In February 2008 the *Applications Area* of the IETF announced a proposal for further work on IDNs. The proposal has noted that the existing RFC documents are tied to version 3.2 of Unicode, while the Unicode Consortium has released version 5.0.0.

The proposed work is to consider revision of the IDN documents to untie the Internet specifications that define validity based on Unicode properties from specific versions of Unicode using algorithms. It is also proposed that these updates study revision of bi-directional algorithms, and to permit the use of some scripts that were inadvertently excluded by the original Internet specification.

This is not intended to be a major rewrite of the IDN approach, and, in particular, IDNs will continue to use the **xn--** prefix, the same Punycode ASCII-compatible encoding, and the bidirectional algorithm is intended to follow the same design as presently specified.

Further Reading

It is possible to reference an overwhelming amount of commentary on this topic, so I have deliberately kept this list of further reading on the topic of IDNs relatively brief:

[A] John Klensin, "Internationalizing Top-Level Domain Names: Another Look," ISOC Member Briefing, September 2004, **http://www.isoc.org/briefings/018/**

[B] John Klensin, "National and Local Characters for DNS Top Level Domain (TLD) Names," RFC 4185, October 2005.

[C] Papers submitted to the ICANN IDN TLD workshop, held in November 2005: **http://www.icann.org/announcements/announcement-17nov05.htm**

[D] Internet Architecture Board, "Review and Recommendations for Internationalized Domain Names (IDNs)," RFC 4690, September 2006.

[E] "ICANN's IDN Roadmap Announcement—Progress and Future," **http://www.icann.org/announcements/announcement-1-01nov06.htm**

[F] "An Important Step Toward the Implementation of IDN Top-Level Domains: New Versions of IDNA Protocol Revision Proposals Posted," **http://www.icann.org/announcements/announcement-26nov07.htm**

[G] ICANN's IDN Evaluation Gateway. Eleven new internationalized domains representing the name **example.test** entirely in scripts other than the Latin characters: **http://idn.icann.org/**

References

[1] **http://en.wikipedia.org/wiki/Horizontal_and_vertical_writing_in_East_Asian_scripts**

[2] **http://en.wikipedia.org/wiki/Roman_script**

[3] **http://unicode.org**

[4] **http://www.omniglot.com/writing/thai.htm**

[5] Mockapetris, P., "Domain Names—Implementation and Specification," RFC 1035, November 1987.

[6] Hoffman, P., and Blanchet, M., "Preparation of Internationalized Strings ("stringprep")," RFC 3454, December 2002.

[7] Hoffman, P., Fältström, P., and Costello, A., "Internationalizing Domain Names in Applications (IDNA)," RFC 3490, March 2003.

[8] Hoffman, P., and Blanchet, M., "Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)," RFC 3491, March 2003.

[9] Costello, A., "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)," RFC 3492, March 2003.

[10] **http://en.wikipedia.org/wiki/Punycode**

[11] Crawford, M., "Non-Terminal DNS Name Redirection," RFC 2672, August 1999.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. The author of numerous Internet-related books, he is currently the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of the Internet Society from 1992 until 2001. E-mail: **gih@apnic.net**

The Locator Identifier Separation Protocol (LISP)

by David Meyer, Cisco Systems

The *Internet Architecture Board's* (IAB)'s October 2006 *Routing and Addressing Workshop*[8] renewed interest in the design of a scalable routing and addressing architecture for the Internet. Many concerns prompted this renewed interest, including the scalability of the routing system and the impending exhaustion of the IPv4 address space. Since the IAB workshop, several proposals have emerged that attempt to address the concerns expressed both at the workshop and in other forums[7,9,12,13,14]. All of these proposals are based on a common concept: the separation of locator and identifier in the numbering of Internet devices, often termed the "Loc/ID split." This article focuses on one proposal for implementing this concept: the *Locator/Identifier Separation Protocol* (LISP)[3].

The basic idea behind the Loc/ID split is that the current Internet routing and addressing architecture combines two functions: *Routing Locators* (RLOCs), which describe how a device is attached to the network, and *Endpoint Identifiers* (EIDs), which define "who" the device is, in a single numbering space, the IP address. Proponents of the Loc/ID split argue that this "overloading" of functions makes it virtually impossible to build an efficient routing system without forcing unacceptable constraints on end-system use of addresses. Splitting these functions apart by using different numbering spaces for EIDs and RLOCs yields several advantages, including improved scalability of the routing system through greater aggregation of RLOCs. To achieve this aggregation, we must allocate RLOCs in a way that is congruent with the topology of the network ("Rekhter's Law"). Today's "provider-allocated" IP address space is an example of such an allocation scheme. EIDs, on the other hand, are typically allocated along organizational boundaries. Because the network topology and organizational hierarchies are rarely congruent, it is difficult (if not impossible) to make a single numbering space efficiently serve both purposes without imposing unacceptable constraints (such as requiring renumbering upon provider changes) on the use of that space.

LISP, as a specific instance of the Loc/ID split, aims to decouple location and identity. This decoupling will facilitate improved aggregation of the RLOC space, implement persistent identity in the EID space, and, in some cases, increase the security and efficiency of network mobility.

Implementing the Locator/ID Separation

There are two basic approaches to implementing the Loc/ID split: *map-and-encap* and *address rewriting*. Each is briefly discussed in the following sections.

Map-and-encap

In the map-and-encap scheme (genenerally considered to have evolved from Bob Hinden's ENCAPS protocol[24]), when a source sends a packet to the EID of a destination outside of the source domain, the packet traverses the domain infrastructure to a border router (or other border element). The border router maps the destination EID to a RLOC that corresponds to an entry point in the destination domain (hence an EID-to-RLOC mapping system is needed; proposals are discussed later in the article). This phase is the "map" phase of map-and-encap. The border router then encapsulates the packet and sets the destination address to the RLOC returned by the mapping infrastructure (if any; it may be statically configured as well). This phase is the "encap" phase of the map-and-encap model.

Thus map-and-encap works by appending a new header to the existing packet; the "inner-header" source and destination addresses are EIDs, and the "outer-header" source and destination addresses are in most cases RLOCs. When an encapsulated packet arrives at the destination border router, the router decapsulates the packet and sends it on to its destination. Note that this process suggests that EIDs may need to be routable in some scope (likely scoped to the domain).

Map-and-encap schemes have the desirable property that they do not in general require host changes or changes to the core routing infrastructure. In addition, map-and-encap schemes work with both IPv4 and IPv6, and retain the original source address (a feature that is useful in various filtering scenarios). Controversy remains, however, as to whether or not the encapsulation overhead of map-and-encap schemes is problematic; opinions exist on both sides of this topic (see, for example, [18]).

Address Rewriting

The basic idea behind the address-rewriting schemes, originally proposed by Dave Clark and later by Mike O'Dell in his 8+8/GSE specification[11], is to take advantage of the 128-bit IPv6 address and use the top 64 bits as the routing locator ("Routing Goop," or RG), and the lower 64 bits as the endpoint identifier (hence rewriting works only for IPv6). In this scheme, when a host emits a packet destined for another domain, the source address contains its identifier (frequently a IEEE MAC address) in the lower 64 bits, and a special value (meaning unspecified) in the RG. The destination address contains the fully specified destination address (RG and EID).

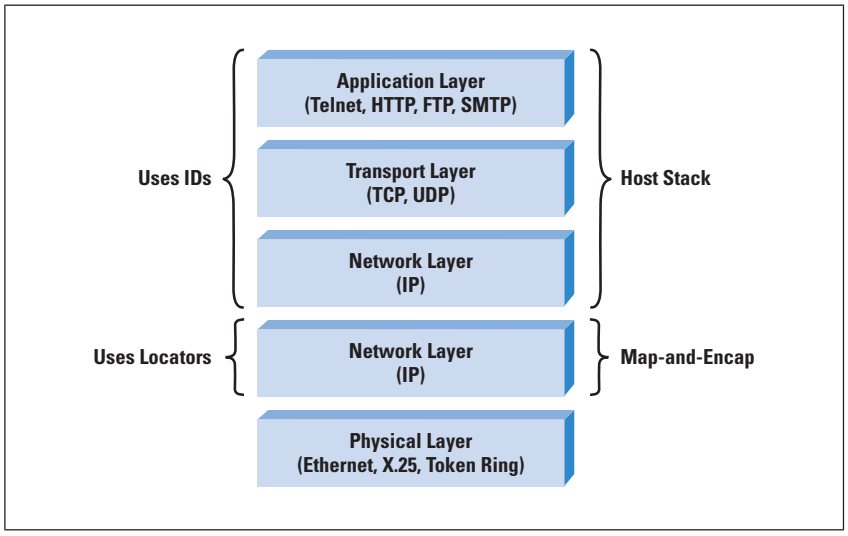
When a packet destined for a remote domain arrives at the local domain egress router, the source RG is filled in (forming a full 128-bit address), and the packet is routed to the remote domain. On ingress to the remote domain, the destination RG is rewritten with the unspecified value, ensuring that the host does not know what its RG is.

This process, in theory, would enable the ease of renumbering that would be required to maintain congruence between prefix assignment and physical network topology that is required for the kind of "aggressive" renumbering envisioned in the 8+8/GSE specification.

The Locator/Identifier Separation Protocol (LISP)

LISP is designed to be a simple, incremental, network-based map-and-encap protocol that implements separation of Internet addresses into EIDs and RLOCs. Because LISP is a map-and-encap protocol, it requires no changes to host stacks and no major changes to existing database infrastructures. It is designed to be implemented in a relatively small number of routers. LISP is also an instance of what is architecturally called a "jack-up," because the existing network layer is "jacked up" and a new network layer is inserted below it (the term "jacked up" is attributed to Noel Chiappa). The LISP jack-up is depicted in Figure 1.

Figure 1: LISP is a Jack-Up



The LISP design aims to improve site multihoming (for example, by controlling site ingress without complex protocols), improve *Internet Service Provider* (ISP) multihoming, decouple site addressing from provider addressing, and reduce the size and dynamic properties of the core routing tables.

The LISP data plane (the map-and-encap operation) and the LISP control plane (the EID-to-RLOC mapping system) are very modular. In particular, although the base LISP specification defines the format of messages to query the mapping system and to receive responses from that system, it makes no assumptions on the architecture of potential mapping systems. As a result, several mapping systems have been proposed[0,1,4,5,6,10].

LISP Network Elements

The LISP specification defines two network elements: The Egress Tunnel Router (ETR) and the Ingress Tunnel Router (ITR).

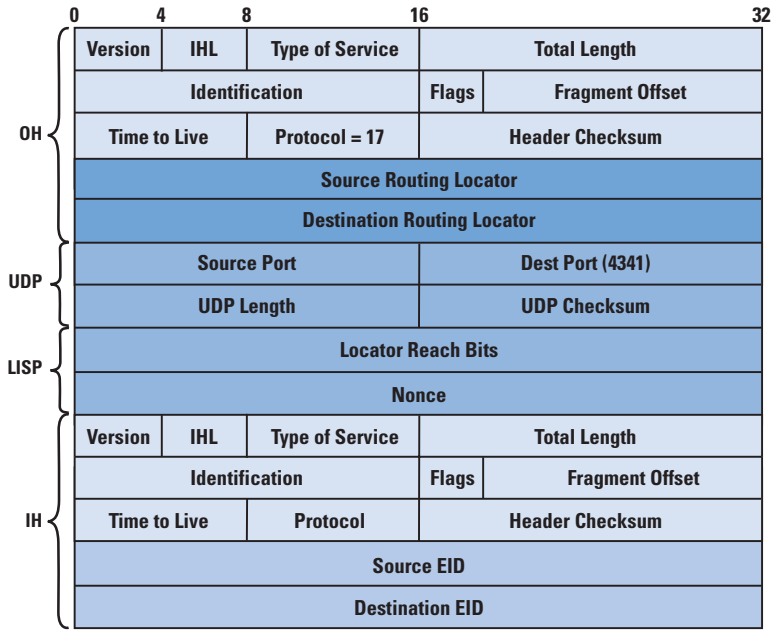
A LISP *Egress Tunnel Router* (ETR) receives LISP-encapsulated IP packets from the Internet on one side and sends decapsulated IP packets to site end systems on the other side. In particular, an ETR accepts an IP packet where the destination address in the "outer" IP header is one of its own RLOCs. The router strips the "outer" header and forwards the packet based on the next IP header found.

A LISP *Ingress Tunnel Router* (ITR) accepts IP packets from site end systems on one side and sends LISP-encapsulated IP packets toward the Internet on the other side. In particular, an ITR accepts an IP packet with a single IP header (more precisely, an IP packet that does not contain a LISP header). The router treats this "inner" IP destination address as an EID and performs an EID-to-RLOC mapping lookup if necessary (that is, it does not already have an EID-to-RLOC mapping for the EID). The router then prepends an "outer" IP header with one of its globally routable RLOCs in the Source Address field and the result of the mapping lookup in the Destination Address field. Note that this destination RLOC may be an intermediate, proxy device that has better knowledge of the EID-to-RLOC mapping closest to the destination EID.

LISP Data-Plane Operation

When a host in a LISP-capable domain emits a packet, it puts its EID in the packet source address, and EID of the correspondent host in its destination address (note that hosts will typically look up EIDs in the *Domain Name System* [DNS]). If the destination of the packet is in another domain, the packet traverses the source domain infrastructure to one of its ITRs. The ITR maps destination EID to a RLOC that corresponds to an ETR that is either in the destination domain or a proxy for the destination domain (how this mapping is accomplished in LISP is discussed later in the article). The ITR then encapsulates the packet, setting the destination address to the RLOC of the ETR returned by the mapping infrastructure or by static configuration. Note that LISP is address family-agnostic and as such can be used with both IPv4 and IPv6 (or any other address family). Figure 2 depicts the LISP IPv4 in IPv4 encapsulation.

Figure 2: LISP Header Format



When the packet arrives at the destination ETR, it decapsulates the packet and sends it on to its destination. Again, note that this scenario implies that EIDs need to be routable in some scope (likely scoped to the domain).

As mentioned previously, the LISP specification defines three packet types designed to support an EID-to-RLOC mapping system. The first type of packet, the *Data Probe*, is a data packet that an ITR may send into the mapping system to probe for the mapping; the authoritative ETR responds to the ITR with a Map-Reply message when it receives such a data packet. Note that in this case the ETR detects that the packet is a Data Probe by noticing that the inner *Destination Address* (DA) was copied to the outer DA by the ITR, that is, the inner DA equals the outer DA and is an EID. The second type of LISP packet used to support the mapping system is the *Map Request*. An ITR may query the mapping system by sending a Map-Request message into the mapping system to request a particular EID-to-RLOC mapping. As in the Data Probe case, the authoritative ETR responds with a Map-Reply message.

The third type of LISP packet used to support the mapping system is the *Map Reply*. An ETR emits a Map Reply under two conditions. First, if the ETR receives a LISP-encapsulated packet in which the outer-header destination address is the same as that of the inner header, it knows that the packet is a Data Probe and can respond with a Map Reply to the source ITR. The ETR may also receive a Map Request, in which case it replies to the requesting ITR with the mapping.

LISP Control Plane

Both map-and-encap and address-rewriting models rely on an additional of level of indirection in the addressing architecture to make the routing system scale reasonably. Because packets are sourced with an EID in the Destination Address field and EIDs are not in general routable on the global Internet, the destination EID must be mapped to an RLOC in order to deliver the packet to another domain (that is, across the Internet). In the case of the map-and-encap schemes, it is a direct translation: an EID is mapped to a RLOC. The situation is subtly different for the rewriting schemes; in general such schemes must look up the entire destination address (usually proposed to reside in the DNS)[11,13], but must somehow determine the source RG when rewriting the source address at the domain border.

In either Loc/ID split model, an EID-to-RLOC mapping service is needed to make the system scale reasonably and to make it operationally viable. There are three important scale parameters to consider when architecting a mapping service: the rate of updates to the mapping database, the state of the mapping service required, and the latency incurred during database lookup. The scaling properties of the database are frequently characterized as a (*Rate* × *State*) problem (ignoring for the moment the subject of lookup latency); because most estimates put the size of the mapping database at $O(10^{10})$, the database update rate must be small (note that this situation is a primary reason that current mapping proposals do not incorporate reachability information into the mapping database). In addition, the choice of push vs. pull also affects latency: if you push the entire database close to the edge, you improve lookup latency at the cost of increased state; if you architect a service that requires a mapping request and you find an authoritative server for that mapping (that is, pull), you reduce state at the cost of increased lookup latency.

LISP-Alternative-Topology: A LISP Control Plane

The basic idea behind *LISP-Alternative-Topology* (LISP-ALT)[4] is to build an alternative logical topology for managing EID-to-RLOC mappings for LISP. This logical topology uses existing technology and tools, specifically the *Border Gateway Protocol* (BGP)[17] and its multiprotocol extension[15], along with the *Generic Routing Encapsulation* (GRE)[16] protocol to construct an overlay network of devices that advertise EID prefixes only.

As was the case for the LISP data plane, an important design goal of LISP-ALT is to minimize the number of changes to existing hardware and software that are required to deploy the mapping system. Therefore, LISP-ALT requires modifications to neither BGP nor GRE.

Note that LISP-ALT is a hybrid push/pull architecture. Aggregated EID prefixes are "pushed" among the LISP-ALT routers and, optionally, to ITRs (which may elect to receive the aggregated information, as opposed to simply using a default mapping). Specific EID-to-RLOC mappings are "pulled" by ITRs either by Map Requests or Data Probes, both of which are routed over the alternate topology and result in Map Replies being generated by ETRs.

The basic idea behind in LISP-ALT, then, is to use BGP running over a GRE overlay to build the reachability required to route Data Probes, Map Requests, and Map Replies over the alternate topology. The ALT *Routing Information Base* (RIB) comprises EID prefixes and associated next hops. The LISP-ALT routers talk *External BGP* (eBGP) to each other in order to propagate EID prefix update information, which is learned either over eBGP connections from the authoritative ETR or by configuration. ITRs may also eBGP peer with one or more LISP-ALT routers in order to route Data Probe packets or Map Requests.

In summary, the LISP-ALT uses BGP to propagate EID-prefix reachability information used by ITRs and ETRs to forward Map Requests, Map Replies, and Data Probes. This reachability is carried as IPv4 or IPv6 *Network Layer Reachability Information* (NLRI) without modification (because the EID space has the same syntax as IPv4 or IPv6). LISP-ALT routers eBGP peer with one another, forming the overlay network. A LISP-ALT router near the edge learns EID prefixes that originate with authoritative ETRs. In general then, LISP-ALT routers aggregate EID prefixes, and forward Data Probes, Map-Requests, and Map-Replies.

Threat Models and Mitigation

As in any Loc/ID split approach, a critical operation is the creation of locator-to-ID binding state that devices will use over time. In the case of LISP, the critical operation is the creation of EID-to-RLOC mappings in the ITR and the ETR. We can obtain these mappings in three ways:

- By using the information obtained from a LISP data packet
- By using the information contained in the Map-Reply message
- By using an EID-to-RLOC mapping database

LISP mitigates attacks on the first two techniques by including a *nonce* in the LISP header; the nonce is a 32-bit randomly generated number (generated by the source ITR) that is used to test route returnability.

More specifically, an ETR echoes the nonce back to the ITR in a Map-Reply message. That is, the nonce, combined with the ITR accepting only solicited Map Replies, provides a base level of authentication for Map Replies. Note however, that these techniques do not protect against man-in-the-middle attacks.

The LISP design assumes that many (if not most) security mechanisms are part of the mapping database service when using control-plane procedures for obtaining EID-to-RLOC mappings. *Denial-of-Service* (DoS) attack prevention, on the other hand, depends on the ability of an implementation to rate-limit Map Requests and Map Replies (in the control plane), as well as its ability to rate limit the number of data-triggered Map Replies (for example, in response to Data Probe packets).

Refer to [19] for a more detailed preliminary threat analysis for LISP.

LISP and Fast Endpoint Mobility

Fast endpoint mobility occurs when an endpoint moves relatively rapidly, changing its IP layer network attachment point, and maintenance of session continuity is a goal. Mobile IPv4[20] and Mobile IPv6[21,22,27] mechanisms can be used in this case; note however, that the interaction of Mobile IP with LISP needs further exploration. Refer to the LISP specification[3] for additional details.

In summary, the major problem introduced by a Loc/ID split scheme is that as an endpoint moves, changes to the mapping between its EID and a set of RLOCs for its new network location may be required. When this change is added to the overhead of mobile IP binding updates, some packets might be delayed or dropped. In general, the problem is controlling the update rate (that is, the [*Rate* × *State*] product described previously), and is an area of ongoing research.

Multicast

A multicast group address, as defined in the original Internet architecture, is an identifier of a grouping of topologically independent receiver host locations. The address encoding itself does not determine the location of the receiver(s). The multicast routing protocol and the network-based state the protocol creates determine the location of the receivers.

In the LISP context, a multicast group address is both an EID and a RLOC. As such, no specific action is necessary for destination addresses; a group address that appears in an inner IP header (built by a source host) is used as the destination EID by an ITR as a destination address when it LISP-encapsulates the packet (that is, the ITR uses the same group address as the destination RLOC).

The source RLOC, as is usually the case, is the ITR IP address (that is, one of its RLOCs).

At the receiving side, *Protocol Independent Multicast* (PIM)[23] has to translate the source-address Join/Prune messages from RLOCs to EIDs when multicast packets are forwarded by the ETR. However, in contrast to the unicast case (where a Map Request is sent by the ITR at forwarding time), a Map Request can be sent when the multicast tree is being built.

Putting It All Together: A Day in the Life of a LISP Packet

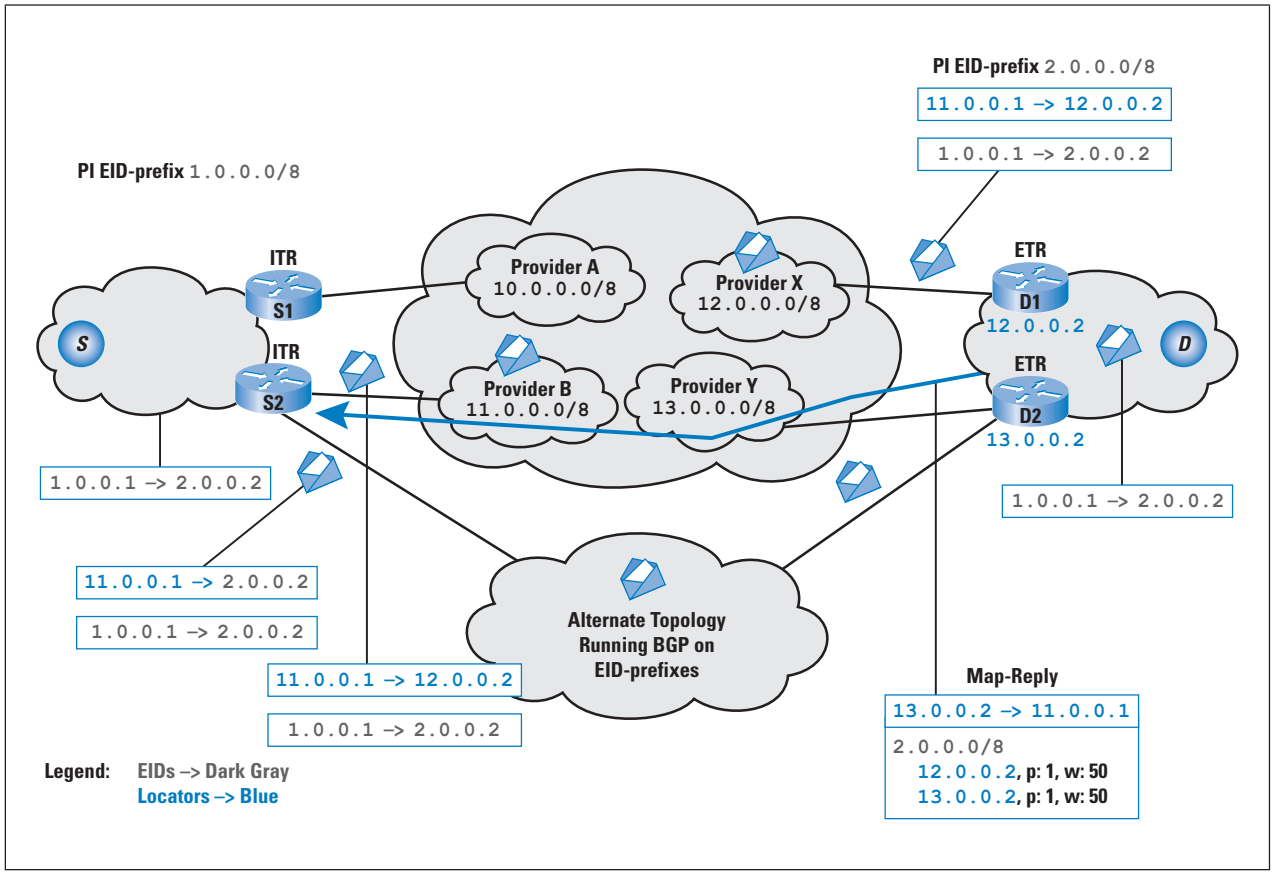
When a host in a LISP-capable domain wants to send a packet, it first looks up the correspondent host's EID in the DNS. It then puts its EID in the packet source address, and EID of the correspondent host in its destination address; if the destination of the packet is in another domain, the packet traverses the source domain infrastructure to one of the domain ITRs.

If the ITR has cached the EID-to-RLOC mapping for the destination EID, it sets the destination RLOC in the outer (encapsulated) header to the cached RLOC, and the source RLOC to its RLOC (note that the inner header has the source host's EID as the source and the destination's EID in the Destination field). The packet is then sent over the Internet to the ETR indicated in the destination RLOC, which decapsulates the packet and sends it on to the destination EID.

If, on the other hand, the ITR does not have a EID-to-RLOC mapping for the destination EID, it encapsulates the packet in a LISP header in which the destination address is the same as the inner header destination address, namely, the EID of the destination host. This packet is a Data Probe packet, and is routed over the LISP-ALT topology to the LISP-ALT router (typically an ETR, but this type of router is not required) that is authoritative for the EID-to-RLOC mapping. When the ETR receives the Data Probe packet, it decapsulates the packet and sends it on to the destination EID and sends a Map Reply to the source ITR so subsequent packets are sent natively over the Internet (as opposed to over the LISP-ALT overlay network). This query/response transaction is required only for the first packet sent between sites; all subsequent packets are sent LISP-encapsulated directly between the ITR and the ETR (and in particular, not over the LISP-ALT topology). Finally, note that the ITR could also preload its cache with mappings for popular destinations using the Map-Request message, avoiding the Data Probe packet (and associated latency, if any) altogether.

For example, consider the scenario depicted in Figure 3. In this case, a source *S* with EID `1.0.0.1` wants to send a packet to destination *D* whose EID is `2.0.0.2`. The packet arrives at ITR S2, which does not have an EID-to-RLOC mapping for `2.0.0.2`. S2 LISP-encapsulates the packet with the outer header having its RLOC (`11.0.0.1`) as the source address, copies the destination EID (`2.0.0.2`) from the inner header to the outer-header destination, and sends the data packet (a Data Probe) into the LISP-ALT topology. The packet follows the paths computed by BGP in the LISP-ALT topology to ETR D2. When D2 receives the packet, it decapsulates it and forwards the packet to the destination `2.0.0.2`; D2 also responds with a Map-Reply message that tells S2 (`11.0.0.1`) that the EID-to-RLOC mapping for `2.0.0.0/8` has two elements, ETR D1 (whose RLOC is `12.0.0.2`) and ETR D2 (whose RLOC is `13.0.0.2`). After receiving the Map Reply, ITR S2 can send LISP-encapsulated packets natively over the Internet (that is, not over the ALT topology).

Figure 3: A Day in the Life of a LISP Packet



Note that the mapping has priority (p) and weight (w) attributes. Priorities tell the ITR which ETRs to use in which order, and weights tell the ITR how to split load across ETRs of a given priority (w is a percentage of traffic that should go to each ETR). In this case, both ETRs have the same priority (1), and have weight 50 (that is, each ETR should receive 50 percent of the traffic).

New Functions Enabled by the Mapping System

Weights and priorities provide new capabilities for multihomed sites, which can use these features to control how traffic ingressing to the site is spread across its links without the complexity and overhead of running BGP. In particular, a multihomed site can configure its mapping database so that its links are used in an "active-active" configuration (that is, both links are in use). This situation is depicted in Figure 3, where the mapping databases entry `2.0.0.0/8` has two ETRs at the same priority that are equally weighted, meaning that the ITR will spread flows equally among the two ETRs.

This function is particularly attractive for *Small Office or Home Office* (SOHO) sites that desire both redundancy in their Internet connections and the ability to easily load share across those links in an active-active configuration, without the complexity and operational expense of running BGP.

Another interesting functionality enabled by the LISP control plane is the ability to mitagate some types of DoS attacks. In particular, if an ETR notices that it the subject of a DoS attack from behind an ITR (that is, DoS packets are destined to an EID-prefix for which it is authorative), it can use the LISP locator reachability bits (see Figure 2) to tell the the source ITR that the RLOC for that EID-prefix is not available. The ETR accomplishes this by sending a locator-reachability bit of zero for the RLOC to the offending ITR. Note that this functionality is similar to Ioannidis and Bellovin's "ICMP Pushback" proposal[25].

Performance Considerations

LISP and its associated mapping protocol(s) have two primary performance concerns:

- Encapsulation overhead
- EID-to-RLOC lookup latency and packet loss

In the case of encapsulation overhead, the concern is that the addition of the LISP header will cause the encapsulate packet to exceed the path *Maximum Transmission Unit* (MTU). As mentioned previously, this area of research is still active (see, for example, [18]).

In the case of lookup latency and packet loss, because LISP-ALT uses BGP to find a particular EID-to-RLOC mapping, there could be latency associated with the first few packets in the first flow between sites (note that it is only the first flow; subsequent flows can use the mapping installed in the ITR). However, this latency is mitigated, and the initial packets are not lost because LISP can send the first few data packets over the control plane; these packets are the Data Probe packets. There is additional latency associated with the time required for the destination ETR to return the Map Reply. However, after this initial transaction is completed, no additional latency is injected by the mapping system.

As mentioned previously, there is a trade-off in the mapping system among the state required to be held by network elements, the rate of updates to the mapping system, and the latency incurred when looking up an EID-to-RLOC mapping. LISP-ALT is a hybrid (push/pull) architecture that attempts to minimize the state requirements on ITRs, while at the same time minimizing lookup latency.

Conclusions

LISP is a new protocol that implements the Loc/ID split using a map-and-encap protocol. It obtains the advantages of the level of indirection afforded by the Loc/ID split while minimizing changes to hosts and to the core routing system. In addition, LISP enables new functions such as BGP-free multihoming in an active-active configuration.

Acknowledgments

The LISP specification and supporting documents are the work of many people, including Scott Brim, Noel Chiappa, Dino Farinacci, Vince Fuller, Eliot Lear, Darrel Lewis, and Dave Oran.

References

[0] Brim, S., et al., "EID Mappings Multicast Across Cooperating Systems for LISP," Internet Draft, Work in Progress, **draft-curran-lisp-emacs-00.txt**

[1] Brim, S., et al., "LISP-CONS: A Content distribution Overlay Network Service for LISP," Internet Draft, Work in Progress, **draft-meyer-lisp-cons-03.txt**

[2] Chiappa, N., "Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture," **http://ana.lcs.mit.edu/~jnc//tech/endpoints.txt**

[3] Farinacci, D., et al., "Locator/ID Separation Protocol (LISP)," Internet Draft, Work in Progress, **draft-farinacci-lisp-06.txt**

[4] Fuller, V., et al., "LISP Alternative Topology (LISP-ALT)," Internet Draft, Work in Progress, **draft-fuller-lisp-alt-01.txt**

[5] Jen, D., et al., "APT: A Practical Transit Mapping Service," Internet Draft, Work in Progress, **draft-jen-apt-01.txt**

[6] Lear, E., "NERD: A Not-so-Novel EID to RLOC Database," Internet Draft, Work in Progress, **draft-lear-lisp-nerd-03.txt**

[7] Massey, D., Wang, L., Zhang, B., and L. Zhang, "A Proposal for Scalable Internet Routing and Addressing," Internet Draft, Work in Progress, **draft-wang-ietf-efit-01.txt**

[8] Meyer, D., et al., "Report from the IAB Workshop on Routing and Addressing," RFC 4984, September 2007.

[9] Narten, T., et al., "Routing and Addressing Problem Statement," Internet Draft, Work in Progress, **draft-narten-radir-problem-statement-01.txt**

[10] Nordmark, E., "Shim6: Level 3 Multihoming Shim Protocol for IPv6," Internet Draft, Work in Progress, **draft-ietf-shim6-proto-09.txt**

[11] O'Dell, M., "GSE - An Alternate Addressing Architecture for IPv6," **http://www.watersprings.org/pub/id/draft-ietf-ipngwg-gseaddr-00.txt**

[12] Templin, F., "The IPvLX Architecture," Internet Draft, Work in Progress, **draft-templin-ipvlx-08.txt**

[13] Vogt, C., "Six/One: A Solution for Routing and Addressing in IPv6," Internet Draft, Work in Progress, **draft-vogt-rrg-six-one-01.txt**

[14] Whittle, R., "Ivip (Internet Vastly Improved Plumbing) Architecture," Internet Draft, Work in Progress, **draft-whittle-ivip-arch-01.txt**

[15] Bates, T., et al., "Multiprotocol Extensions for BGP-4," RFC 2858, June 2000.

[16] Farinacci, D., et al., "Generic Routing Encapsulation (GRE)," RFC 2784, March 2000.

[17] Rekhter, Y., (Ed.), et al., "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, January 2006.

[18] Templin, F., "Subnetwork Encapsulation and Adaptation Layer," Internet Draft, Work in Progress, **draft-templin-seal-02.txt**

[19] Bagnulo, M., "Preliminary LISP Threat Analysis," Internet Draft, Work in Progress, **draft-bagnulo-lisp-threat-01.txt**

[20] Perkins, C., "IP Mobility Support for IPv4, revised," Internet Draft, Work in Progress, **draft-ietf-mip4-rfc3344bis-05.txt**

[21] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6," RFC 3775, June 2004.

[22] Arkko, J., Vogt, C., and W. Haddad, "Enhanced Route Optimization for Mobile IPv6," RFC 4866, May 2007.

[23] Fenner, B., et al., "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)," RFC 4601, August 2006.

[24] Hinden, R., "New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG," RFC 1955, June 1996.

[25] Ioannidis John, and Bellovin, S., "Pushback: Router-Based Defense Against DDoS Attacks," `http://citeseer.ist.psu.edu/420554.html`

[26] Huston, G., "More ROAP: Routing and Addressing at IETF68," *The Internet Protocol Journal,* Volume 10, No. 2, June 2007.

[27] Carlos J. Bernardos, Ignacio Soto, and María Calderón, "IPv6 Network Mobility," *The Internet Protocol Journal,* Volume 10, No. 2, June 2007.

DAVID MEYER is currently a Director in the Advanced Research and Technologies Group at Cisco Systems, where he works on future directions for Internet technologies. E-mail: `dmm@cisco.com`

Book Review

Patterns in Network Architecture

Patterns in Network Architecture: A Return to Fundamentals, by John Day, ISBN-10: 0132252422, ISBN-13: 9780132252423, Prentice Hall, 2007. **http://www.informit.com/store/product.aspx?isbn=0132252422**

It isn't every day (pun intended) that one of the true Old Guard writes and publishes a book, and it behooves us to take notice. In this case, the author's expertise and his subject matter are of particular timeliness, because of the worldwide resurgence of activities with regard to next-generation network architectures, that is, a replacement, or upgrade to the Internet (dare one say "Internet 2.0"?).

John Day is a well-known scholar of historical cartography, and this book, in a way, is a roadmap of network architecture. The roadmap starts back in 1970, tracing from the roots of connectionless packet-switched dynamically routed systems such as Cyclades, and the ARPANET, through to recent discussions on multihoming, multicast, and mobility, with a view along the way of naming, addressing, protocol stack design, protocol design, and concepts of layering.

That description makes the book sound fairly standard in terms of structure and content, but it isn't. The book includes many discursive elements whose intent is to provide a collection of *patterns*. Design patterns originated in the building trade as a way for crafts people to pass on successful methods of construction (in the sense of affordable and noncollapsing) to less-inventive people (or people who want to spend their inventive efforts in different areas). Software engineers picked up on this idea, applying the techniques in both the microscopic world: patterns allow you to decide what algorithm is applicable in solving a problem in the small; and the macroscopic world: architectural patterns allow you to decide on an approach to breaking down a large system into the right kind of components.

Essentially, this book does the same thing, at the protocol stack level, and at the system level, with a collection of historical and contemporary examples to support the arguments.

The book makes interesting reading, especially as it represents a fair balance in reporting the early ideas that came not just from the United States, and restates the importance of the *Opens Systems Interconnection* (OSI) model (not the ISO protocols) in understanding layering and beads-on-a-string, as well as reasserting the use of the model in clarifying the perennially confusing concepts of names, addresses, and routes.

The book begins with a discussion of seven principles that emerged through the early history of networking (I won't spoil the book for readers by listing them here), and ends in the tenth and final chapter, entitled "Backing Out of a Blind Alley," with an appeal to fundamentals. Essentially, the author points out that researchers (especially academics) are strongly motivated to keep moving on with claims of ever-newer tricks, but rarely to consolidate these tricks into a set of principles that stand for a long time (because then they would have to completely change the topic of their research). Thus uncovering a foundational theory of networking would put a whole generation of networkers out of work (or funding at least).

The book is peppered (saltily) with fine quotes and fascinating asides from philosophy (for this reader, especially, the Chinese diversions were most novel and illuminating). Illustrative of the range is that one finds Wittgenstein and Dave Clark, Confucius, and Dr. Seuss—Frege's useful reminder that "The sign '=' should be read as 'is easily confused with'" would make an excellent IETF T-shirt.

I found the book extremely readable and enjoyable, and although I might argue with some of the opinions in the book, I think that this is just more evidence that I should recommend the book to anyone interested in knowing why we are where we are in networking, and being better informed about where we should go next.

—Jon Crowcroft, University of Cambridge
Jon.Crowcroft@cl.cam.ac.uk

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

ICANN Recovers Large Block of Internet Address Space

The *Internet Corporation for Assigned Names and Numbers* (ICANN) has found a little breathing room in the IPv4 address space with its recovery of a block of 16 million IPv4 addresses.

The IP addresses recovered were once used to connect older protocol packet-data networks with the fledgling Internet. The block of addresses, technically referred to as `14.0.0.0/8`, is also known as "Net-14."

"Net-14 was the easiest network to reclaim, the so-called low hanging fruit," said Barbara Roseman, General Manager with the *Internet Assigned Numbers Authority* (IANA), which is operated by ICANN. "None of the other legacy assignments in the IPv4 space are likely to be completely reclaimed as they are all in active use."

A small percentage of the addresses in Net-14 had been assigned, most more than 15 years ago. The assignments were so old that finding people who knew about them was a lengthy process. Nearly 50 organizations worked cooperatively with ICANN staff throughout 2007 to confirm that the 984 registrations were no longer in use. IANA undertook the reclamation effort to ensure that the greatest number of IPv4 addresses can be made available to Internet users as the overall free pool of IPv4 addresses is depleted. IANA allocates IPv4 and IPv6 addresses to *Regional Internet Registries* (RIRs). The five RIRs allocate addresses to network operators in their local regions. IANA allocated more than one /8 (16m IPv4 addresses) per month in 2007 and the rate of allocation is not expected to slow in 2008. The reclamation of Net-14 means there are now 43 unallocated /8s left.

"The recovery of these addresses offers some breathing room as the four billion addresses in IPv4 space are depleted, but it is only a temporary solution," added Roseman. "The real and lasting solution is the technical move to IPv6—the protocol that will make 340 trillion trillion trillion unique IP addresses available."

IPv6 Address Added for Root Servers in the Root Zone

ICANN recently took another step along the path of deployment for the next-generation IPv6 Internet addressing system. IPv6 addresses were added for six of the world's 13 *root server* networks (A, F, H, J, K, M) to the appropriate files and databases. This move allows for the possibility of fuller IPv6 usage of the *Domain Name System* (DNS). Prior to today, those using IPv6 had needed to retain the older IPv4 addressing system in order to be able to use domain names.

"The ISP community welcomes this development as part of the continuing evolution of the public Internet," said Tony Holmes, chair of ICANN's Internet Service and Connectivity Provider Constituency. "IPv6 will be an essential part our future and support in the root servers is essential to the growth, stability, and reliability of the public Internet."

Name server software relies on the root servers as a key part in translating domains like icann.org into the routing identifiers used by computers to connect to one another. In 2007 the ICANN *Security and Stability Advisory Committee* concluded that ICANN should move forward with the enhancement of the DNS root service by adding IPv6 addresses for the root servers."The addition of IPv6 addresses for the root servers enhances the end-to-end connectivity for IPv6 networks, and furthers the growth of the global interoperable Internet," added David Conrad, ICANN's Vice President of Research and IANA Strategy. "This is a major step forward for IPv6-only connectivity and the global migration to IPv6."

Further technical information on the move is available at:

`http://www.iana.org/reports/root-aaaa-announcement.html`

RIPE NCC Publishes Case Study of YouTube Hijack

As you may be aware from recent news reports, traffic to the **`youtube.com`** Website was "hijacked" on a global scale on Sunday February 24, 2008. The incident was a result of the unauthorized announcement of the prefix **`208.65.153.0/24`** and caused the popular video sharing Website to become unreachable from most, if not all, of the Internet. The RIPE NCC conducted an analysis into how this incident was seen and tracked by the RIPE NCC's *Routing Information Service* (RIS) and has published a case study at:

`http://www.ripe.net/news/study-youtube-hijacking.html`

The RIPE NCC RIS is a service that collects *Border Gateway Protocol* (BGP) routing information from roughly 600 peers at 16 *Internet Exchange Points* (IXPs) across the world. Data is stored in near real-time and can be instantly queried by anyone to provide multiple views of routing activity for any point in time. The RIS forms part of the RIPE NCC's suite of Information Services, which together provide a deeper insight into the workings of the Internet. The RIPE NCC is a neutral and impartial organization, and commercial interests therefore do not influence the data collected. The RIPE NCC Information Services suite also includes the *Test Traffic Measurement* (TTM) service, the *DNS Monitoring* (DNSMON) service and Hostcount. All of these services are available to anyone, and most of them are offered free of charge.

More information about RIPE NCC Information Services can be found at: **`http://is-portal.ripe.net`**

IETF Examines Future of the Internet by Going IPv6 Native

The *Internet Engineering Task Force* (IETF) put a spotlight on the next generation of Internet addressing when it switched off attendees' access to IPv4 during its March 2008 meeting. For an hour, Internet engineers at the meeting could only access the Internet using an IPv6 network.

During this event, IETF participants were encouraged to explore the Internet as it appears today in the IPv6 environment. The purpose of this exploration was to determine the next steps necessary toward deployment of IPv6 as the next generation of Internet addressing. The IETF undertook this activity at a time when IPv6-implementation is becoming a matter of global importance for the Internet. The event provided all IETF meeting attendees a first-hand opportunity to work with the Internet over an exclusive IPv6 network. "We get a lot of reports from members of our community who use IPv6, but this was an opportunity for everyone to observe and discuss the technical issues as a group," said Russ Housley, Chair of the IETF. "This first-hand data helps to inform our engineering decisions."

Some members of the Internet technical community assert that the ongoing deployment of IPv6 has been held back by a lack of IPv6-accessible Websites, creating the classic first-step dilemma for network operators. "It has been incredible to observe as members of the community organized themselves and updated their home networks to be ready for this event," said Leslie Daigle, Chief Internet Technology Officer at the Internet Society. "As we continue to solve the engineering and implementation obstacles to IPv6 deployment, creative engineers around the world will develop new uses for the Internet, through IPv6, in ways we can't yet imagine."

The IETF has provided dual stack IPv4/IPv6 network connectivity at its meetings for years, which has been useful for its regular IPv6-using attendees. The difference during this meeting was that a strictly IPv6 network was made available as well, and all attendees were encouraged to explore and experiment with the Internet as seen from IPv6. This focus was heightened when IPv4 access was deliberately shut off for an hour, leaving only IPv6 for connectivity. Following this—and other similar experiments—the engineering community expects to have a better understanding of the next steps necessary in the development of protocols and standards to support the continued deployment of IPv6 in support of the global Internet. The Comcast Corporation provided the facilities to conduct the live test of IPv6 and was the host sponsor of IETF-71 in Philadelphia.

For more information about this event, and similar events please see:

http://www.isoc.org/educpillar/resources/ipv6_faq.shtml

http://wiki.tools.isoc.org/IETF71_IPv4_Outage

http://www.civil-tongue.net/clusterf/

Postel Network Operator's Scholarship 2008

The *North American Network Operators' Group* (NANOG) and the *American Registry for Internet Numbers* (ARIN) have been unique and successful cooperative fora for Internet builders in North America and other parts of the world. Senior practitioners from around the world contribute their time to NANOG and ARIN as presenters, teachers and trainers, to produce consistent non-commercial conferences of high-quality.

Since 2007, the generosity of an anonymous donor and the administration of the Internet Society, have allowed NANOG and ARIN to provide financial support to a person from a developing country to participate in the October joint NANOG/ARIN meeting through the *Postel Network Operator's Scholarship*.

The Scholarship Committee cordially invites suitable applicants to apply for fellowship funding to participate in the October 2008 joint NANOG/ARIN meeting. The Scholarship targets personnel from developing countries who are actively involved in Internet development, in any of the following roles: Engineers (Network Builders), Operational and Infrastructure Support Personnel, and Educators, Teachers, and Trainers

Successful applicants will be provided with transportation to and from the meetings and a reasonable allowance for food and accommodation. In addition all fees for participation in the conferences, tutorials, and social events will be waived. Applicants from any part of the world will be considered. The deadline for application is June 1, 2008, and the awardee will be informed by July 1, 2008.

To apply for the fellowship please read **`http://www.nanog.org/postel-scholarship.html`** and submit your application by e-mail to **`PostelNOS@nanog.org`**

For more information about NANOG and ARIN meetings, see: **`http://www.nanog.org/`** and **`http://www.arin.net/`**

JPNIC Releases IPv4 Exhaustion Report

The *Japan Network Information Center* (JPNIC) has released a report entitled "Study Report on the IPv4 Address Space Exhaustion Issue (Phase I)." The report can be downloaded from the following link:

`http://www.nic.ad.jp/en/ip/ipv4pool/ipv4exh-report-071207-en.pdf`

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, fire-walls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

June 2008 Volume 11, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

Ten years ago we published the first issue of *The Internet Protocol Journal* (IPJ). Since then, 41 issues and a total of 1,612 pages have been produced. Today, IPJ has about 37,000 subscribers all around the world. Although most of our readers prefer the paper edition, a growing number of subscribers are reading IPJ online or downloading the PDF version. This shift in reading habits may be related to the changes in technology over the last 10 years. Lower costs and higher-resolution displays and printers, as well as improvements in Internet access technologies, have made the online "experience" a lot better than in 1998.

Publishing is by no means the only area that has seen dramatic changes in the last decade. We asked Vint Cerf and Geoff Huston to reflect on Internet developments in this period, and the resulting articles, "A Decade of Internet Evolution" and "A Decade in the Life of the Internet," are included in this issue.

Let me take this opportunity to thank all those people who have made IPJ possible. Our authors deserve a round of applause for carefully explaining both established and emerging technologies. They are assisted by an equally insightful set of reviewers and advisors who provide feedback and suggestions on every aspect of our publications process. The process itself relies heavily on two individuals: Bonnie Hupton, our copy editor, and Diane Andrada, our designer. Thanks go also to our printers and mailing and shipping providers. Last, but not least, our readers provide encouragement, suggestions, and feedback. This journal would not be what it is without them.

Because we are considering some Internet history in this issue, I would like to announce a project that takes us even further back. Before joining Cisco in 1998 I worked at the Interop Company, where I was responsible for the monthly publication of *ConneXions—The Interoperability Report,* published from 1987 through 1996. Unlike IPJ, *ConneXions* was produced in the "old-fashioned way" using various pieces of text and artwork assembled onto paste-up boards, and then photographed for subsequent plate making and offset printing. Thus no PDF files were produced at the time, but I am pleased to announce that *The Charles Babbage Institute* at the University of Minnesota has scanned the complete collection (117 issues) and it is now available at: **http://www.cbi.umn.edu/hostedpublications/Connexions/index.html**

Our final article is a look at Mobile WiMAX. WiMAX is an emerging technology that was originally designed as a fixed wireless broadband technology, a "DSL replacement," but has evolved to support mobility.

— Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

A Decade of Internet Evolution

by Vinton G. Cerf, Google

In 1998 the Internet had about 50 million users, supported by approximately 25 million servers (Web and e-mail hosting sites, for example, but not desktops or laptops). In that same year, the *Internet Corporation for Assigned Names and Numbers* (ICANN)[1] was created. Internet companies such as Netscape Communications, Yahoo!, eBay, and Amazon were already 3 to 4 years old and the Internet was in the middle of its so-called "dot-boom" period. Google emerged that year as a highly speculative effort to "organize the world's information and make it accessible and useful." Investment in anything related to the Internet was called "irrational exuberance" by the then head of the U.S. Federal Reserve Bank, Alan Greenspan.

By April 2000, the Internet boom ended—at least in the United States—and a notable decline in investment in Internet application providers and infrastructure ensued. Domino effects resulted for router vendors, Internet service providers, and application providers. An underlying demand for Internet services remained, however, and it continued to grow, in part because of the growth in the number of Internet users worldwide.

During this same period, access to the Internet began to shift from dial-up speeds (on the order of kilobits to tens of kilobits per second) to broadband speeds (often measured in megabits per second). New access technologies such as digital subscriber loops and dedicated fiber raised consumer expectations of Internet capacity, in turn triggering much interest in streaming applications such as voice and video. In some locales, consumers could obtain gigabit access to the Internet (for example, in Japan and Stockholm). In addition, mobile access increased rapidly as mobile technology spread throughout the world, especially in regions where wireline telephony had been slow to develop.

Today the Internet has an estimated 542 million servers and about 1.3 billion users. Of the estimated 3 billion mobile phones in use, about 15 percent are Internet-enabled, adding 450 million devices to the Internet. In addition, at least 1 billion personal computers are in use, a significant fraction of which also have access to the Internet. The diversity of devices and access speeds on the Internet combine to produce challenges and opportunities for Internet application providers around the world. Highly variable speeds, display areas, and physical modes of interaction create a rich but complex canvas on which to develop new Internet applications and adapt older ones.

Another well-documented but unexpected development during this same decade is the dramatic increase in user-produced content on the Internet. There is no question that users contributed strongly to the utility of the Internet as the World Wide Web made its debut in the early 1990s with a rapidly growing menu of Web pages.

But higher speeds have encouraged user-produced audio and video archives (*Napster* and *YouTube*), as well as sharing of all forms of digital content through peer-to-peer protocols. Voice over IP, once a novelty, is very common, together with video conferencing (*iChat* from Apple, for example).

Geographically indexed information has also emerged as a major resource for Internet users. In the scientific realm, *Google Earth* and *Google Maps* are frequently used to display scientific data, sensor measurements, and so on. Local consumer information is another common theme. When I found myself in the small town of Page, Arizona, looking for saffron to make paella while in a houseboat on Lake Powell, a Google search on my Blackberry quickly identified markets in the area. I called one of them and verified that it had saffron in stock. I followed the map on the Website and bought 0.06 ounces of Spanish saffron for about $12.99. This experience reinforced my belief that having locally useful information at your fingertips no matter where you are is a powerful ally in daily living.

New business models based on the economics of digital information are also emerging. I can recall spending $1,000 for about 10 MB of disk storage in 1979. Recently I purchased 2 TB of disk storage for about $600. If I had tried to buy 2 TB of disk storage in 1979, it would have cost $200 million, and probably would have outstripped the production capacity of the supplier. The cost of processing, storing, and transporting digital information has changed the cost basis for businesses that once required the physical delivery of objects containing information (books, newspapers, magazines, CDs, and DVDs). The Internet can deliver this kind of information in digital form economically—and often more quickly than physical delivery. Older businesses whose business models are based on the costs of physical delivery of information must adapt to these new economics or they may find themselves losing business to online competitors. (It is interesting to note, however, that the Netflix business, which delivers DVDs by postal mail, has a respectable data rate of about 145 kbps per DVD, assuming a 3-day delivery time and about 4.7 GB per DVD. The CEO of Netflix, Reed Hastings, told me nearly 2 years ago that he was then shipping about 1.9 million DVDs per day, for an aggregate data rate of about 275 Gbps!)

Even the media that have traditionally been delivered electronically such as telephony, television, and radio are being changed by digital technology and the Internet. These media can now be delivered from countless sources to equally countless destinations over the Internet. It is common to think of these media as being delivered in streaming modes (that is, packets delivered in real time), but this need not be the case for material that has been prerecorded. Users of iPods have already discovered that they can download music faster than they can listen to it.

With gigabit access to the Internet, one could download an hour's worth of conventional video in about 16 seconds. This fact certainly changes my understanding of "video on demand" from a streaming delivery to a file transfer. The latter is much easier on the Internet because one is not concerned about packet inter-arrival times (jitter), loss, or even orderly delivery because the packets can be reordered and retransmitted during the file transfer. I am told that about 10 hours of video are being uploaded to YouTube per second.

The battles over *Quality of Service* (QoS) are probably not over yet either. Services such as *Skype* and applications such as iChat from Apple demonstrate the feasibility of credible, real-time audio and video conferencing on the "best-efforts" public Internet. I have been surprised by the quality that is possible when both parties have reasonably high-capacity access to the Internet.

Technorati is said to be tracking on the order of 112 million blogs, and the *China Internet Network Information Center* (CNNIC) estimates 72 million Chinese blogs that are probably in addition to those tracked by Technorati. Adding to these are billions of Web pages and, perhaps even more significant, an unknown amount of information online in the form of large databases. The latter are not indexed in the same way that Web pages can be, but probably contain more information. Think about high-energy physics information, images from the Hubble and other telescopes, radio telescope data including the *Search for Extra-Terrestrial Intelligence* (SETI)[2], and you quickly conclude that our modern society is awash in digital information.

It seems fair to ask how long accessibility of this information is likely to continue. By this question I do not mean that it may be lost from the Internet but, rather, that we may lose the ability to interpret it. I have already encountered such problems with image files whose formats are old and whose interpretation by newer software may not be possible. Similarly, I have ASCII text files from more than 20 years ago that I can still read, but I no longer have operating software that can interpret the formatting instructions to produce a nicely formatted page. I sometimes think of this problem as the "year 3000" problem: It is the year 3000 and I have just finished a Google search and found a PowerPoint 1997 file. Assuming I am running Windows 3000, it is a fair question whether the format of this file will still be interpretable. This problem would arise even if I were using open-source software. It seems unlikely that application software will last 1000 years in the normal course of events unless we deliberately take steps to preserve our ability to interpret digital content. Absent such actions, we will find ourselves awash in a sea of rotting bits whose meaning has long since been lost.

This problem is not trivial because questions will arise about intellectual property protection of the application, and even the operating system software involved. If a company goes out of business or asserts that it will no longer support a particular version of an application or operating system, do we need new regulations that require this software to be available on the public Internet in some way?

Even if we have skirted this problem in the past by rendering information into printed form, or microfilm, the complexity of digital objects is increasing. Consider spreadsheets or other complex objects that really cannot be fully "rendered" without the assistance of application software. So it will not be adequate simply to print or render information in other long-lived media formats. We really will need to preserve our ability to read and interpret bits.

The year 2008 also marks the tenth anniversary of a project that started at the U.S. Jet Propulsion Laboratory: *The Interplanetary Internet*. This effort began as a protocol design exercise to see what would have to change to make Internet-like capability available to manned and robotic spacecraft. The idea was to develop networking technology that would provide to the space exploration field the kind of rich and interoperable networking between spacecraft of any (Earth) origin that we enjoy between devices on the Internet.

The design team quickly recognized that the standard TCP/IP protocols would not overcome some of the long delays and disruptions to be expected in deep space communication. A new set of protocols evolved that could operate above the conventional Internet or on underlying transport protocols more suited to long delays and disruption. Called "delay and disruption tolerant networking"[3, 4] or DTN, this suite of protocols is layered in the same abstract way as the Internet. The Interplanetary system could be thought of as a network of Internets, although it is not constrained to use conventional Internet protocols. The analog of IP is called the *Bundle Protocol*[5], and this protocol can run above TCP or the *User Datagram Protocol* (UDP) or the new *Licklider Transport Protocol* (for deep space application). Ironically, the DTN protocol suite has also proven to be useful for terrestrial applications in which delay and disruption are common: tactical military communication and civilian mobile communication.

After 10 years of work, the DTN system will be tested onboard the Deep Impact mission platform late in 2008 as part of a program to qualify the new technology for use in future space missions. It is hoped that this protocol suite can be standardized for use by any of the world's space agencies so that spacecraft from any country will be interoperable with spacecraft of other countries and available to support new missions if they are still operational and have completed their primary missions. Such a situation already exists on Mars, where the Rovers are using previously launched orbital satellites to relay information to Earth's Deep Space Network using store-and-forward techniques like those common to the Internet.

The Internet has gone from dial-up to deep space in just the past 10 years. One can only begin to speculate about its application and condition 10 years hence. We will all have to keep our subscriptions to *The Internet Protocol Journal* to find out!

References

[1] Cerf, V., "Looking Toward the Future," *The Internet Protocol Journal*, Volume 10, No. 4, December 2007.

[2] `http://www.seti.org`

[3] `http://www.dtnrg.org/wiki`

[4] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," RFC 4838, April 2007.

[5] Scott, K., and S. Burleigh, "Bundle Protocol Specification," RFC 5050, November 2007.

VINTON G. CERF is vice president and chief Internet evangelist for Google. Cerf served as a senior vice president of MCI from 1994 through 2005. Widely known as one of the "Fathers of the Internet," Cerf is the co-designer of the TCP/IP protocols and the architecture of the Internet. He received the U.S. National Medal of Technology in 1997 and the 2004 ACM Alan M. Turing award. In November 2005, he was awarded the Presidential Medal of Freedom. Cerf served as chairman of the board of the Internet Corporation for Assigned Names and Numbers (ICANN) from 2000 through 2007 and was founding president of the Internet Society. He is a Fellow of the IEEE, ACM, the American Association for the Advancement of Science, the American Academy of Arts and Sciences, the International Engineering Consortium, the Computer History Museum, and the National Academy of Engineering. He is an honorary Freeman of the City of London. Cerf holds a Bachelor of Science degree in Mathematics from Stanford University and Master of Science and Ph.D. degrees in Computer Science from UCLA. E-mail: `vint@google.com`

A Decade in the Life of the Internet

by Geoff Huston, APNIC

The evolutionary path of any technology can often take strange and unanticipated turns and twists. At some points simplicity and minimalism can be replaced by complexity and ornamentation, while at other times a dramatic cut-through exposes the core concepts of the technology and removes layers of superfluous additions. The technical evolution of the Internet appears to be no exception, and contains these same forms of unanticipated turns and twists.

This article presents a personal perspective of the evolution of the Internet over the last decade, highlighting my impressions of what has worked, what has not, and what has changed over this period. It has been an extraordinary decade for the Internet, encompassing a boom and a bust that would rate among history's best, a comprehensive restructuring of the communications industry, and a set of changes that have altered the way in which each of us now works and plays. And the Internet has even added a few new words to the language on the way.

Rather than offer a set of random observations, I will use the Internet Protocol model as a template, starting with the underlying transmission media, then looking at the internetwork layer, the transport layer, then applications and services, and, finally looking at the business of the Internet.

The Transmission Media Layer

It seems like it was in an entirely different lifetime, but the *Internet Service Provider* (ISP) business of 1998 was still centrally involved in the technology of dial-up modems. The state-of-the-art of modem speed had been continually refined from 9,600 bps to 14.4 kbps, to 28 kbps, to finally, 56 kbps, squeezing every last bit out the phase amplitude space contained in an analogue 3-KHz voice circuit. Modems were the bane of an ISP's life. They were capricious, constantly being superseded by the next technical refinement, unreliable, difficult for customers to use, and they were just slow. Almost everything else on the Internet was tailored to download reasonably quickly over a modem connection. Webpages were carefully tailored with compressed images, and plaintext was the dominant medium as a consequence.

Not all forms of Internet access were dial-up. ISDN was used in some places, but it was never cheap enough to take over as the ubiquitous access method. There were also access services based on *Frame Relay,* X.25, and various forms of digital data services. At the high end of the speed spectrum were T1 access circuits with 1.5-Mbps clocking, and T3 circuits clocked at 45 Mbps.

ISPs leased circuits from a telephony company (telco). In 1998 the ISP industry was undergoing a transition of its trunk IP infrastructure from T1 circuits to T3 circuits. It was not going to stop here, but squeezing even more capacity from the network was proving to be a challenge. Deployment of 622-Mbps IP circuits occurred, although many of these were constructed using 155-Mbps *Asynchronous Transfer Mode* (ATM) circuits using router load balancing to share the IP load over four of these circuits in parallel. Gigabit circuits were just beginning, and the initial tests of IP over 2.5-Gbps *Synchronous Digital Hierarchy* (SDH) circuits began in 1998.

In some ways 1998 was a pivotal year for IP transmission. Until this time IP was still just another application that was positioned as just another customer of the telco's switched-circuit infrastructure that was constructed primarily to support telephony. From the analogue voice circuits to the 64K digital circuit through to the trunk bearers, IP had been running on top of the voice network. By 1998 things were changing. The Internet had started to make ever larger demands on transmission capacity, and the factor accelerating further growth in the network was now not voice, but data. It made little sense to provision an ever larger voice-based switching infrastructure just to repackage it as IP, and by 1998 the industry was starting to consider just what an all-IP high-speed network would look like, from the photon all the way through to the application.

At the same time the fiber-optic systems were changing with the introduction of *Wavelength-Division Multiplexing* (WDM). Older fiber equipment with electro-optical repeaters and *Plesiochronous Digital Hierarchy* (PDH) multiplexers allowed a single fiber pair to carry around 560 Mbps of data. WDM allowed a fiber pair to carry multiple channels of data using different wavelengths, with each channel supporting a data rate of up to 10 Gbps. Channel capacity in a fiber strand is between 40 to 160 channels using *Dense WDM* (DWDM). Combined with the use of all-optical amplifiers, the most remarkable part of this entire evolution in fiber systems is that a Tbps cable system can be constructed today for much the same cost as a 560-Mbps cable system of the mid-1990s. The factor that accelerated deployment of these high-capacity fiber systems was never based on expansion of telephony, because the explosive growth of the industry was all about IP. So it came as no surprise that at the same time as the demand for IP transmission was increasing there was a shift in the transmission model, where instead of plugging routers into telco switching gear and using virtual point-to-point circuits for IP, we started to plug routers into wavelengths of the DWDM equipment and operate all-IP networks in the core of the Internet.

The evolution of access networks has seen a shift away from modems to numerous digital access methods, including DSL, cable modems, and high-speed wireless services. The copper pair of the telco network has proved surprisingly resilient, and DSL has achieved speeds of tens of megabits per second through this network, with the prospect of hundred-megabit systems appearing soon.

So, in terms of transmission, the last 10 years has seen the network migrate from an overlay system of kilobit-per-second access with multimegabit trunks operating as a customer of the telco switched network to a comprehensive IP network with access of megabits per second with multigigabit trunks, or a thousandfold increase in basic network capacity in that period.

The demand of the Internet for capacity continues, and we are now seeing work on standardizing 40- and 100-Gbps transmission systems in the IEEE; the prospect of terabit transmissions is now taking shape for the Internet.

The Internet Layer

If transmission has seen dramatic changes in the past decade, then what has happened at the IP layer over the same period?

The glib answer is "absolutely nothing!" But that answer would be ignoring a large amount of activity in this area. We have tried to change many parts of IP in the past decade, but, interestingly, none of the proposed changes has managed to gain any significant traction in the network, and IP today is largely no different from IP of a decade ago. *Mobility*[1], *Multicast*[2], and *IP Security* (IPSec)[3] remain poised in the wings, still awaiting adoption by the Internet mainstream.

Quality of Service (QoS) was a "hot" topic in 1998, and it involved the search for a reasonable way for some packets to take the fast path while others took a more leisurely way through the network. We experimented with various forms of signaling, packet classifiers, queue-management algorithms, and interpretations of the *Type of Service* bits in the IPv4 packet header, and we explored the QoS architectures of *Integrated and Differentiated Services* in great detail. However, QoS never managed to achieve wide acceptance in mainstream Internet service environments. In this case the Internet took a simpler direction: In response to not enough network capacity, the alternate approach to installing additional mechanisms in the network—in the host protocol stack and even in the application in order to ration the capacity you have—is to simply expand the network to meet the total level of demand. So far the simple approach has prevailed in the network, and QoS remains largely unused[4].

We have experimented with putting circuits back into the IP architecture in various ways, most notably with the *Multiprotocol Label Switching* (MPLS) technology[5]. This technology used the label-swapping approach used in X.25, Frame Relay, and ATM virtual circuit switching systems; it created a collection of virtual paths from each network ingress to each network egress. The idea was that in the interior of the network you no longer needed to load up a complete routing table into each switching element, and instead of performing destination-address lookup you could perform a much smaller, and hopefully faster, label lookup.

This process did not eventuate, and switching packets using the 32-bit destination address continued to present much the same level of cost-efficiency at the hardware level as virtual circuit label switching. When you add the additional overhead of an additional level of indirection in terms of operational management of MPLS networks, MPLS became another technology that so far has not managed to achieve traction in mainstream Internet networks. However, MPLS is by no means a dormant technology, and one place where MPLS has enjoyed considerable deployment is in the corporate service sector where many *Virtual Private Networks*[6] are constructed using MPLS as the core technology, steadily replacing a raft of traditional private data systems that used X.25, Frame Relay, ATM, *Switched Multimegabit Data Service* (SMDS), and switched Ethernet.

Of course one change at the IP level of the protocol stack that was intended in the past decade but has not occurred is *IP Version 6*[7]. In 1998 we were forecasting that we would have consumed all the remaining unallocated IPv4 addresses by around 2008. We were saying at the time that, because we had completed the technical specification of IPv6, the next step was that of deployment and transition. There was no particular sense of urgency, and the comfortable expectation was that with a decade to go we did not need to raise any alarms. And this plan has worked, to some extent, in that today's popular desktop operating systems of Windows, MacOS, and UNIX all have IPv6 support. But other parts of this transition have been painfully slow. It was only a few months ago that the root of the *Domain Name System* (DNS) was able to answer queries using the IPv6 protocol as transport, and provide the IPv6 addresses of the root nameservers. Very few mainstream services are configured in a dual-stack fashion, and the prevailing view is still that the case for IPv6 deployment has not yet reached the necessary threshold. Usage measurements for IPv6 point to a level of deployment of around one-thousandth of the IPv4 network, and, perhaps more worrisome, this metric has not changed to any appreciable level in the past 4 years. So what about that projection of IPv4 unallocated pool exhaustion by 2008? How urgent is IPv6 now? The good news is that the *Internet Assigned Numbers Authority* (IANA) still has some 16 percent of the address space in its unallocated pool, so IPv4 address exhaustion is unlikely to occur this year. The bad news is that the global consumption rate of IP addresses is now at a level such that the remaining address pool can fuel the Internet for less than a further 3 years, and the exhaustion prediction is now sometime around 2010 to 2011.

So why have we not deployed IPv6 more seriously yet? And if we are not going to deploy IPv6, then what is the alternative? Of all the technical refinements to IP that have occurred, one that received little fanfare when it was first published has enjoyed massive deployment over the past decade, and that is the technology of *Network Address Translation* (NAT)[8]. Today NAT devices are ubiquitous. It seems that every home access unit, every corporate firewall, every data center, and every service includes a NAT device.

One measure of the ubiquity of NATs is the transformation that has occurred in the application space. By 2008 applications have either adopted a strict client-server approach, where the client always initiates the network transaction, or were forced down a more complex path. Where there is some form of peer interaction, applications are now equipped with additional capabilities, including NAT behavior discovery, NAT binding management, application-level name spaces, and multiparty rendezvous mechanisms, all required to allow the application to function across NATs. So far we have managed to offload the problem of looming address scarcity in the Internet onto NATs, and the really significant change that has occurred in the past decade at the IP level is the default assumption about the semantics of an IP address. An IP address is no longer synonymous with the persistent identity of the remote party that anyone can use to initiate a communication, but a temporary token to allow a single transaction to complete. As a consequence, most Internet services have retreated into data centers and the business of hosting services has thrived. And the change that would have preserved the coherent end-to-end architecture of the Internet IP layer, namely IPv6, is still waiting for wide-scale deployment.

The next few years promise to be "interesting" in every form of meaning of the word. The exhaustion of the remaining IPv4 address pool is imminent, and if we are going to substitute IPv6 in place of IPv4, then we simply do not have enough time to achieve this substitution before the remaining IPv4 address pool is depleted. And although so far NATs have conveniently pushed the problem of increasing address scarcity off the network and over to the edge devices and onto applications, it is not clear that this approach can sustain an ever-growing Internet indefinitely. We have yet to understand just what a "carrier-grade NAT" might be, or whether it can even work in any useful manner. NATs were an accidental addition to the Internet, and their role in the coming years is unclear.

The early 1990s saw a flurry of activity in the routing space, and protocols were quickly developed and deployed. By 1998 the "standard" Internet environment involved the use of either *Intermediate System-to-Intermediate System* (IS-IS) or *Open Shortest Path First* (OSPF) as large-scale interior routing protocols and *Border Gateway Protocol 4* (BGP4) as the interdomain routing protocol[9]. This picture has remained constant over the past decade. In some ways it is reassuring to see a technology that is capable of sustaining a quite dramatic growth rate, but perhaps that is not quite the complete picture.

We never quite completed the specification for the next interdomain routing protocol, and BGP4 is now showing signs of stress[10]. The pool of *Autonomous System* (AS) numbers is forecast to run out early in 2011, and by then we need to have fielded a new variant of BGP that can operate with a much larger pool of AS numbers[11].

Fortunately the technology development has been completed and an approach that allows incremental deployment has been devised, so this transition is not quite the traumatic transition that is associated with IPv6. But deployment is slow, and of the current level of adoption of the larger AS number set is, oddly enough, comparable to IPv6, at a level of around one-thousandth of the total AS number pool. The routing system has also been growing inexorably, and the capability of switching systems to cope with ever larger routing tables while at the same time offering continual improvements in cost-efficiencies is now looking less certain. So, once again we appear to be examining routing protocol theory and practice, and looking at alternate approaches to routing that can offer superior scaling properties to BGP for the future.

No listing of the major highlights in IP over the past decade would be complete without some mention of the perennial issue of *location* and *identity*.[25] One of the original simplifications in the IP architecture was to place the semantics of identity, location, and forwarding into an IP address. Although that process has proved phenomenally effective in terms of simplicity of applications and simplicity of IP networks, it has posed some serious challenges with regard to mobility, routing, and network management. Each of these aspects of the Internet would benefit considerably if the Internet architecture allowed identity to be distinct from location. Numerous efforts have been directed at this problem over the past decade, particularly in IPv6, but so far we really have not arrived at an approach that feels truly comfortable in the context of IP.

So although it is possible to observe that not much has happened at the IP level in the past decade that is deployed in the Internet—and IP is still IP—there is still a considerable agenda to tackle at the Internet layer.

The Transport Layer

A decade ago, in 1998, the transport layer of the IP architecture consisted of the *User Datagram Protocol* (UDP) and TCP, and the network usage pattern was around 95-percent TCP and 5-percent UDP. Here, as well, not much has changed in the intervening 10 years.

We have developed two new transport protocols, the *Datagram Congestion Control Protocol* (DCCP) and the *Stream Control Transmission Protocol* (SCTP)[12], which can be regarded as refinements of TCP to cover flow control for datagram streams in the case of DCCP and flow control over multiple reliable streams in the case of SCTP. However, in a world of transport-aware middleware that is the Internet today, the level of capability to actually deploy these new protocols in the public Internet is marginal at best.

TCP has proved to be remarkably resilient over the years, but as the capacity of the network increases the ability of TCP to continue to deliver ever faster data rates over distances that span the globe is becoming a significant concern. Recent times have seen much work to devise revised TCP flow-control algorithms that still share the network fairly with other concurrent TCP sessions, yet can ramp up to multigigabit-per-second data-transfer rates and sustain those rates over extended periods[13]. At this stage much of this work is still in the area of research and experimentation, and TCP today as deployed on the Internet is much the same as TCP of a decade ago, with perhaps a couple of notable exceptions. The latest TCP stack from Microsoft in Vista uses dynamic tuning of the Receive window, and a larger inflation factor of the Send window in congestion avoidance where there is a large bandwidth delay product, and improved loss-recovery algorithms that are particularly useful in wireless environments. Linux now includes an implementation of *Binary Increase Congestion control* (BIC), which undertakes a binomial search to reestablish a sustainable send rate. Both of these approaches can improve the performance of TCP, particularly when sending the TCP session over long distances and trying to maintain high transfer speeds.

The Application and Service Layer

This area, unlike the transport layer, has seen quite profound changes over the past decade. A decade ago the Internet was on the cusp of portal mania, where *LookSmart* was the darling of the Internet boom and everyone were all trying to promote their own favorite "one stop shop" for all their Internet needs. We were still using various forms of hand-compiled directories, and navigation of the Internet was still the subject of various courses and books.

By 1998 *AltaVista* has made its debut, and change was already evident. This change, from directories and lists to active search, completely changed the Internet. These days we simply assume that we can type any query we have into a search engine and the search machinery will deliver a set of pointers to relevant documents. Each time this process occurs our expectations about the quality and utility of search engines are reinforced, and we have moved beyond swapping URLs as pointers and simply exchange search terms as an implicit reference to the material. Content is also changing as a result, because users no longer remain on a "site" and navigate around the site. Instead users are directing the search engines, and pulling the relevant page form the target site without reference to any other material.

Another area of profound change has been the rise of active collaboration over content, best typified in wikis. *Wikipedia* is perhaps the most cited example of user-created content, but almost every other aspect of content generation is also being introduced into the active user model, including *YouTube, Flickr, Joost,* and similar content.

Underlying these changes is another significant development, namely the changes in the content economy. In 1998 content providers and ISPs were competing for user revenue. Content providers were unable to make pay per view and other forms of direct financial relationship with users work in their favor, and were arguing that ISPs should fund content, because, after all, the only reason that users paid for Internet access was because of their perceived value of the content. ISPs, on the other hand, promoted the idea that content providers were enjoying a free ride across the ISP-funded infrastructure, and content providers should contribute to network costs. The model that has gained ascendency as a result of this unresolved tension was that of advertised-funded content services, and this model has sustained a vastly richer, larger, and more compelling content environment.

At the same time the peer-to-peer network has emerged, and from its beginnings as a music-sharing subsystem, the distributed data model of content sharing now dominates the Internet with audio, video, and large data sets now using this form of content distribution and its associated highly effective transport architecture. Various measurements of Internet traffic have placed peer-to-peer content movement at between 40 and 80 percent of the overall traffic profile of the network.

In many ways applications and services have been the high frontier of innovation in the Internet in the past decade. An entire revolution in open interconnection of content elements is embraced under the generic term *Web 2.0*, and "content" is now a very malleable concept. It is no longer the case of "my computer, my applications, and my workspace" but an emerging model where not only the workspace for each user is held in the network, but where the applications themselves are part of the network, and all are accessed through a generic browser interface.

Any summary of the evolution of the application space over the last decade would not be complete without noting that whereas in 1998 the Internet was still an application that sat on top of the network infrastructure used to support the telephone network, by 2008 voice telephony was just another application layered on the infrastructure of the Internet, and the Internet had even managed to swallow the entire telephone number space into its DNS, using an approach called *ENUM*[14].

The Business Layer

As much as the application environment of the Internet has been wildly erratic over the past decade, the business environment has been unpredictable as well, and the list of business winners and losers includes some of the historical giants of the telephone world as well as the Internet-bred new wave of entrants.

In 1998, despite the growing momentum of public awareness, the Internet was still largely a curiosity. It was an environment inhabited by geeks, game players, and academics, whose rites of initiation were quite arcane. As a part of the data networking sector, the Internet was just one further activity among many, and the level of attention from the mainstream telco sector was still relatively small. Most Internet users were customers of independent ISPs, and the business relationship between the ISP sector and the telco was tense and acrimonious. The ISPs were seen as opportunistic leeches on the telco industry; they ordered large banks of phone lines, but never made any calls; their customers did not hang up after 3 minutes, but kept their calls open for hours or even days at a time, and they kept ordering ever larger inventories of transmission capacity, yet had business plans that made the back of an envelope look professional by comparison. The telco was unwilling to make large long-term capital investments in additional infrastructure to pander to the extravagant demands of a wildcat set of Internet speculators and their fellow travelers. The telco, on the other hand was slow, expensive, inconsistent, ill-informed, and hostile to the ISP business. The telco wanted financial settlements and bit-level accounting, whereas the ISP industry appeared to manage quite well with a far simpler system of peering and tiering that avoided putting a value on individual packets or flows[15]. This relationship was never going to last, and it resolved itself in ways that in retrospect were quite predictable. From the telco perspective it quickly became apparent that the only reason the telco was being pushed to install additional network capacity at ever increasing rates was the requirements of the ISP sector. From the ISP perspective the only way to grow at a rate that matched customer demand was to become one's own carrier and to take over infrastructure investment. And, in various ways, both outcomes occurred. Telcos bought ISPs, and ISPs became infrastructure carriers.

All this activity generated considerable investor interest, and the rapid value escalation of the ISP industry and then the entire Internet sector generated the levels of wild-eyed optimism that are associated only with an exceptional boom. By 2000 almost anything associated with the Internet, whether it was a simple portal, a new browser development, a search engine, or an ISP, attracted investor attention, and the valuations of Internet start-ups achieved dizzying heights. Of course one of the basic lessons of economic history is that every boom has an ensuing bust, and in 2001 the Internet bust happened. The bust was as inevitable and as brutal as the preceding boom was euphoric. But, like the railway boom and bust of the 1840s, when the wreckage was cleared away, what remained was a viable—and indeed a valuable—industry.

By 2003 the era of the independent retail ISP was effectively over. ISPs still exist, but those that are not competitive carriers tend to operate as IT business consultants that provide services to niche markets. Their earlier foray in to the mass market paved the way for the economies of scale that only the carrier industry could implement on the market.

But the grander aspirations of these larger players have not been met, and effective monopoly positions in many Internet access markets have not translated to effective control over the user's experience of the Internet, or anything even close to such control. The industry was already "unbundled," with intense competition occurring at every level of the market, including content, search, applications, and hosting. The efforts of the telco sector to translate their investment into mass-market Internet access into a more comprehensive control over content and its delivery in the Internet has been continually frustrated. The content world of the Internet has been reinvigorated by the successful introduction of advertiser-funded models of content generation and delivery, and this process has been coupled with the more recent innovations of turning back to the users themselves as the source of content, so that the content world is once again the focus of a second wave of optimism, bordering on euphoria.

And Now?

It has been a revolutionary decade for us all, and in the last 10 years the Internet has directly touched the lives of almost every person on this planet. Current estimates put the number of regular Internet users at 19 percent of the world's population.

Over this decade some of our expectations were achieved and then surpassed with apparent ease, whereas others remained elusive. And some things occurred that were entirely unanticipated. At the same time very little of the Internet we have today was confidently predicted in 1998, whereas many of the problems we saw in 1998 remain problems today.

What we have today is not the technical Internet we thought we were building a decade ago. It is not a coherent end-to-end network with clear signaling across commodity packet switching fabric, but a network that is replete with all forms of active middleware[16], from NATs to firewalls[17] and filters, including packet shapers, torrent detectors, *Voice over IP* (VoIP) blockers, and load balancers. It is neither a secure nor a safe network, but one that includes a continual barrage on end hosts in the form of more than a million different forms of viruses[18], worms, and assorted malware[19], as well as a barrage on users in the form of torrents of spam[20]. The network is a host to a litany of hostile attacks, including gigabit traffic swamping attacks, redirection, inspection, passing off, and denial-of service attacks[21]. The attacks are directed at links, routers[22], the routing protocols[23, 24], hosts, and applications. Our ability to effectively defend the network and its connected hosts continues to be, on the whole, ineffectual. Our level of interest in paying a premium to support highly secure systems still remains slight. But somehow we are not deterred by this situation. Somehow each of us has found a way to make our Internet work for us.

I am not sure that the next decade will bring the same level of intensity of structural change to the global communications sector, and perhaps that is a good thing given the collection of other challenges that are confronting us all in the coming decades. At the same time I think it would be good to believe that the past decade of development of the Internet has completely rewritten what it means to communicate, rewritten the way in which we can share our experience and knowledge, and, hopefully, rewritten the ways in which we can work together on these challenges.

References

The Internet Protocol Journal (IPJ) has published articles on all the major aspects of the technical evolution of the Internet over the past decade. To illustrate the extraordinary breadth of these articles, I have included as references here only articles that have been published in the IPJ.

[1] Stallings, W., "Mobile IP," *IPJ,* Volume 4, No. 2, June 2001.

[2] Handley, M., and Crowcroft, J., "Internet Multicast Today," *IPJ,* Volume 2, No. 4, December 1999.

[3] Stallings, W., "IP Security," *IPJ,* Volume 3, No. 1, March 2000.

[4] Huston, G., "QoS — Fact or Fiction?" *IPJ,* Volume 3, No. 1, March 2000.

[5] Stallings, W., "MPLS," *IPJ,* Volume 4, No. 3, September 2001

[6] Ferguson, P., and Huston, G, "What is a VPN?" *IPJ,* Volume 1, No. 1 & No. 2, June & September 1998.

[7] Fink, R., "IPv6," *IPJ,* Volume 2, No. 1, March 1999.

[8] Huston, G., "Anatomy: Inside Network Address Translators," *IPJ,* Volume 7, No. 3, September 2004.

[9] Huston, G., "The BGP Routing Table," *IPJ,* Volume 4, No. 1, March 2001.

[10] Huston, G., "Scaling inter-Domain Routing," *IPJ,* Volume 4, No. 4, December 2001.

[11] Huston, G, "Exploring Autonomous System Numbers," *IPJ,*Volume 9, No. 1, March 2006.

[12] Huston, G., "The Future for TCP," *IPJ,* Volume 3, No. 3, September 2000.

[13] Huston, G., "Gigabit TCP," *IPJ,* Volume 9, No. 2, June 2006.

[14] Huston, G., "ENUM," *IPJ,* Volume 5, No. 2, June 2002.

[15] Huston, G., "Peering and Settlements," *IPJ,* Volume 2, No. 1 & No. 2, March & June 1999.

[16] Huston, G., "The Middleware Muddle," *IPJ,* Volume 4, No. 2, June 2001.

[17] Avolio, F., "Firewalls and Internet Security," *IPJ,* Volume 2, No. 2, June 1999.

[18] Fraser, B., Rogers, L., and Pesante, L., "Was the Melissa Virus So Different?" *IPJ,* Volume 2, No. 2, June 1999.

[19] Chen, T., "Virus Trends," *IPJ,* Volume 6, No. 3, September 2003.

[20] Crocker, D., "Challenges in Anti-Spam Efforts," *IPJ,* Volume 8, No. 4, December 2005.

[21] Patrikakis, C., Masikos, M., and Zouraraki, O., "Distributed Denial of Service Attacks," *IPJ,* Volume 7, No. 4, December 2004.

[22] Lonvick, C., "Securing the Infrastructure," *IPJ,* Volume 3, No. 3, September 2000.

[23] Kent, S., "Securing BGP: S-BGP," *IPJ,* Volume 6, No. 3, September 2003.

[24] White, R., "Securing BGP: soBGP," *IPJ,* Volume 6, No. 3, September 2003.

[25] Meyer, D., "The Locator Identifier Separation Protocol (LISP)," *IPJ,* Volume 11, No. 1, March 2008.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. The author of numerous Internet-related books, he is currently the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of the Internet Society from 1992 until 2001. E-mail: **gih@apnic.net**

Mobile WiMAX

by Jarno Pinola and Kostas Pentikousis, VTT Technical Research Centre of Finland

One of the technologies that can lay the foundation for the next generation (fourth generation [4G]) of mobile broadband networks is popularly known as "WiMAX." WiMAX, *Worldwide Interoperability for Microwave Access,* is designed to deliver wireless broadband bitrates, with *Quality of Service* (QoS) guarantees for different traffic classes, robust security, and mobility. This article provides an overview of mobile WiMAX, which is based on the wireless local and *Metropolitan-Area Network* (MAN) standards IEEE 802.16-2004[1] and 802.16e-2005[2]. We introduce WiMAX and focus on its mobile system profile and briefly review the role of the WiMAX Forum. We summarize the critical points of the WiMAX network reference model and present the salient characteristics of the PHY and MAC layers as specified in [1] and [2]. Then we address how mobile nodes enter a WiMAX network and explain the fundamentals of mobility support in WiMAX. Finally, we briefly compare WiMAX with *High-Speed Packet Access* (HSPA), another contender for 4G.

The Role of the WiMAX Forum

The WiMAX Forum is a nonprofit organization formed in 2001 to enhance the compatibility and interoperability of equipment based on the IEEE 802.16 family of standards. The IEEE 802.16 standards provide a large set of fundamentally different options for designing a wireless broadband system, including, for example, multiple options for *Physical* (PHY) layer implementation, *Media Access Control* (MAC) architecture, frequency bands, and duplexing. So many options lead to several possible system variants, which are all compatible with the IEEE standards. Although such multiplicity allows for deployment in very diverse environments, it may spell either solely vertical, single-vendor deployments or no deployment at all, because operators do not want to be locked in with any particular implementation. Thus, a major motivation for establishing the WiMAX Forum was to develop predefined system profiles for equipment manufacturers, which include a subset of the features included in the IEEE 802.16 standards. WiMAX Forum-certified products are guaranteed to be interoperable and to support wireless broadband services from fixed to fully mobile scenarios. The aim is to enable rapid market introduction of new standard-compliant WiMAX equipment and to promote the use of the technology in different sectors.

From IEEE 802.16 to Mobile WiMAX

The IEEE 802.16 standard was originally meant to specify a fixed wireless broadband access technique for point-to-point and point-to-multipoint links. During its development, however, it was decided that mobility support should also be considered.

The WiMAX Forum defines two system profiles based on [1] and [2], called *fixed* and *mobile* system profiles, respectively. Both include mandatory and optional PHY and MAC layer features that are required from all corresponding WiMAX-certified products. Because [1] and [2] specify only the PHY and MAC layers, an end-to-end architecture specification was deemed necessary in order to enable fast growth in manufactured quantities, market share, and interoperability. In response, the WiMAX Forum established the *Network Working Group* (NWG) with the aim of developing an end-to-end network reference model architecture based on IP supporting both fixed and mobile WiMAX (refer to [3] and [4]).

In short, according to the NWG reference model, a WiMAX network is partitioned into three independent architectural components: the user equipment (also referred to as *Customer Premises Equipment* [CPE]), the *Radio Access Network* (RAN, based on IEEE 802.16), and the network providing IP connectivity with the rest of the Internet. Clearly, this model allows a single operator to freely mix and match offerings from different manufacturers for these three parts, at least after interoperable equipment becomes readily available. Furthermore, in principle, each of these components of an operational network can be deployed and managed by different service providers. This scenario makes the network architecture flexible, eases network operation and maintenance, can increase competition under certain conditions, and is conducive to new business models. For example, municipalities can venture jointly with local or national network operators to deploy WiMAX in suburban and rural areas.

In contrast with earlier wireless data networks[5], IP is fundamental in a WiMAX network. Indeed, IP currently plays a dominant role in the present state of the telecommunications industry. The premise is that by embracing IP, service providers and equipment manufacturers will face fewer problems when introducing WiMAX into their networks and product portfolios. Moreover, protocols standardized by the *Internet Engineering Task Force* (IETF) are preferred over proprietary solutions and are adopted as extensively as possible in the reference model.

Mobile WiMAX Network Reference Model

The WiMAX Forum NWG network reference model defines three basic architectural entities: the *Mobile Station* (MS), the A*ccess Service Network* (ASN), and the *Connectivity Service Network* (CSN). The role of the MS is to provide user access to the WiMAX network. The ASN is the Radio Access Network and is formed by numerous *Base Stations* (BSs) and *ASN Gateways* (ASN-GWs), managed by a *Network Access Provider* (NAP). CSN is the network entity providing IP connectivity to the WiMAX radio equipment, including all the IP core network functions required for internetworking with the rest of the world. CSNs are maintained by *Network Service Providers* (NSPs).

The ASN and CSN are further broken up into smaller functional entities, which communicate with each other using standardized interfaces called *reference points*. These reference points guarantee that a certain set of protocols and procedures are always supported and can function irrespective of the underlying hardware. The currently defined reference points are used for different control and management purposes, as well as for data bearing between the network entities. Figure 1 illustrates the network reference model and the main reference points.

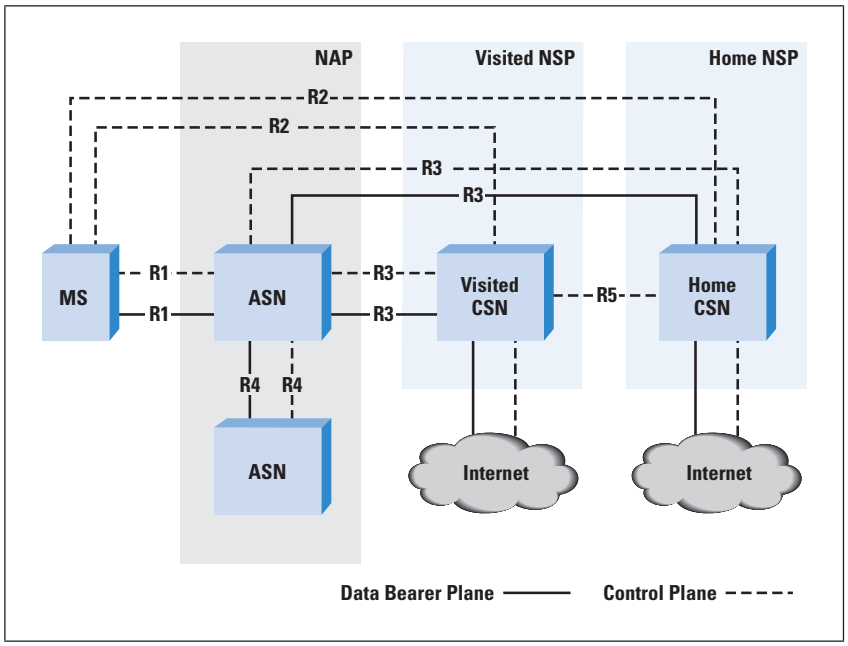


Figure 1: WiMAX Forum NWG Network Reference Model

The reference points are defined as follows in [3]: Reference point R1 consists of protocols and procedures compliant to [1], [2], and [6]. R1 implements the specifications of the air interface between the MS and the BS. R2, an interface between the MS and a CSN, is used solely for management purposes, including mobility management. R3 serves the same purpose between an ASN and a CSN, and R4 is used for micromobility management between two ASNs. R5 enables interworking between two CSNs for macromobility management.

In addition to reference points R1–R5, another three intra-ASN reference points are defined (not illustrated in Figure 1). R6, which consists of a set of control- and bearer-plane protocols for BS and ASN-GW communication, controls the data path and MS mobility events between these two ASN entities. R7 is an optional set of protocols used for coordinating R6 functions. Finally, R8 consists of bearer-plane protocols that enable data transfer between the base stations involved in a handover (also called *handoff*).

With respect to mobility, the reference model considers two different scenarios called *ASN-anchored mobility* and *CSN-anchored mobility.* ASN-anchored mobility (or intra-ASN mobility, or micromobility) management is employed when MS handovers occur from one BS to another, and both are controlled by the same ASN-GW. On the other hand, CSN-anchored mobility (or inter-ASN mobility, or macromobility) management is employed when MS movement dictates a handover from the currently serving BS to another one that is in a different subnetwork, controlled by a different ASN-GW. In the ASN-anchored case, handovers are managed solely by the MS and the ASN. In the CSN-anchored case, both ASN and CSN entities are engaged in mobility management.

Typically, ASN-anchored mobility procedures take precedence and CSN-anchored mobility management is employed only if necessary. Because ASN-anchored mobility takes place inside a single ASN, it does not change the MS network layer (IP) configuration. Three different functions are specified for ASN-anchored mobility management, all considered peer-to-peer interactions between different architectural entities:

- The *handoff* (HO) function controls the handover decision operation and handover signaling. The HO function supports mobile- and network-initiated handovers and, additionally, it may support *Fast Base Station Switching* (FBSS) or *Macro Diversity Handover* (MDHO)[2].
- The *Data Path* (DP) function manages the data path setup and data packet transmission between two functional entities.
- The context function addresses the exchanges required in order to retrieve or set up any state in the network elements.

On the other hand, when MS movement necessitates CSN-anchored mobility management, the MS IP layer configuration changes as a result of the handover. In this case, mobility management is based on *Mobile IPv4* (MIPv4)[7] or *Mobile IPv6* (MIPv6)[8], if the MS supports it. Alternatively, the reference model adopts *Proxy MIP* (PMIP)[9] to handle the handover. In PMIP, the MIP function is moved from the MS to a network instance called a *PMIPv4 client,* which takes care of all MIP signaling on behalf of the MS. Support for PMIP is specified only for MIPv4 in [3] and [4]. Note that in a handover from one ASN to another, MIP is used to complement ASN-anchored mobility management. The latter is still necessary to control the link-layer handover procedures. That is, after the micromobility handover is successfully completed, MIP independently takes care of the macromobility handover, that is, establishes communication paths between the new ASN-GW and the CSN. CSN-anchored mobility handovers are always network-initiated.

By embracing IETF protocols and providing an end-to-end architecture with independent functional entities, the WiMAX Forum NWG network reference model provides a clear framework for the application developers to work in. The model provides only operational requirements and does not prescribe particular technical solutions to realize them, allowing for proprietary yet standards-compliant implementations and enabling technical competition between different manufacturers.

Before examining mobility support in WiMAX, we review the basics of the IEEE 802.16 PHY and MAC layers.

OFDM and OFDMA

IEEE 802.16 and thus WiMAX adopted *Orthogonal Frequency Division Multiplexing* (OFDM), a multicarrier modulation scheme, as its PHY layer. In OFDM, the available bandwidth is divided into several parallel orthogonal subcarriers with lower bandwidth. A wideband channel is defined as a group of adjacent narrowband channels: a high-bitrate data stream is divided into these subcarriers and multiple narrowband data streams are transmitted over the air. Because the data symbol duration is inversely proportional to bitrate, the transmitted symbol duration is increased and the level of *Inter-Symbol Interference* (ISI) can be reduced. ISI is caused by multipath propagation in the wireless communication medium, where the transmitted data symbols can arrive at the receiver through different propagation routes because of reflections from buildings in urban areas and from hills and trees in rural areas. OFDM also uses guard intervals between successive data symbols and cyclic prefixes in order to decrease the effect of ISI even more.

One reason for the wide adoption of OFDM in modern broadband communication systems is its hardware implementation simplicity. OFDM signals can be formed and processed using *Inverse Fast Fourier Transform* (IFFT) and *Fast Fourier Transform* (FFT), at the transmitter and receiver, respectively, and both transforms can be implemented directly in hardware for higher performance. OFDM bodes well for mobile broadband systems through frequency diversity and adaptivity in both modulation and channel coding. By using *Adaptive Modulation and Coding* (AMC), the end-to-end quality deterioration due to the excess delays and deep fading conditions caused by mobility can be prevented, or at least diminished.

OFDM can also be used as a multiaccess scheme by having subcarriers grouped into subchannels, which can be assigned to different users contending for the data link. Each subchannel can contain a different number of subcarriers, and by altering the subcarrier group sizes and observing the channel conditions, it is possible to use differentiation in the channel allocation for different users.

This technique of using OFDM as a multiaccess scheme is called *Orthogonal Frequency Division Multiple Access* (OFDMA). Mobile WiMAX uses OFDMA as its PHY layer instead of plain OFDM, and subchannelization to both uplink and downlink transmissions is possible.

In OFDMA, the subcarriers assigned to subchannels can be either concurrent or taken from different regions of the total bandwidth. Both of these allocation schemes have advantages. When subcarriers assigned to one subchannel are distributed over the available bandwidth, frequency diversity can be attained. In mobile systems this diversity is advantageous because it can be used to make the transmission link more resistant against fast fading. A subchannelization scheme based on dispersed subcarrier allocation to subchannels, called *Partial Usage of Subcarriers* (PUSC), is mandatory in all mobile WiMAX implementations.

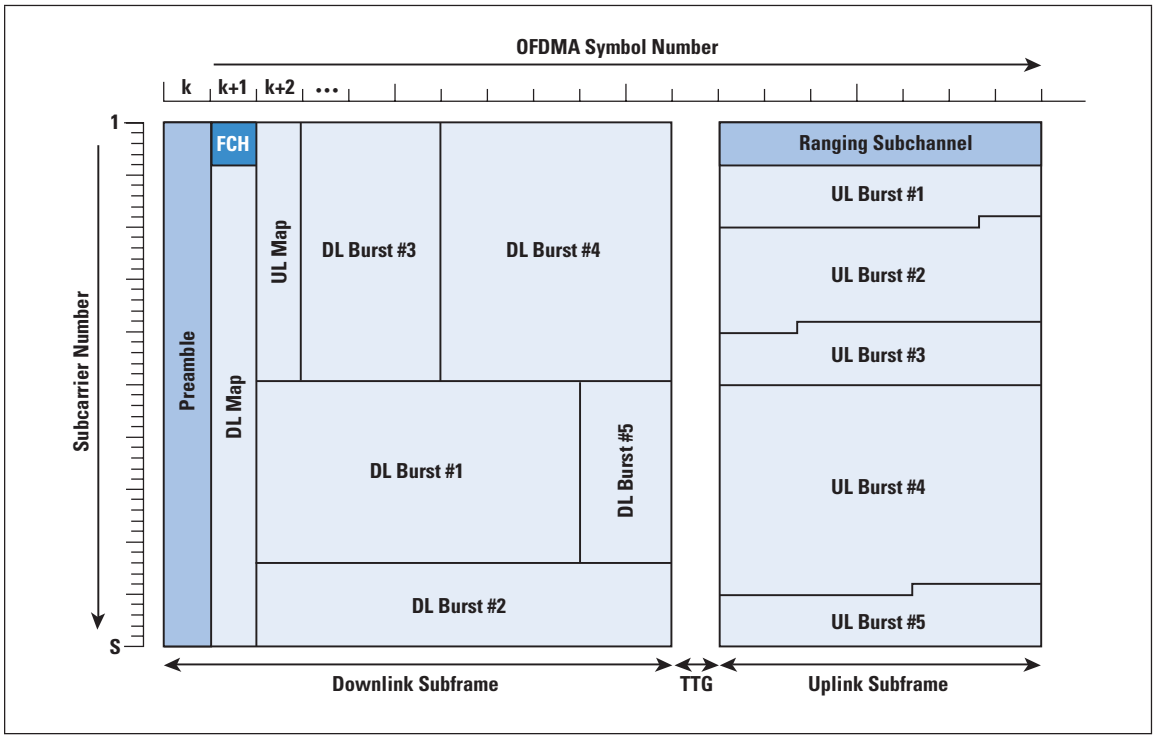
WiMAX systems can use *Time-Division Duplexing* (TDD) or *Frequency-Division Duplexing* (FDD) when allocating air interface resources to users. In TDD, the uplink and downlink transmissions are done over the same carrier frequencies and the separation between the transmission directions is done by assigning time slots, in which the transmission to one direction at a time is scheduled. In FDD, uplink and downlink transmissions are done simultaneously over different carrier frequencies.

Commonly used in mobile WiMAX equipment, TDD allows more flexible sharing of the available bandwidth between the uplink and downlink transmissions. On balance, TDD requires synchronization between multiple adjacent base stations so that transmissions in neighboring cells do not interfere with each other. A TDD frame (Figure 2) is divided into two subframes: first comes a downlink frame and after a short guard interval, called the *Transmit/Receive Transition Gap* (TTG), an uplink frame follows in the same frequency band. Each downlink subframe starts with a preamble, which is used for synchronization and channel estimation. To enhance tolerance against mobility-inflicted channel impairments, WiMAX allows optional support for a more frequent preamble repetition during transmission. In the uplink, short preambles, also called *midambles,* can be used after 8, 16, or 32 OFDM symbols, and in the downlink, short preambles in front of every data burst can be used. After the preamble comes a *Frame Control Header* (FCH), which consists of uplink and downlink *Media Access Protocol* (MAP) messages, which inform users about their transmission parameters.

Flexible data multiplexing from different users into one OFDM or OFDMA frame is also supported, as illustrated in Figure 2. Both uplink and downlink subframes can include data bursts of different types from multiple users, and they can be of variable length.

A small portion of the uplink subframe is reserved for transmission parameter adjustment and bandwidth request purposes. Moreover, small amounts of user data can be sent in this portion of the uplink subframe. The total OFDM frame size can range between 2.5 and 20 ms, but the initially supported frame size in present WiMAX equipment is 5 ms.

Figure 2: An example of a WiMAX OFDMA Frame

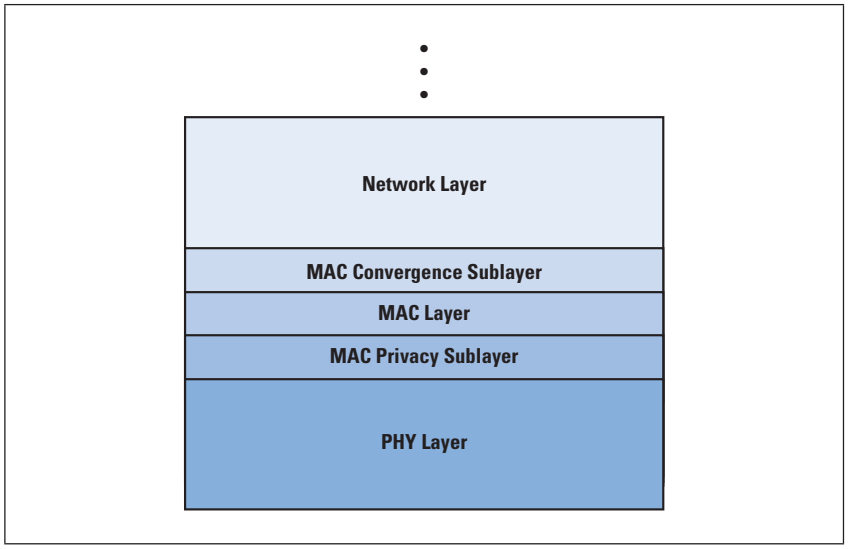


Media Access Control

The MAC layer is primarily an adaptation layer between the PHY layer and the upper layers. Its most important task, when transmitting data, is to receive *MAC Service Data Units* (MSDUs) from the layer above, aggregate and encapsulate them into *MAC Protocol Data Units* (MPDUs), and pass them down to the OFDM or OFDMA PHY layer for transmission. When data is received, the MAC layer takes MPDUs from the PHY layer, decapsulates and reorganizes them into MSDUs, and passes them on to the upper-layer protocols.

An additional layer between the MAC and upper protocol layers called the *Convergence Sublayer* (CS) is also defined in [1] and [2] and illustrated in Figure 3. For the upper layers, CS functions as an interface to the MAC layer. Even though in principle a CS is presented for a variety of different protocols, currently [3] and [4] support CS only for IP and Ethernet. Other protocols can, of course, use these CSs through encapsulation. The CS may also support upper-protocol header compression.

Figure 3: WiMAX Protocol Stack



Similarly with the PHY layer, shown in Figure 3, the MAC layer allows flexible allocation of transmission capacity to different users. Variably sized MPDUs from different flows can be included into one data burst before being handed over to the PHY layer for transmission. Multiple small MSDUs can be aggregated into one MPDU and, conversely, one big MSDU can be fragmented into multiple small ones in order to further enhance system performance. For example, by bundling up several MPDUs or MSDUs, the PHY and MAC layer header overheads, respectively, can be reduced.

It is important to remember that the BS MAC layer manages bandwidth allocation for both uplink and downlink transmissions. The BS assigns bandwidth for the downlink transmission according to incoming network traffic. For the uplink transmission, bandwidth is allocated based on the requests received from the MS. Because basically all connections are controlled by the BS, QoS can be efficiently implemented into WiMAX equipment. Currently, the MAC layer of a mobile WiMAX BS should include support for five different QoS classes, briefly summarized in Table 1.

Table 1: Mobile WiMAX QoS Classes

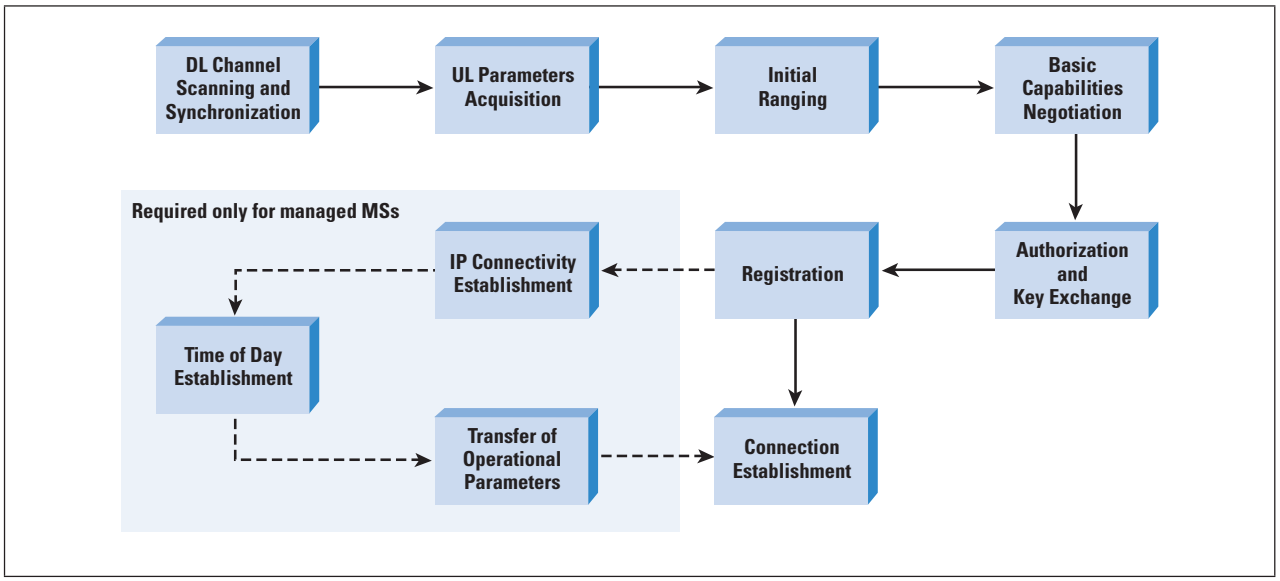
QoS Class	Supported Service	Example Application
Unsolicited Grant Services (UGS)	Latency- and jitter-sensitive applications with fixed-size data packets at Constant Bitrate (CBR)	Voice over IP (VoIP) without silence suppression
Real-Time Variable Rate (RT-VR)	Real-time applications with variable-size data packet bursts	Video and audio streaming
Non-Real-Time Polling Services (nrtPS)	Delay-tolerant applications with variable-size data packets and guaranteed bitrate demands	File transfers
Extended Real-Time Variable Rate (ERT-VR)	Real-time applications with Variable Bitrate (VBR) data streams and guaranteed bitrate and delay demands	VoIP with silence suppression
Best Effort (BE)	Data streams with no minimum service-level demands	Web browsing, instant messaging, and data transfer

Prior to any data transmission over a WiMAX link, the MS and the BS must form a unidirectional connection between their respective MAC layers. A unique identifier, called *Connection Identifier* (CID), is assigned to each uplink and downlink connection pair. The CID serves as a temporary address for the transmitted data packets over the WiMAX link. Another identifier, called *Service Flow Identifier* (SFID), is assigned by the BS to unidirectional packet flows with the same QoS parameters, that is, service flows. The BS also handles the mapping of SFIDs to CIDs in the QoS control process. Note that the MAC layer incorporates sophisticated power-management techniques and robust, state-of-the-art security features, but these features are out of scope for this article.

Network Entry and Reentry

Figure 4 illustrates the basic steps that every MS must go through when entering or reentering a WiMAX network. First, a MS scans the downlink channel and synchronizes with the BS, after which the MS acquires the transmit parameters for the uplink transmission from the BS *Uplink Channel Descriptor* (UCD) message and performs initial ranging, hence acquiring the correct timing offset and power adjustments. A MS extracts an initial ranging-interval time slot from an uplink MAP message. If a MS cannot complete the initial ranging successfully, it must start scanning for a new downlink channel.

Figure 4: Network Entry/Reentry Procedure



The basic capabilities negotiation process starts when the MS sends a message containing its capabilities to the BS; the BS responds with a message containing the capabilities it has in common with the MS. If *Privacy Key Management* (PKM) is enabled at both the MS and the BS, the next step is to perform the authorization and key-exchange procedure, so that the MS can register with the network. The BS sends back a registration response message that contains the secondary management CID, if the MS is managed.

After a managed MS obtains this secondary management CID, it becomes "manageable." The successful reception of the registration response message is a prerequisite for any MS in order to be able to transmit to and receive from the network.

When a managed MS enters the network, the next step is to establish IP connectivity by using the assigned secondary management connection and by either invoking the *Dynamic Host Configuration Protocol* (DHCP)[10] or DHCPv6[11], or using the IPv6 stateless address autoconfiguration[12], depending on the information provided by the BS registration response message. If the MS uses MIPv4 or MIPv6, it can secure its address by using the secondary management connection with MIP. The establishment of IP connectivity and time of day, as well as the transfer of the operational parameters, are needed only for managed MSs. These parameters can be managed with IP management messages through a secondary management connection, for example, by using the DHCP, *Trivial File Transfer Protocol* (TFTP)[13], or *Simple Network Management Protocol* (SNMP)[14]. These additional steps during network entry are necessary for the operation of the IP management protocols.

If DHCP is used to establish IP connectivity, a managed MS must also establish the time of day so that the management system can time-stamp certain events. Both the MS and the BS must be set at the same time of day, with an accuracy of the nearest second. The time of day is retrieved using the secondary management connection with the *Time Protocol*[15]. The current time is formed by combining the time retrieved from the server with the time offset extracted from the DHCP reply message. Although the time of day is not needed for the registration to complete successfully, it is required in order to keep the connection operational. Finally, the managed MS must acquire its operational parameters with TFTP.

After a managed MS has obtained its operational parameters, or after an unmanaged MS has registered with the network, the MS preprovisioned service-flow connections are established.

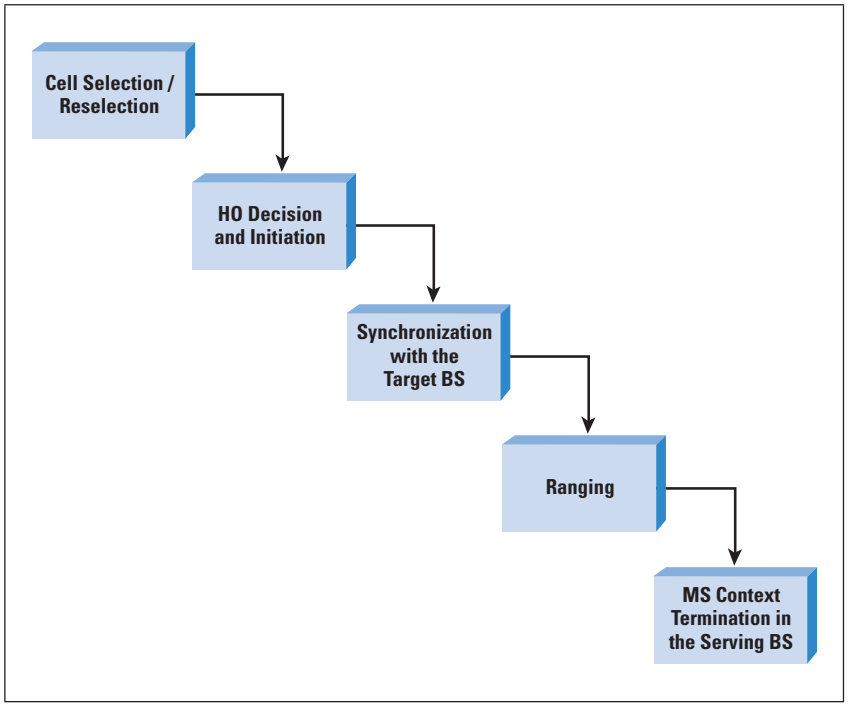
Mobility Support

As discussed previously, IEEE 802.16e introduced mobility support, defining an OFDMA PHY layer and signaling mechanisms to enable location and mobility management, paving the way for mobile WiMAX. The WiMAX Forum details four mobility scenarios in addition to the fixed WiMAX scenario. In the nomadic and portable mobility scenarios, the point of attachment of a fixed *Subscriber Station* (SS) can change. The simple mobility scenario allows MSs to roam within the coverage area with speeds up to 60 km/h, but handovers may cause connection interruptions of up to 1 second. In the so-called *full-mobility scenario,* the MS speed can be as much as 120 km/h, and transparent handovers are supported. This last scenario is what many might consider as the real mobile WiMAX scenario, but all five scenarios are "standards-compliant."

Although three different types of handovers are defined in [2], *Hard Handover* (HHO), *Macro Diversity Handover* (MDHO), and *Fast Base Station Switching* (FBSS), only HHO is mandatory for all mobile WiMAX equipment. This type of handover is often referred to as a *break-before-make handover:* first, the MS disconnects from the serving BS and then connects to the target BS. Because of the short disconnection period, packets may be lost; HHO is less sophisticated than either MDHO or FBSS and may be inappropriate for some applications. The MS must also register with the target BS and reauthenticate with the network, typically meaning further delays before actual data exchange can (re)start. If multiple handover types are supported and enabled, the BS decides which type should take precedence over the other. MDHO and FBSS are enabled or disabled during the registration of the MS with the BS.

Figure 5 illustrates the five stages of a successful HHO in mobile WiMAX. The first stage is to select the target BS cell based on information about the network topology surrounding the serving BS through periodically broadcasted neighbor advertisements. The advertisements include the same information on the serving BS neighbors that the *Downlink Channel Descriptor* (DCD) and *Uplink Channel Descriptor* (UCD) messages of the neighboring BSs would include. For example, a neighbor advertisement message includes channel information of the neighboring BSs so that the MS can synchronize with them and perform scanning operations to evaluate their suitability as potential targets for a HO.

Figure 5: The Five Phases of a Successful HHO



The second phase is to make the actual decision to initiate the handover procedure, when a certain network (say, congestion in the serving cell requires load balancing) or channel condition threshold (for example, low received *Signal-to-Interference + Noise Ratio* [SINR] in the current cell) is crossed. The actual decision to start the message exchange for the MS to migrate from the radio interface of the serving BS to the radio interface of another BS can be made by the MS, BS, or the network. In the third phase, the MS synchronizes with the downlink transmission of the target BS and obtains the transmission parameters for the downlink and the uplink. The time consumed to perform the synchronization procedure depends on the amount of information the MS received about the target BS in the neighbor advertisement messages prior to the handover. The average synchronization latency without previously acquired information about the target BS ranges from two to three frame cycles, or approximately 4 to 40 ms depending on the OFDMA frame duration used in the system. The more extensive the channel parameter list received in the neighbor advertisement messages prior to the handover, the shorter the time to achieve the synchronization.

After synchronizing, the MS and the target BS initiate the ranging procedure. During this fourth step in HHO, MS and BS exchange the required information so that the MS can reenter the network. The target BS can request information about the MS from the (previously) serving BS and other network entities. Again, the more information made available to the target BS, the shorter the time to reenter the network, because the target BS may skip some steps from the network (re)entry procedure described earlier. In short, sharing context information before the actual handover optimizes the handover procedure and decreases its latency. In the last step of a HHO, the MS context at the serving BS is terminated and resources are released.

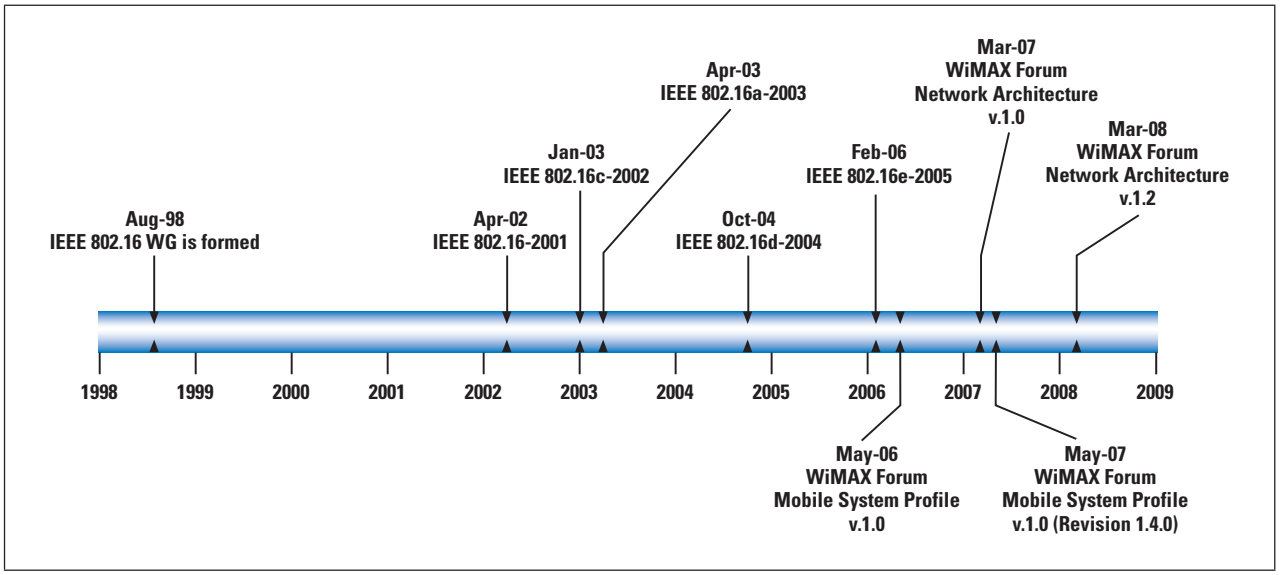
If MDHO and FBSS are supported, the following stages, in addition to those already described in the HHO procedure, must be performed: (a) decision to enable MDHO or FBSS, (b) diversity set update, and (c) anchor BS selection. In macrodiversity communications the MS maintains a connection to one or more serving BSs simultaneously, enabling soft or make-before-break handovers. In [2], the transition of the MS from the air interface of one or more serving BSs to the air interface of one or more target BSs is referred to as a MDHO. The MS and the BS both maintain a list called the *diversity set,* which includes all serving BSs involved in the MDHO communication. The MS maintains both uplink and downlink unicast connections to all the BSs in the diversity set, and one of the serving BSs is defined as the anchor BS. Note that all BSs involved in the diversity set use the same set of CIDs for the connections established between the MS and the serving BSs.

In FBSS, the MS transmits to and receives data from a single serving BS during any frame period. The BS, to which the MS has the connection to at any given frame, is called the *anchor BS*. The MS maintains a diversity set, which includes all active BSs in its range, and can change its anchor BS on a frame-by-frame basis, based on certain criteria. The transition from the serving anchor BS to the target anchor BS in FBSS is done without invocation of the normal handover procedure, and only the anchor BS update procedure is needed. After all, the MS has collected all required information about all BSs during the diversity set update ranging procedures.

Mobile WiMAX vs. HSPA

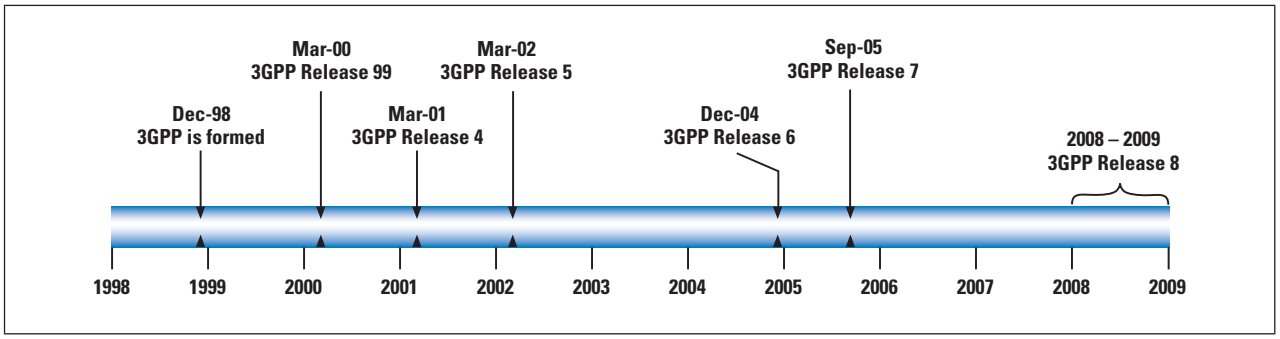
Mobile WiMAX and *High-Speed Packet Access* (HSPA) are expected to be the two major contestants in the rapidly growing wireless broadband market. The two, however, come from different origins. Figure 6 summarizes the evolution toward mobile WiMAX. It all started with the establishment in August 1998 of the IEEE 802.16 working group, which published its first standard (IEEE 802.16-2001) in April 2002. This first version defines a single carrier system operating in the 10- to 66-GHz frequency band and only under *line-of-sight* (LOS) conditions. The IEEE 802.16c-2002 amendment detailed system profiles for the original standard based on the 10- to 66-GHz frequency band. IEEE 802.16a-2003 introduced support for 2- to 11-GHz frequencies and *non-line of sight* (NLOS) operation, and adopted the use of OFDM and OFDMA. IEEE 802.16d-2004[1] consolidated all these previous versions and amendments in a single document, and further enhanced the system. Fixed WiMAX is based on IEEE 802.16d-2004, [3], and [4]. Mobile WiMAX is based on the IEEE 802.16e-2005 amendment[2], which introduced mobility support, as well on [3] and [4].

Figure 6: The Road Toward Mobile WiMAX



HSPA is a set of technological enhancements to the already widely deployed *Wideband Code Division Multiple Access* (WCDMA) cellular networks defined by the *Third Generation Partnership Project* (3GPP). Figure 7 illustrates the WCDMA specification evolution. The origins of HSPA can be traced in the foundation of 3GPP in December 1998. The original aim of 3GPP was to develop a third-generation WCDMA system, and in the process, HSPA was introduced. In March 2000, Release 99, the original standard specifying the WCDMA system, was published. A year later, the first enhancements were published in Release 4, which introduced, among others, an IP-based core network. Release 5 introduced *High-Speed Downlink Packet Access* (HSDPA) and defined the 3GPP *IP Multimedia Subsystem* (IMS). *High-Speed Uplink Packet Access* (HSUPA) and some further improvements to HSDPA were defined in Release 6 (December 2004). Release 7 further enhanced QoS support and defined mechanisms to decrease network latency. Release 8 is expected to be published in 2008, and it will include specifications for the next step, called 3GPP *Long-Term Evolution* (LTE). LTE is meant to deliver maximum cell throughputs an order of magnitude larger than HSPA.

Figure 7: The Evolution of the 3GPP WCDMA Standard



Mobile WiMAX evolved out of a broadband wireless LAN/MAN technology, and vendors currently report that it can deliver maximum cell capacities of 46 and 7 Mbps in downlink and uplink transmissions, respectively. However, mobility management is a later addition and, according to Maravedis, by September 2007 only 12 percent of all deployed *Customer Premises Equipment* (CPE) was IEEE 802.16e-2005-compliant[16]. On the other hand, HSPA is based on a solid foundation of mobility management techniques with wide deployment in cellular networks around the globe, but can currently deliver maximum cell throughputs of only 14.4 and 5.8 Mbps in downlink and uplink transmissions, respectively.

Either commercial or trial networks of both technologies have already been implemented all over the world. However, according to the *Global Mobile Suppliers Association* (GSA), HSPA networks have yet to be deployed in China and India, both of which are large and rapidly growing market areas for wireless communications. According to Maravedis, both India and China have at least WiMAX trial deployments in place.

As mentioned already, the vast majority of current WiMAX deployments do not support mobility. Up to now, fixed WiMAX has been used mainly for last-mile broadband connectivity for sparsely populated rural areas. The largest commercial IEEE 802.16e-2005-compliant system is currently the *Wireless Broadband* (WiBro)[17] network in South Korea, which supports simple mobility up to 60 km/h. Even though WiMAX and WiBro are both based on the same standards, WiBro was developed by the South Korean telecommunications industry before the WiMAX Forum adopted mobility support for its system profiles. WiMAX and WiBro are often cited as separate technologies, even though cooperation is in place in order to assure interoperability between the two.

Summary

In this article we presented an overview of mobile WiMAX, a much-heralded technology for next-generation mobile broadband networks; mobile WiMAX is an intricate system. We introduced WiMAX and the role of the WiMAX Forum, and summarized the important points of the WiMAX network reference model and the PHY and MAC layers. We addressed mobility support, but not the security aspects. Finally, we briefly compared WiMAX with HSPA, presenting their respective evolutions and illustrating their worldwide deployments. We hope that this article will serve as a valuable primer, and we highly recommend that those interested in the mobile WiMAX technology check the bibliography.

Bibliography

[1] IEEE 802.16 Working Group, "IEEE Standard for Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems," IEEE Standard 802.16-2004, October 2004.

[2] IEEE 802.16 Working Group, "IEEE Standard for Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands," IEEE Standard 802.16e-2005, February 2006.

[3] WiMAX Forum Network Working Group, "WiMAX Forum Network Architecture—Stage 2: Architecture Tenets, Reference Model and Reference Points—Release 1, Version 1.2," WiMAX Forum, January 2008.

[4] WiMAX Forum Network Working Group, "WiMAX Forum Network Architecture—Stage 3: Detailed Protocols and Procedures—Release 1, Version 1.2," WiMAX Forum, January 2008.

[5] K. Pentikousis, "Wireless Data Networks," *The Internet Protocol Journal,* Volume 8, No. 1, March 2005, pp. 6–14.

[6] IEEE 802.16 Working Group, "IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. Amendment 3: Management Plane Procedures and Services," IEEE Standard 802.16g-2007, December 2007.

[7] C. Perkins (Ed.), "IP Mobility Support for IPv4," RFC 3344, August 2002.

[8] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775, June 2004.

[9] K. Leung, G. Dommety, P. Yegani, and K. Chowdhury, "WiMAX Forum/3GPP2 Proxy Mobile IPv4," Internet-Draft, Work in Progress.

[10] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131, March 1997.

[11] R. Droms (Ed.), J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3315, July 2003.

[12] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration," RFC 2462, December 1998.

[13] K. Sollins, "The TFTP Protocol (Revision 2)," RFC 1350, July 1992.

[14] J. Case, M. Fedor, M. Schoffstall, and J. Davin "A Simple Network Management Protocol (SNMP)," RFC 1157, May 1990.

[15] J. Postel and K. Harrenstien, "Time Protocol," RFC 868, May 1983.

[16] K. Pentikousis, J. Pinola, E. Piri, F. Fitzek, T. Nissilä, and I. Harjula, "Empirical Evaluation of VoIP Aggregation over a Fixed WiMAX Testbed," Proceedings of The 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TRIDENTCOM), 18–20 March, 2008, Innsbruck, Austria.

[17] Telecommunications Technology Association, "Specifications for 2.3GHz Band Portable Internet (WiBro™) Service," TTA Standard TTAS.KO-06.0082/R1, December 2005.

JARNO PINOLA received his M.Sc. from the University of Oulu, Oulu, Finland, in Spring 2008. During his studies, he specialized in telecommunication systems and wrote his Master's Thesis on mobility management issues in wireless broadband systems. Currently he is working as a Research Scientist at VTT Technical Research Centre of Finland in Oulu, Finland. He can be contacted via e-mail at: **jarno.pinola@vtt.fi**

KOSTAS PENTIKOUSIS studied computer science at Aristotle University of Thessaloniki, Thessaloniki, Greece (B.Sc. 1996), and Stony Brook University, Stony Brook, New York, USA (M.Sc. 2000, Ph.D. 2004). He is a tenured Senior Research Scientist at VTT Technical Research Centre of Finland, in Oulu, Finland. He has published internationally in several areas, including mobile computing (mobility triggers, multiaccess, media-independent handovers, and energy consumption); transport protocols; applications; network traffic measurements and analysis; and simulation and modeling. Visit **http://ipv6.willab.fi/kostas** for more information and contact details.

Letters to the Editor

IDNs

The DNS protocol is 8-bit clean ("Internationalizing the Domain Name System," IPJ, Volume 11, No. 1, March 2008), even if some DNS clients and servers are not. The hardest thing about changing any Internet protocol is coordinating clients and servers during the transition.

And yet, with the DNS, no transition is needed to support UTF-8 domain names. If you want to publish a UTF-8 domain name, then run a name server that supports UTF-8. If you want to be able to access domain names in your own language, switch to DNS software that supports it. Implementations that are 8-bit clean are already available; ordinary market mechanisms will handle the rest.

Punycode is a gross hack that makes my stomach roil. You know it, I know it, any engineer will agree with you, so how did it get through the IETF?

The argument for where to stop internationalization does not spread to `protocol://` because it's "gobble-de-gook" in English, too. Dots are a completely arbitrary character used to separate the hierarchy. There's plenty of space at the top for UTF-8 names.

The real problem with IDN is homoglyphs.

—Russ Nelson,
`nelson@crynwr.com`

The author responds:

It would certainly make more sense in terms of design elegance and minimalism within the DNS if the label that was stored in the DNS was precisely the same label that was used in the interface between applications and the DNS client software. There is something rather clumsy about the approach that stores an encoded version of a canonical version of the label value, and relies on the application being capable of performing the *stringprep* and encoding functions in consistent and uniform ways. The resultant limitations on what can actually sit in DNS labels on a language-by-language basis are, in part, an outcome of the potential indeterminism of this canonicalization function.

But indeterminism is not a tolerable outcome of the DNS. The DNS is not a guessing game, and inconsistencies in the mapped transforms that are provided by the DNS trigger intolerable insecurities in the networked environment. So the *nameprep* profiles and the related restrictions on allowable Unicode code points are unavoidable if we want to avoid this indeterminism in the DNS.

So if *nameprep* is required in any case, then what we are left with to consider is the decision to use the Punycode *ASCII Compatible Encoding* (ACE) to map Unicode labels into the *Letter-Digit-Hyphen* (LDH) subset of ASCII. But is the Punycode ACE really that much of a problem? Within the overall IDN framework the Punycode algorithm is not so complex that the risk of incorrect implementations is significant, the algorithm is not processor-intensive, and the outcome does not inflate the encoded labels to an impossible length. The advantage of Punycode is that the DNS servers do not require modification, and the clients that manipulate IDNs required additional *nameprep* functions in any case, so Punycode was evidently intended to be the least-impact approach that spared DNS servers from a potential requirement for modification.

To me, this solution appears to be a design tradeoff, in so far as the ACE approach circumvents the observed problem of non-8-bit clean DNS servers sitting within the deployed DNS, and does not in and of itself demand novel roles and functions on the part of the clients of the DNS in addition to what was already necessitated by the IDN *nameprep* function. However, at the same time it creates an annoying inconsistency in the overall framework of the design of the DNS, where certain labels in the DNS are intended to trigger a Punycode transform into an equivalent Unicode string while other labels are meant to be used without further transforms applied.

My judgment of the short-term path of least risk sits with the ACE approach as adopted for IDNs, but at the same time I agree with Russ' discomfort that the path that preserves the long-term essential broad utility and function of the DNS through consistency of design and application sits in an 8-bit clean DNS without the adornment of any form of an ACE.

And, yes, I agree with Russ that the most significant problem with IDNs is homoglyphs, because of continued reliance of an underlying approach of "appearance is everything" in terms of the integrity of the DNS as an identity framework.

—Geoff Huston,
gih@apnic.net

More IDNs

The LDH restriction referred to in "Internationalizing the Domain Name System" (IPJ, Volume 11, No. 1, March 2008) was relaxed in RFC 1123[1] to allow a host name to begin with either a letter or a digit.

—Andrew Friedman

[1] R. Braden, Editor, "Requirements for Internet Hosts – Application and Support," RFC 1123, October 1989.

The author responds:

My thanks to Andrew for pointing this out. It has been commonly recounted that this relaxation of the LDH convention was associated with the successful registration of the DNS name **3com.com** and that the RFC paperwork was revised following this registration. Since then the most visible set of names that used this "liberal" revision of LDH with names that have leading digits were telephone number mapping name sets, including the venerable **tpc.int** domain of the early 1990s and, more recently, ENUM. As for names with leading hyphens, I don't believe that we are at the point of allowing Morse code into the DNS yet, but I'm sure that someone somewhere is working on it!

—Geoff
(--. . --- ..-. ..-.)

We want to hear from You

Your feedback is important to us. Please send your comments and suggestions to **ipj@cisco.com**. And don't forget to visit our Website at **http://www.cisco.com/ipj** where you can read or download back issues, update and renew your subscription, and find articles using our index files. We also encourage you to participate in our online forum at **http://ipjforum.org**

Fragments

OLSR stands for *Optimized Link State Routing Protocol.*

DUMBO

The *Digital Ubiquitous Mobile Broadband OLSR* (DUMBO) project deploys mobile wireless networks on an ad hoc basis for emergency conditions, such as after a natural disaster when a fixed network infrastructure is not available.

A *Mobile ad hoc Network* (MANET) consists of mobile nodes that automatically cooperate to support the exchange of information through wireless medium. Since the MANET does not rely on fixed telecommunication infrastructure, it is suitable for emergency situations and can be set up in a short amount of time. Using lightweight portable mobile nodes, MANET coverage can penetrate deep into areas not easily accessible by roads or into areas where the telecommunication infrastructure has been destroyed.

DUMBO allows streaming video, *Voice over IP* (VoIP) and short messages to be simultaneously transmitted from a number of mobile laptops to a central command center, or to the other rescuers at the same or different disaster sites. The DUMBO command center has a face recognition module that identifies potential matches between unknown victims' face photos taken from the field and a collection of stored known face images. In addition, sensors can be deployed to measure environmental data such as temperature and humidity. Data from the sensors can be sent to the command center which analyzes or passes it on to the other mobile nodes. The command center can be located either in the disaster area or anywhere with Internet access. DUMBO technology is currently being deployed in cyclone-ravaged Burma. See `http://www.interlab.ait.ac.th/dumbo/` and `http://www.relief.asia/`

Upcoming Events

The *Internet Engineering Task Force* (IETF) will meet in Dublin, Ireland, July 27 – August 1 and in Minneapolis, Minnesota, November 16 – 21, see `http://www.ietf.org/`

APNIC, the *Asia Pacific Network Information Centre,* will hold its Open Policy meeting in Christchurch, New Zealand, August 25 – 29, see `http://www.apnic.net/meetings/26/`

[*Ed.: I will be organizing a pipe organ demonstration event on August 26 as part of the opening reception for APNIC 26, see* `http://organdemo.info`]

The *North American Network Operators' Group* (NANOG) will meet in Los Angeles, California, October 12 – 14. Immediately following the NANOG meeting, the *American Registry for Internet Numbers* (ARIN) will meet in the same location, October 15 – 17. See `http://nanog.org` and `http://arin.net`

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in Paris, France, June 22 – 26, and in Cairo, Egypt, November 2 – 7. See `http://icann.org`

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

September 2008 Volume 11, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

From The Editor

If you are reading the printed version of this journal you will notice a subtle change in the paper. This issue is printed on an uncoated stock, specifically Exact® Offset Opaque White 60#, a recycled paper made by Wausau Paper Corporation. This paper is slightly thinner, and thus lighter, than the paper we have been using. It is also less reflective and easier to write notes on. We invite your feedback on this paper as we experiment with various solutions to reduce our carbon footprint. As always, send your comments to: **ipj@cisco.com**

This journal has a long history of covering existing and emerging technologies that form part of the underlying infrastructure for both the global Internet and private enterprise networks. Recent articles have focused on wireless systems such as WiMAX, and we have other articles on wireless technologies in the pipeline. This time, however, we look at *optical networking*, specifically *Generalized Multiprotocol Label Switching* (GMPLS) as a technology for next-generation internets. The article is by Francesco Palmieri.

The topic of IP Version 4 address exhaustion has been discussed in several articles in this journal, and is currently being heavily debated in the *Regional Internet Registries* (RIRs). As we approach the inevitable date when the IPv4 address pool "runs out," we are returning to this topic with several articles. The first of these articles is included in this issue. Geoff Huston sets the stage by reviewing some of the history and answering the basic question of "why" we find ourselves at a point in history where the IPv4 addresses will run out before we have deployed any significant amount of IPv6 systems. In future issues we will follow Geoff's introduction with several other perspectives on this situation.

Once again, let me remind you to visit our Website at **http://www.cisco.com/ipj**, where you can renew and update your subscription, download back issues, and find additional resources such as our online forum at **http://ipjforum.org**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

GMPLS Control Plane Services in the Next-Generation Optical Internet

by Francesco Palmieri, Federico II University of Napoli, Italy

One of the major concerns in the Internet-based information society today is the tremendous demand for more and more bandwidth. Optical communication technology has the potential for meeting the emerging needs of obtaining information at much faster yet more reliable rates because of its potentially limitless capabilities—huge bandwidth (nearly 50 terabits per second[1]), low signal distortion, low power requirement, and low cost. The challenge is to turn the promise of optical networking into reality to meet our Internet communication demands for the next decade. With the deployment of *Dense Wavelength Division Multiplexing* (DWDM) technology, a new and very crucial milestone is being reached in network evolution. The speed and capacity of such wavelength switched networks—with hundreds of channels per fiber strand—seem to be more then adequate to satisfy the medium to long term connectivity demands. In this scenario, carriers need powerful, commercially viable and scalable devices and control plane technologies that can dynamically manage traffic demands and balance the network load on the various fiber links, wavelengths, and switching nodes so that none of these components is over- or underused.

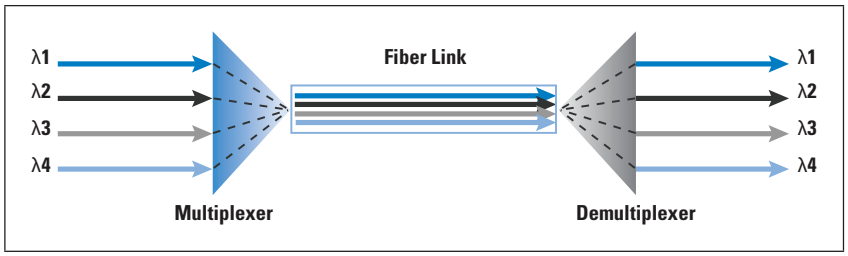
This process of adaptively mapping traffic flows onto the physical topology of a network and allocating resources to these flows—usually referred to as *traffic engineering*—is one of the most difficult tasks facing Internet backbone providers today. *Generalized Multiprotocol Label Switching* (GMPLS) is the most promising technology. GMPLS will play a critical role in future IP pure optical networks by providing the necessary bridges between the IP and optical layers to deliver effective traffic-engineering features and allow for interoperable and scalable parallel growth in the IP and photonic dimension. The GMPLS control plane technology, when fully available in next-generation optical switching devices, will support all the needed traffic-engineering functions and enable a variety of protection and restoration capabilities, while simplifying the integration of new photonic switches and existing label switching routers.

Wavelength Division Multiplexing

Traditional *Electronic Time-Division Multiplexed* (ETDM) networks use an electrical signal form to switch traffic along routes and restore signal strength. These networks do not fully exploit the bandwidth available on optical fibers because only a single frequency (wavelength or *lambda*) of light is used on each fiber to transmit data signals that can be modulated at a maximum bit rate of the order of 40 Gbps. The high bandwidth of optical fibers can be better used through WDM technology by which distinct data signals may share an optical fiber, provided they are transmitted on carriers having different wavelengths[2].

In more detail, the optical transmission spectrum is divided into numerous nonoverlapping wavelengths, with each wavelength supporting a single communication channel. Each channel, which can be viewed as a *light path*, is transmitted at a different wavelength (or frequency). Multiple wavelengths are multiplexed into a single optical fiber and multiple light-path data is transmitted as shown in Figure 1.

Figure 1: WDM Functional Model



Dense WDM (DWDM), an evolution of WDM referring essentially to the closer spacing of channels, is the current favorite multiplexing technology for long-haul communications in modern optical networks. Hence, all the major carriers today devote significant effort to developing and applying DWDM technology in their business.

All-optical networks employing the concept of WDM and wavelength routing are thought to be the transport networks for the future[3]. In such networks, two adjacent nodes are connected by one or multiple fibers, each carrying multiple wavelengths or channels. Each node consists of a dynamically configurable optical switch that supports fiber switching and wavelength switching; that is, the data on a specified input fiber and wavelength can be switched to a specified output fiber on the same wavelength[4]. In order to transfer data between source–destination node pairs, a light path needs to be established by allocating the same wavelength throughout the route of the transmitted data. Benefiting from the development of all-optical amplifiers, light paths can span more than one fiber link and remain entirely optical from end to end. It has been demonstrated that the introduction of wavelength-routing networks not only offers the advantages of higher transmission capacity and routing node throughput, but also satisfies the growing demand for protocol transparency and simplified operation and management[3] [5].

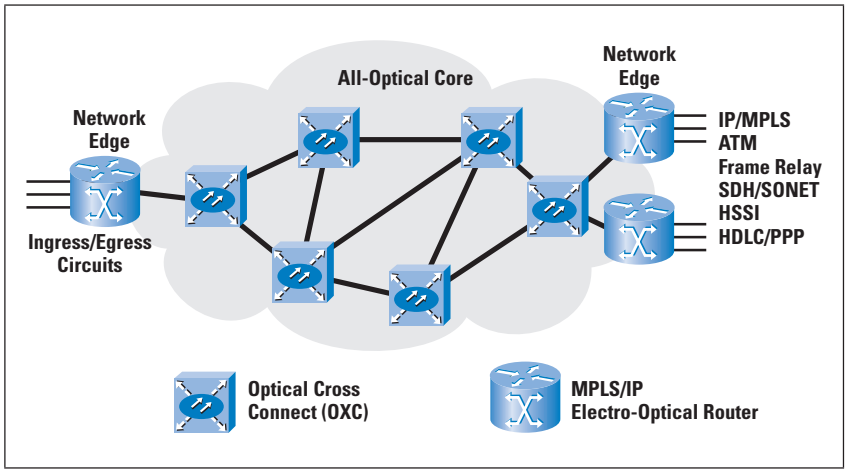
Optical Transport Backbones

The modern Internet transport infrastructure can be physically seen as a very complex mesh of variously interconnected optical or traditional ETDM subnetworks, where each subnetwork consists of several heterogeneous routing and switching devices built by the same or different vendor and operating according to the same control plane protocols and policies. With these very different types of devices, all the forwarding decisions will be based on a combination of packet or cell, timeslot, wavelengths, or physical ports, depending on the position (edge or core) and role (intermediate or termination or gateway node) of the switching devices in the network layout.

In particular, WDM-switched optical subnetworks are typically used as backbone infrastructures to interconnect a large number of different IP as well as other packet networks such as SDH, ATM, and Frame Relay.

New optical devices such as DWDM multiplexers, *Add/Drop Multiplexers* (ADM), and *Optical Cross-Connects* (OXC) are making possible an intelligent all-optical core where packets are routed through the network without leaving the optical domain. The optical network and the surrounding IP networks are independent of each other, and an edge IP router interacts with its ingress switching node only over a well-defined *User-Network Interface* (UNI). Clearly, the optical network is responsible for setting up light paths between the edge IP routers. A light path can be either switched or permanent. Switched light paths are established in real time using proper signaling procedures, and they may last for a short or a long period of time. Permanent light paths are set up administratively by subscription, and they typically last for a very long time. An edge IP router requests a switched light path from its ingress optical switching device using a proper signaling protocol over the UNI. See Figure 2.

Figure 2: The Optical Transport Infrastructure

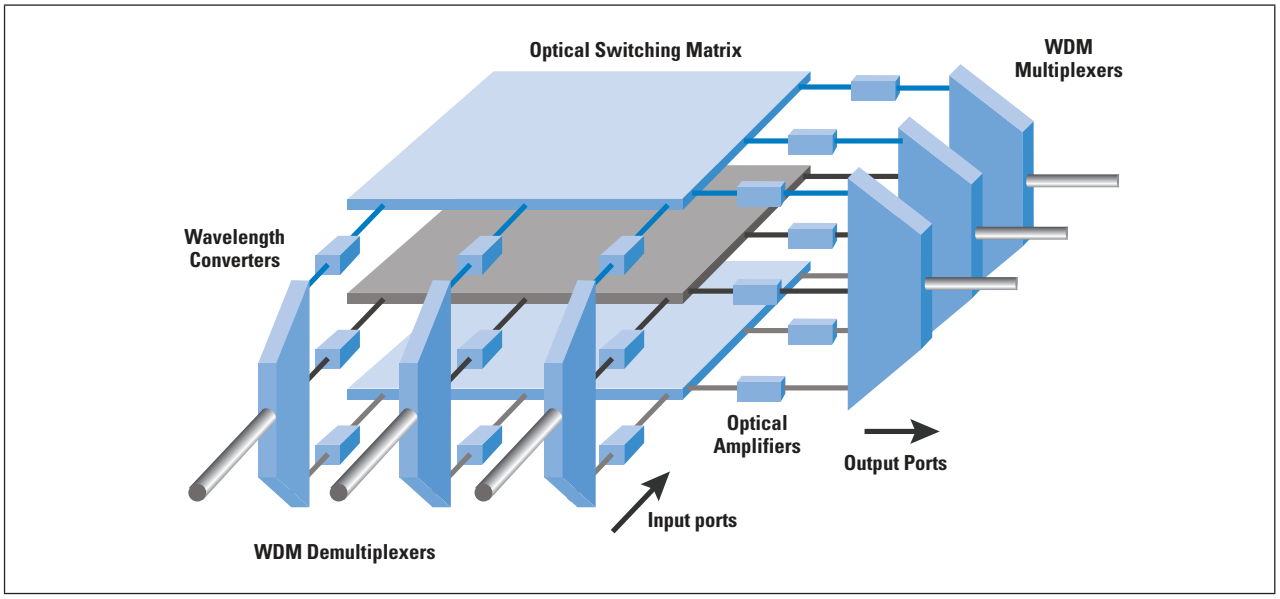


The key concept to guarantee desirable speeds and correct functional behavior in these networks is to maintain the signal in pure optical form, thereby avoiding the prohibitive overhead of conversion to and from electrical form. Such a network would be "optical transparent" in the sense that it would be able to transport client signals with any format and with a wide range of bit rates (at least from about 10 Mbps to more than 10 Gbps). In particular, transparent OXCs, used to selectively switch wavelengths between their input and output ports, are likely to emerge as the preferred option for switching multigigabit or even terabit data streams, because any slow electronic per-packet processing is avoided.

Transparent Optical Switching Nodes

Transparent OXC systems are expected to be the cornerstone of the photonic layer, offering carriers more dynamic and flexible options in building network topologies with enhanced performance and scalability. The development of large and flexible transparent OXCs, now enabled by a new generation of optical components such as optical amplifiers, tunable lasers, and wavelength filters, is still a significant challenge[1]. Their architecture makes use of optical switching fabrics, wavelength multiplexers and demultiplexers, and transparent wavelength converters, which eliminate the need for optoelectronic transponders. A simple and linear architectural model for an optical transparent OXC is shown in Figure 3.

Figure 3: OXC Architectural Model



Here, the WDM demultiplexers separate incoming grouped wavelengths from input ports into individual lambdas. A sufficiently large low-loss connectivity and compact-design, all-optical switching fabric can be realized by using the reflection of light and *Micro-Electromechanical Systems* (MEMS) technology, now widely available on the market. This multilayer switching fabric driven by a micromachined electrical actuator redirects, according to the control plane instructions, each wavelength into appropriate output ports passing through optical amplifiers, typically *Erbium-Doped Fiber Amplifiers* (EDFAs), which boost the signal power in line without the need for any optoelectronic conversion to cope with the effects of light dispersion and attenuation on long distances. The WDM multiplexer then groups the wavelengths from the above multiple layers of cross-connects. Furthermore, the wavelength that arrives into an OXC can be directly passed to the optical switching fabric, to be switched to the appropriate output fiber or previously converted, based on the control plane instructions, to another particular wavelength with the use of a tunable wavelength converter (without being transformed to electricity) if the former output wavelength is not available.

This architecture is transparent; that is, the optical signal does not need to be transformed to electricity at all, implying that this architecture can support any protocol and any data rate. Hence, possible upgrades in the wavelength transport capacity can be accommodated at no extra cost. Furthermore, this architecture decreases the cost because it involves the use of fewer devices than the other architectures. In addition, transparent wavelength conversion eliminates constraints on conversions. In this way the real switching capacity of the OXC is increased, leading to cost reduction. First-generation OXCs require manual configuration. Clearly, an automatic switching capability allowing optical nodes to dynamically modify the network topology based on changing traffic demand is highly desirable.

Automatically Switched Optical Networks

For automatically switched networks, where network nodes may directly initiate or terminate new connections or perform wavelength-level switching in the network, sophisticated and flexible control functions are needed.

The *control plane* supports connection management by clients and also provides protection and restoration services. The control plane of an optical network is also responsible for tracking the network topology and for notifying the state of the network resources. Two families of protocols achieve this task:

- *Routing protocols* are specifically responsible for the reliable advertisement of the optical network topology and the available bandwidth resources within and between network domains. In particular, some areas are relevant within this context: the bundling of links with equivalent or logically bundled characteristics, the definition of the routing areas in an optical domain, the rich specifications of an optical link resource as opposed to a typical advertisement of the up or down interface of IP networks, and the advertisement of the shared risk group (optical fibers flowing in the same cable or duct) to which an optical connection belongs.
- *Signaling protocols* are responsible for provisioning, maintaining, and deleting connections. Optical networks are characterized by connection-oriented paradigms that require a resource reservation protocol. State-of-the-art control plane technologies operating on traditional IP-based networks focus on soft-state protocols that require periodic refresh throughout the participating nodes. In optical networks, where the data plane is separated from the control plane, a possible solution is also to adopt a hard state reservation protocol without periodic refresh to limit the effect caused on the data plane by a failure in the control plane. Furthermore, redundant, generalized label binding is encouraged to reserve protection paths in the mesh network.

Data transport is the most obvious task and the main purpose of an optical network *data plane*. It provides uni- or bidirectional information transport (transmission and switching) between users, detects faults, and monitors signal quality. More specifically, the data plane performs, under the directions of the control plane, data routing to the appropriate ports; channel adds and drops to external, older networks (using the edge interfaces); and label or lambda swapping through an array of WDM demultiplexers, wavelength converters, OXCs, optical amplifiers, and multiplexers.

An important concern that must be addressed in designing an optical network is the cross effect of the failure of a data or control plane. Failures of the data plane are usually addressed by the control plane itself by rerouting the disrupted flows at the appropriate level. The control plane must then advertise quickly the new network state to the neighboring nodes to avoid the presence of stale information in the link databases. A failure of the IP-based control plane usually significantly affects the data plane.

Traffic Engineering in Optical Networks

Traffic engineering should be viewed as assistance to the routing and switching infrastructure that provides additional information used in forwarding traffic along alternate paths across the network, trying to optimize service delivery throughout the network by improving its balanced usage and avoiding congestion caused by uneven traffic distribution. Traffic engineering is required in the modern Internet mainly because the current dynamic routing protocols always use the shortest paths to forward traffic. This practice, obviously, conserves network resources, but it causes some of them to be overused while the other resources remain underused. Furthermore, the routing protocols mentioned earlier never account for specific traffic flow requirements such as bandwidth and *Quality of Service* (QoS) needs. Practitioners in the field often assert that traffic engineering essentially signifies the ability to place traffic where the capacity exists to accommodate it—whereas network engineering denotes the ability to install capacity where the traffic exists.

When a traffic-engineering application implements the right set of features, it should provide precise control over the placement of traffic flows within a routing and switching domain, gaining better network use and realizing a more manageable network. A traffic-engineering solution suitable for transparent optical networks always consists of numerous basic functional components; for example:

- *Traffic monitoring, analysis, and aggregation*—This function collects traffic statistics from the network elements; for example, the OXCs. Then the statistics are analyzed or aggregated to prepare for the traffic engineering and network reconfiguration related to decision making.

- *Bandwidth demand projection*—Bandwidth demand projection estimates the bandwidth requirements in the near future based on past and present measurements and the characteristics of the traffic arrival processes. The bandwidth projections are used for subsequent allocation.
- *Reconfiguration trigger*—This variable consists of a set of policies that decide when a network-level reconfiguration is performed. This decision is based on traffic measurements, bandwidth predictions, and operational areas; for example, to suppress the influence of transitional factors and reserve adequate time for the network to converge.
- *Topology design*—Topology design provides a network topology based on the traffic measurements and predictions. Conceptually this process can be considered as optimizing a graph (that is, OXC connected by light paths at the WDM layer) for specific objectives (for example, maximizing throughput), subject to certain constraints (for example, nodal degree or interface capacity), for a given load matrix (that is, traffic load applied to the network.) This area is, in general, a NP-hard problem. Because reconfiguration is regularly triggered by continually changing traffic patterns, an optimized solution may not be stable. It may be more practical to develop heuristics that place more emphasis on factors such as fast convergence, and less on ongoing traffic, rather than on optimality.
- *Topology migration*—Topology migration consists of algorithms to coordinate the network migration from an old topology to a new one. Because WDM reconfiguration deals with large-capacity channels, changing allocation of channel resources in this coarse granularity significantly affects a large number of end-user flows. Traffic flows have to adapt to the light-path changes at and after each migration step. These effects can potentially spread over the routing pattern of the network, in turn possibly affecting more user flows.

Traditionally, all provisioning and engineering in optical networks has required manual planning and configuration, resulting in setup times of days or even weeks and a marked reluctance among network managers to de-provision resources in case doing so would affect other services. In the last few years, during which control protocols have been deployed to dynamically provide traffic engineering and provisioning or management assistance in optical networks, the control protocols have been proprietary and have greatly suffered from interoperability problems. Consequently, a new standardized control plane framework, supporting evolutionary traffic-engineering features, is needed for automatically switched optical transport networks to foster the expedited development and deployment of a new class of versatile optical switches that specifically address the optical transport needs of the Internet.

The important remaining challenge to be addressed in developing a dynamically reconfigurable optical network is that of controlling the optical resources, especially under distributed control where the network elements exchange information among themselves in a standardized multivendor environment. Performance and reliability requirements make this challenge of paramount importance to photonic networks. Beyond eliminating proprietary "islands of deployment," this common control plane enables independent innovation curves within each product class, and faster service deployment with end-to-end provisioning using a single set of semantics.

The GMPLS Paradigm

GMPLS, the emerging paradigm for the design of control planes for OXCs, aims to address and solve all the challenges mentioned previously, trying to automatically and dynamically configure any kind of network element. It was proposed shortly after *Multiprotocol Label Switching* (MPLS) to extend its packet control plane to encompass time division (for example, for SONET/SDH), wavelength (for optical lambdas) and spatial switching (for example, for incoming port or fiber to outgoing port or fiber). Nongeneralized MPLS overlays a packet-switched IP network to facilitate traffic engineering and allow resources to be reserved and routes predetermined. It provides virtual links or tunnels through the network to connect nodes that lie at the edge of the network. For packets injected into the ingress of an established tunnel, normal IP routing procedures are suspended; instead the packets are label-switched so that they automatically follow the tunnel to its egress.

With the success of MPLS in packet-switched IP networks, optical network providers have accelerated a process to generalize the applicability of MPLS to cover all-optical networks as well. The premise of GMPLS is that the idea of a label can be generalized to be anything that is sufficient to identify a traffic flow. For example, in an optical fiber whose bandwidth is divided into wavelengths, the whole of one wavelength could be allocated to a requested flow. The *Label Switch Routers* (LSRs) at either end of the fiber simply have to agree on which frequency to use. From a control plane perspective, an LSR bases its functions on a table that maintains relations between incoming label or port and outgoing label or port. It should be noted that in the case of the OXC, the table that maintains the relations is not a software entity but it is implemented in a more straightforward way, for example, by appropriately configuring the micro-mirrors of the optical switching fabric.

There are several constraints in reusing the GMPLS control plane. These constraints arise from the fact that LSRs and OXCs use different data technologies. More specifically, LSRs manipulate packets that bear an explicit label, and OXCs manipulate wavelengths that bear the label implicitly; that is, the label value is implicit in the fact that the data is being transported within the agreed frequency band.

Furthermore, because the analogy of a label in the OXC is a wavelength or an optical channel, there are no equivalent concepts of label merging nor label push and pop operations in the optical domain, and label swapping can be realized through wavelength conversion. The transparency and multiprotocol properties of such a control plane approach would allow an OXC to route optical channel trails carrying various types of digital payloads (including IP, ATM, SDH, etc.) coherently and uniformly.

GMPLS Control Plane Functions and Services

GMPLS focuses mainly on the control plane services that perform connection management for the data plane (the actual forwarding logic) for both packet-switched interfaces and non-packet-switched interfaces. The GMPLS control plane essentially facilitates four basic functions:

- *Routing control*—Provides the routing capability, traffic engineering, and topology discovery
- *Resource discovery*—A mechanism to keep track of the system resource availability such as bandwidth, multiplexing capability, and ports
- *Connection management*—Provides end-to-end service provisioning for different services, including connection creation, modification, status query, and deletion
- *Connection restoration*—Implements an additional level of protection to the networks by establishing for each connection one or more presignaled backup paths and enabling very fast switching in case of failure between them.

The fundamental service offered by the GMPLS control plane is dynamic end-to-end connection provisioning. The operators need only to specify the connection parameters and send them to the ingress node. The network control plane then determines the optical paths across the network according to the parameters that the user provides and signals the corresponding nodes to establish the connection. The whole procedure can be done within seconds instead of hours. The other important service is bandwidth on demand, which extends the ease of provisioning even further by allowing the client devices that connect to the optical network to request the connection setup in real time as needed. In order to establish a connection that will be used to transfer data between a source–destination node pair, a light path needs to be established by allocating, in presence of the so-called *continuity constraint,* the same wavelength throughout the route of the transmitted data or selecting the proper wavelength conversion-capable nodes across the path. In fact, if the wavelength continuity constraint is not fully enforced, some wavelength conversion-capable nodes can be placed in the network to reduce the overall blocking probability in case of wavelength resource exhaustion on some nodes. Light paths can span more than one fiber link and remain entirely optical from end to end.

However, according to the mandatory clash constraint, two light paths traversing the same fiber link cannot share the same wavelength on that link. That is, each wavelength on a given fiber is not a sharable resource between light paths.

In general, if there are multiple feasible wavelengths (lambdas) between a source node and a destination node, then a Wavelength Assignment algorithm is required to select a wavelength for a given light path. The wavelength selection can be performed either after an optical route has been determined (in the so-called *decoupled approach*), or in parallel with finding a route. In the latter case, we refer to the coupled approach, in which the entire job is accomplished by a single *Routing and Wavelength Assignment* (RWA) algorithm. When light paths are established and taken down dynamically, routing and wavelength assignment decisions must be made as connection requests arrive to the network. It is possible that, for a given connection request, there may be insufficient network resources to set up a light path, in which case the connection request is blocked. The connection may also be blocked if there is no common wavelength available on all the links along the chosen route. Thus, the objective in the dynamic situation is to choose a route and a wavelength that maximizes the probability of setting up a given connection, while at the same time attempting to minimize the blocking for future connections.

In addition, because the quality of an optical signal degrades as it travels through several optical components and fiber segments, the deployment of "long-distance" light paths may require signal regeneration at strategic locations in a nationwide or global WDM network. As a result, the algorithms performing routing and wavelength assignment, virtual-topology embedding, wavelength conversion, etc. must also be mindful of the locations of the sparse signal regenerators in the network. Such regenerators, which are placed at select locations in the network, "clean up" the optical WDM signal either entirely in the optical domain or through an optoelectronic conversion followed by an electro-optic conversion. Thus the signal from the source travels through the network as far as possible before its quality drops below a certain threshold, thereby requiring it to be regenerated at an intermediate node. The same signal could be regenerated several times in the network before it reaches the destination.

Furthermore, in current multilayer transport networks the bandwidth demanded by traffic typically is orders of magnitude lower than the capacity of lambda links, and the number of available wavelengths per fiber is limited and costly. Hence, it is not worth assigning exclusive end-to-end light paths to these demands, so a better sub-lambda granularity is required. Thus, to increase the throughput of a network with a limited number of lambdas per fiber, *traffic grooming* is required in certain nodes, typically those on the network edge.

The GMPLS control plane ensures traffic-grooming capability on edge nodes by operating on a two-layer model; that is, an underlying pure optical wavelength routed network and an "optoelectronic" time-division multiplexed layer built over it. In the wavelength routed layer, operating exclusively at lambda granularity, when a transparent light path connects two physically adjacent or distant nodes, these nodes will seem adjacent for the upper layer. The upper layer can perform multiplexing of different traffic streams into a single wavelength-based light path through simultaneous time and space switching. Similarly it can demultiplex different traffic streams of a single lambda path. It can also perform remultiplexing: some of the demands demultiplexed can be again multiplexed into some other wavelength paths and handled together along it. This is due to the "generalized" and hence multilayer nature of the GMPLS control plane.

The electronic layer is clearly required for multiplexing packets coming from different ports. This upper electronic layer can be a classical or "next-generation" technology, such as IP/MPLS, but it can also be based on any other networking technology (that is SDH/SONET, ATM, Ethernet, etc.). However, the technology of the upper layer must be unique for all traffic streams that have to be demultiplexed and then multiplexed again, because the network cannot directly multiplex, for example, ATM cells with Ethernet frames.

Another service that gives greatest flexibility to users in handling their own virtual network topologies on the transport core is the *Optical Virtual Private Network* (OVPN), which allows users to have full network resource control of a defined partition of the carrier optical network. Although users have full network resource control of that portion of the network, the OVPN is just a logical network partition and the end users still do not have access and visibility to the carrier's networks. This service can save the carrier's operation resources by allowing end users to perform circuit provisioning and setup procedures.

GMPLS Interfaces

GMPLS encompasses control plane signaling for multiple interface types. The diversity of controlling not only switched packets and cells but also TDM network traffic and optical network components makes GMPLS flexible enough to position itself in the direct migration path from electronic to all-optical network switching. The five main interface types supported by GMPLS follow:

- *Packet Switching Capable* (PSC)—These interfaces recognize packet boundaries and can forward packets based on the IP header or a standard MPLS "shim" header.
- *Layer 2 Switch-Capable* (L2SC)—These interfaces recognize frame and cell headers and can forward data based on the content of the frame or cell header (for example, an ATM LSR that forwards data based on its *Virtual Path Identifier/Virtual Circuit Identifier* (VPI/VCI) value, or Ethernet bridges that forward the data based on the MAC header).

- *Time-Division Multiplexing-Capable* (TDMC)—These interfaces forward the data based on the time slot in a repeating cycle (for example, SDH cross-connect or ADM, interfaces implementing the Digital Wrapper G.709, and *Plesichronous Digital Hierarchy* [PDH] interfaces).
- *Lambda Switch-Capable* (LSC)—These interfaces are for wavelength-based MPLS control of optical devices and wavelength switching devices, such as *optical ADMs* (OADMs) and OXCs, operating at the granularity of the single wavelength or group of wavelengths (waveband). These interfaces forward the optical signal from an incoming optical wavelength to an outgoing optical wavelength. Traffic is forwarded based upon wavelength or waveband.
- *Fiber-Switch-Capable* (FSC)—These interfaces forward the signal from one or more incoming fibers to one or more outgoing fibers for spatial control of interface selection, automated patch panels, and physical fiber switching systems. Traffic is forwarded based on port, fiber, or interface.

These supported interfaces are hierarchal in structure and controlled simultaneously by GMPLS.

Generalized Label

GMPLS defines several new forms of label—the *generalized label* objects. These objects include the generalized label request, the generalized label, the explicit label control, and the protection flag. The generalized label can be used to represent timeslots, wavelengths, wavebands, or space-division multiplexed positions.

With plain MPLS labels embedded in the cell or packet structure for in-band control plane signaling, with the different kinds of interfaces supported by GMPLS it is impossible to embed label-specific information, in terms of fiber port or wavelength switching, into the traffic packet structure. Consequentially, new "virtual" labels have been added to the MPLS label structure. These virtual labels comprise specific indicators that represent wavelengths, fiber bundles, or fiber ports and are distributed to GMPLS nodes through out-of-band GMPLS signaling. GMPLS out-of-band signaling causes a control-channel separation problem.

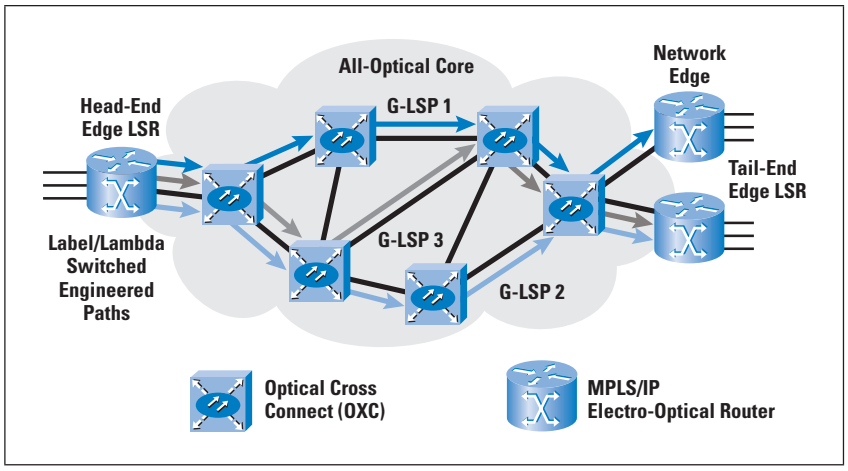
With MPLS, the control information is found in the label, which is directly attached to the data payload. However, when you send the control information out of band, the label is separated from the data that it is attempting to control. GMPLS provides a means for identifying explicit data channels. Having the ability to identify data channels allows the control message to be associated with a particular data flow, whether it is a wavelength, fiber, or fiber bundle.

Generalized Label-Switched Paths

The handling of *label-switched paths* (LSPs) under GMPLS differs from that of MPLS. MPLS does not provide for bidirectional LSPs. Each direction LSP has to be established in turn. Under GMPLS, the LSP can be established bidirectionally. The traffic-engineering requirements for the bidirectional LSP are the same in both directions, and it is established for both directions through only one signaling message, allowing for reductions in latency-related setup time. In the optical environment, OXC translates label assignments into corresponding wavelength assignments and sets up *generalized LSPs* (G-LSPs) using their local control interfaces to the other switching devices. Subsequent to G-LSP setup, no explicit label or lambda lookup or processing operations are performed by the OXC nodes.

GMPLS supports traffic engineering by allowing the node at the network ingress to specify the route that a G-LSP will take by using explicit light-path routing. An explicit route is specified by the ingress as a sequence of hops and wavelengths that must be used to reach the egress, which is different from the hop-by-hop routing that is usually associated with PSC networks.

Figure 4: G-LSPs Ensuring Traffic Engineering



GMPLS also maintains the capability already available with MPLS to nest G-LSPs. Nested G-LSPs make possible the building of a forwarding hierarchy. At the top of this hierarchy are nodes that have FSC interfaces, followed by nodes that have LSC interfaces, followed by nodes that have TDMC interfaces, and followed by nodes with PSC interfaces. Nesting of G-LSPs between interface types increases flexibility in service definition and makes it possible for service providers operating a GMPLS network to deliver both bundled and unbundled services.

Because the deployment of DWDM equipment makes feasible the creation a large number of individual connections between two adjacent nodes, another very useful feature of bundling is the ability to simultaneously handle multiple adjacent links. Link bundling treats the traffic of these links as a single link.

In order for the adjacent links to be bundled, they must be on the same GMPLS segment, they must be of the same type, and they must have the same traffic-engineering requirements. These requirements reduce the amount of link advertisements that need to be maintained throughout the network, thereby increasing the control plane scalability. Just as in MPLS label stacking, GMPLS labels only contain information about a single level of hierarchy. The difference for GMPLS is that this hierarchy can be fiber-, wavelength-, timeslot-, packet- or cell-based.

For instance, if a connection is desired from one PSC interface to another PSC interface, and the traffic traverses physically separate fibers, a unique LSP has to be established for each level in turn. First, the FSC LSP, then the LSC LSP, then the TDMC LSP, and finally the PSC LSP have to be established through GMPLS signaling.

Signaling and Routing Protocols

In order to set up a light path, a signaling protocol is also required to exchange control information among nodes, to distribute labels, and to reserve resources along the path. In our case, the signaling protocol is closely integrated with the routing and wavelength assignment protocols. Suitable GMPLS signaling protocols for the GMPLS control plane include *Resource Reservation Protocol* (RSVP) and *Constraint-Based Label Distribution Protocol* (CR-LDP). Any of the objects that are defined within the GMPLS specification can be carried within the message of either of these signaling protocols that are responsible for all the connection management actions such as setup, modify, or remove the G-LSPs. Clearly, support for provisioning and restoration of end-to-end optical trails within a photonic network consisting of heterogeneous networking elements imposes new requirements for these signaling protocols. Specifically, optical trails require small setup latency (especially for restoration purposes), support for bidirectional trails, rapid failure detection and notification, and fast intelligent trail restoration.

Both RSVP and CR-LDP can be used to reserve a single wavelength for a light path if the wavelength is known in advance. These protocols can also be modified to incorporate wavelength selection functions into the reservation process[7]. In RSVP, signaling takes place between the source and destination nodes. The signaling messages may contain information such as QoS requirements for the carried traffic and label requests for assigning labels at intermediate nodes that reserve the appropriate resources for the path. CR-LDP uses TCP sessions between nodes in order to provide a hop-by-hop reliable distribution of control messages, indicating the route and the required traffic parameters for the route. Each intermediate node reserves the required resources, allocates a label, and sets up its forwarding table before backward signaling to the previous node.

To correctly perform resource reservation, allocation, and topology discovery on the available optical link resources, each node needs to maintain a representation of the state of each link in the network. The link state includes the total number of active channels, the number of allocated channels, and the number of channels reserved for light-path restoration. Additional parameters can be associated with allocated channels; for example, some light paths can be preemptable or have associated hold priorities. When the local inventory is constructed, the node engages in a routing protocol to distribute and maintain the topology and resource information. Standard IP routing protocols, such as *Open Shortest Path First* (OSPF) or *Intermediate System-to-Intermediate System* (IS-IS) with GMPLS Traffic Engineering extensions, can be used to reliably propagate the information.

The extensions to OSPF and IS-IS add additional information about links and nodes into the link-state database. Such information includes the type of LSPs that can be established across a given link (for example, packet forwarding, SONET/SDH trails, wavelengths, or fibers), as well as the current unused bandwidth, the maximum size of G-LSP that can be established, and the administrative groups supported. This information allows the node computing the explicit route for an LSP to do so more intelligently. Furthermore, any switching node cooperating in the GMPLS control plane will maintain a per-interface or per-fiber *Wavelength Forwarding Information Base* (WFIB) because lambdas and channels (labels) are specific to a particular interface or fiber, and the same lambda or channel (label) could be used concurrently on multiple interfaces or fibers.

Link Management Protocol

GMPLS also uses the *Link Management Protocol* (LMP) to communicate proper cross-connect information between the network elements. LMP runs between adjacent systems for link provisioning and fault isolation. It can be used for any type of network element, particularly in natively photonic switches. LMP automatically generates and maintains associations between links and labels for use in label swapping[6]. Automating the labeling process simplifies management and avoids the errors associated with manual label assignment. LMP provides control-channel management, link-connectivity verification, link-property correlation, and fault isolation. Control-channel management establishes and maintains connectivity between adjacent nodes using a keepalive protocol. Link verification verifies the physical connectivity between nodes, thereby detecting loss of connections and misrouting of cable connections. Fault isolation pinpoints failures in both electronic and optical links without regard to the data format traversing the link.

In order for these link bundles to be handled accordingly, GMPLS needed a method to manage the links between adjacent nodes. LMP was developed to address several link-specific problems that surfaced when generalizing the MPLS protocol across different interface types. The main responsibilities of the LMP follow:

- *Control-Channel Management*—Establishment of a control channel is critical to GMPLS signaling. The maintenance of the control channel between adjacent nodes must be able to exchange information related to LSP establishment.
- *Link-Property Correlation*—When link bundling occurs, GMPLS requires a way to verify that all traffic-engineering requirements are similar between links of adjacent nodes. Link-property correlation performs the verification and the aggregation of such links.
- *Link-Connectivity Verification*—This feature is used by GMPLS to verify the connectivity between data links when the control channel is separate from each data link.
- *Fault Management*—Fault management helps the network isolate faults down to the individual link.

Although LMP assumes the messages are IP encoded, it does not dictate the actual transport mechanism used for the control channel. However, the control channel must terminate on the same two nodes that the bearer channels span. Therefore, this protocol can be implemented on any OXC, regardless of the internal switching fabric. A requirement for LMP is that each link has an associated bidirectional control channel and that free bearer channels must be opaque (that is, able to be terminated); however, when a bearer channel is allocated, it may become transparent. Note that this requirement is trivial for optical cross-connects with electronic switching planes, but is an added restriction for photonic switches.

Conclusion

Innovations in the field of optical components will take advantage of the introduction of all-optical networking in all areas of information transport and will offer system designers the opportunity to create new solutions that will allow smooth evolution of all telecommunication networks. A new class of versatile IP-addressable optical switching devices is emerging, operating according to a common GMPLS-based control plane to support full-featured traffic engineering in modern optical transparent infrastructures.

The main advantage of this approach is that it is based on already existing and widely deployed protocols while simplifying network management and engineering tasks that can be performed in a unified way in both the data and the optical domains. Furthermore, it offers a function framework that can accommodate future expectations concerning the way networks will work and the way services will be provided to clients. Thus we envision a horizontal network, harmonized by a common GMPLS-based control plane, where all network elements work as peers to dynamically establish optical paths through the network.

This new photonic internetwork will make it possible to provision high bandwidth in tenths of seconds, and enable new revenue-generating services and dramatic cost savings for service providers.

In the same way that digital communication technologies changed the twentieth century into the "electronic century," the optical technologies discussed in this article will make the next century "the photonic century." All winning strategies must rely on such GMPLS-based photonic infrastructures—an environment in which innovations work at the speed of light.

For Further Reading

[1] B. E. A. Saleh and M. C. Teich, *Fundamentals of Photonics,* John Wiley & Sons Inc., 1991.

[2] P. Raghavan and E. Upfal, "Efficient Routing in All-Optical Networks," *Proceedings of ACM STOC'94,* 1994.

[3] B. Mukherjee, *Optical Communication Networks,* McGraw-Hill, 1997.

[4] A. Mokhtar and M. Azizoglu, "Adaptive Wavelength Routing in All-Optical Networks," *IEEE/ACM Transactions on Networking,* vol. 6, pp. 197–206, April 1998.

[5] E. Karasan and S. Banerjee, "Performance of WDM Transport Networks," *IEEE Journal on Selectected Areas in Communications,* vol. 16, pp. 1081–1096, September 1998.

[6] A. Banerjee, J. Drake, J. Lang, B. Turner, K. Kompella, and Y. Rekhter, "Generalized Multiprotocol Label Switching: An Overview of Routing and Management Enhancements," *IEEE Communications Magazine,* January 2001.

[7] A. Banerjee, J. Drake, J. Lang, B. Turner, D. O. Awduche, L. Berger, K. Kompella, and Y. Rekhter, "Generalized Multiprotocol Label Switching: An Overview of Signalling Enhancements and Recovery Techniques," *IEEE Communications Magazine,* July 2001.

FRANCESCO PALMIERI holds two Computer Science degrees from Salerno University, Italy. Since 1997, he has led the network management and operation centre of the Federico II University, in Napoli, Italy. He has been closely involved with the development of the Internet in Italy in the last few years, particularly within the academic and research sector, and is actually a member of the Technical Scientific Committee and of the Computer Emergency Response Team of the Italian NREN GARR. He worked for several international companies on a variety of networking-related projects concerned with nationwide communication systems, network management, transport protocols, and IP networking. He is an active researcher in the fields of high-performance, evolutionary networking, and network security. He regularly publishes in leading technical journals and conferences and gives invited talks and keynote speeches. E-Mail: **Francesco.Palmieri@unina.it**

The Changing Foundation of the Internet: Confronting IPv4 Address Exhaustion

by Geoff Huston, APNIC

Throughout its relatively brief history, the Internet has continually challenged our preconceptions about networking and communications architectures. For example, the concepts that the network itself has no role in management of its own resources, and that resource allocation is the result of interaction between competing end-to-end data flows, were certainly novel innovations, and for many they have been very confrontational. The approach of designing a network that is unaware of services and service provisioning and is not attuned to any particular service whatsoever—leaving the role of service support to end-to-end overlays—was again a radical concept in network design. The Internet has never represented the conservative option for this industry, and has managed to define a path that continues to present significant challenges.

From such a perspective it should not be surprising that the next phase of the Internet story—that of the transition of the underlying version of the IP protocol from IPv4 to IPv6—refuses to follow the intended script. Where we are now, in late 2008, with IPv4 unallocated address pool exhaustion looming within the next 18 to 36 months, and IPv6 still largely not deployed in the public Internet, is a situation that was entirely uncontemplated and, even in hindsight, entirely surprising.

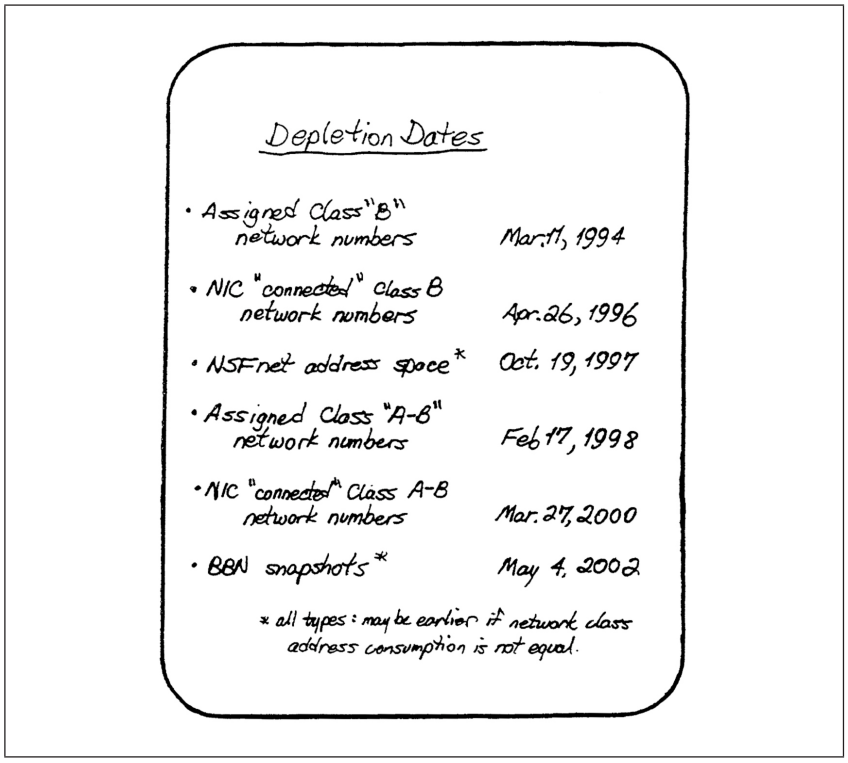
The topic examined here is *why* this situation has arisen, and in examining this question we analyze the options available to the Internet to resolve the problem of IPv4 address exhaustion. We examine the timing of the IPv4 address exhaustion and the nature of the intended transition to IPv6. We consider the shortfalls in the implementation of this transition, and identify their underlying causes. And finally, we consider the options available at this stage and identify some likely consequences of such options.

When?

This question was first asked on the TCP/IP list in November 1988, and the responses included foreshadowing a new version of IP with longer addresses and undertaking an exercise to reclaim unused addresses[1]. The exercise of measuring the rate of consumption of IPv4 addresses has been undertaken many times in the past two decades, with estimates of exhaustion ranging from the late 1990s to beyond 2030. One of the earliest exercises in predicting IPv4 address exhaustion was undertaken by Frank Solensky and presented at IETF 18 in August 1990. His findings are reproduced in Figure 1.

At that time the concern was primarily the rate of consumption of Class B network addresses (or of /16 prefixes from the address block `128.0.0.0/2`, to use current terminology). Only 16,384 such Class B network addresses were within the class-based IPv4 address plan, and the rate of consumption was such that the Class B networks would be fully consumed within 4 years, or by 1994. The prediction was strongly influenced by a significant number of international research networks connecting to the Internet in the late 1980s, with the rapid influx of new connections to the Internet creating a surge in demand for Class B networks.

Figure 1: Report on IPv4 Address Depletion[2]



Successive predictions were made in the context of the *Internet Engineering Task Force* (IETF) in the *Address Lifetime Expectancy* (ALE) Working Group, where the predictive model was refined from an exponential growth model to a logistical saturation function, attempting to predict the level at which all address demands would be met.

The predictive technique described here is broadly similar, using a statistical fit of historical data concerning address consumption into a mathematical model, and then using this model to predict future address consumption rates and thereby predict the exhaustion date of the address pool.

The predictive technique models the IP address distribution framework. Within this framework the pool of unallocated /8 address blocks is distributed by the *Internet Assigned Numbers Authority* (IANA) to the five *Regional Internet Registries* (RIRs). (A "/8 address block" refers to a block of addresses where the first 8 bits of the address values are constant. In IPv4 a /8 address block corresponds to 16,777,216 individual addresses.) Within the framework of the prevailing address distribution policies, each RIR can request a further address allocation from IANA when the remaining RIR-managed unallocated address pool falls below a level required to meet the next 9 months of allocation activity. The amount allocated is the number of /8 address blocks required to augment the RIR's local address pool to meet the anticipated needs of the regional registry for the next 18 months. However, in practice, the RIRs currently request a maximum of 2 /8 address blocks in any single transaction, and do so when the RIR-managed address pool falls below a threshold of the equivalent of 2 /8 address blocks.

As of August 2008 some 39 /8 address blocks are left in IANA's unallocated address pool. A predictive exercise has been undertaken using a statistical modeling of historical address consumption rates, using data gathered from the RIRs' records of address allocations and the time series of the total span of address space announced in the Internet interdomain default-free routing table as basic inputs to the model. The predictive technique is based on a least-squares best fit of a linear function applied to the first-order differential of a smoothed copy of the address consumption data series, as applied to the most recent 1,000 days' data.

The linear function, which is a best fit to the first-order differential of the data series, is integrated to provide a quadratic time-series function to match the original data series. The projection model is further modified by analyzing the day-of-year variations from the smoothed data model, averaged across the past 3 years, and applying this daily variation to the projection data to account for the level of seasonal variations in the total address consumption rate that has been observed in the historical data. The anticipated rate of consumption of addresses from this central pool of unallocated IPv4 addresses is expected to be about 15 /8s in 2009, and slightly more in 2010.

RIR behaviors are modeled using the current RIR operational practices and associated address policies, which are used to predict the times when each RIR will be allocated a further 2 /8s from IANA. This RIR consumption model, in turn, allows the IANA address pool to be modeled.

This anticipated rate of increasing address consumption will see the remaining unallocated addresses that are held by IANA reach the point of exhaustion in February 2011. The most active RIRs are anticipated to exhaust their locally managed unallocated address pools in the months following the time of IANA exhaustion.

The assumptions behind this form of prediction follow:

- The current policy framework relating to the distribution of addresses will continue to apply without any further alteration through to complete exhaustion of the unallocated address pool.
- The demand curves will remain consistent, meaning that there will be no forms of disruption to demand, such as a panic rush on the remaining addresses or some introduced externality that affects total address demand.
- The level of return of addresses to the unallocated address pool will not vary significantly from existing levels of address return.

Although the statistical model is based on a complete data set of address allocations and a detailed hourly snapshot of the address span advertised in the Internet routing table, a considerable level of uncertainty is still associated with this prediction.

First, the behavior of the *Internet Service Provider* (ISP) industry and the other entities that are the direct recipients of RIR address allocations and assignments are not ignorant of the impending exhaustion condition, and there is some level of expectation of some form of last-minute rush or panic on the part of such address applicants when exhaustion of this address pool is imminent. The predictive model described here does not include such a last-minute acceleration of demand.

The second factor is the skewed distribution of addresses in this model. From 1 January 2007 until 20 July 2008, 10,402 allocation or assignments transactions were recorded in the RIRs' daily statistics files. These transactions accounted for a total of 324,022,704 individual IPv4 addresses, or the equivalent of 19.3 /8s. Precisely one-half of this address space was allocated or assigned in just 107 such transactions.

In other words, some 1 percent of the recipients of address space in the past 18 months have received some 50 percent of all the allocated address space. The reason why this distribution is relevant here is that this predictive exercise assumes that although individual actions are hard to predict with any certainty, the aggregate outcome of many individuals' actions assumes a much greater level of predictability.

This observation about aggregate behavior does not apply in this situation, however, and the predictive exercise is very sensitive to the individual actions of a very small number of recipients of address space because of this skewed distribution of allocations. Any change in the motivations of these larger-sized actors that results in an acceleration of demand for IPv4 will significantly affect the predictions of the longevity of the remaining unallocated IPv4 address pool.

The third factor is that this model assumes that the policy framework remains unaltered, and that all unallocated addresses are allocated or assigned under the current policy framework, rather than under a policy regime that is substantially different from today's framework. The related assumption here is that the cost of obtaining and holding addresses remains unchanged, and that the perceptions of future scarcity of addresses do not affect the policy framework of address distribution of the remaining unallocated IPv4 addresses.

Given this potential for variation within this set of assumptions, a more accurate summary of the current expectations of address consumption would be that the exhaustion of the IANA unallocated IPv4 address pool will occur sometime between July 2009 and July 2011, and that the first RIR will exhaust all its usable address space within 3 to 12 months from that date, or between October 2009 and July 2012.[3]

What Next?

Apart from the exact date of exhaustion that is predicted by this modeling exercise, none of the information relating to exhaustion of the unallocated IPv4 address pool should be viewed as particularly novel information. The IETF *Routing and Addressing* (ROAD) study of 1991 recognized that the IPv4 address space was always going to be completely consumed at some point in the future of the Internet[4].

Such predictions of the potential for exhaustion of the IPv4 address space were the primary motivation for the adoption of *Classless Inter-Domain Routing* (CIDR) in the *Border Gateway Protocol* (BGP), and the corresponding revision of the address allocation policies to craft a more exact match between planned network size and the allocated address block. These predictions also motivated the protracted design exercise of what was to become the IPv6 protocol across the 1990s within the IETF. The prospect of address scarcity engendered a conservative attitude to address management that, in turn, was a contributory factor in accelerating the widespread use of *Network Address Translation* (NAT)[5] in the Internet during the past decade. By any reasonable metric this industry has had ample time to study this problem, ample time to devise various strategies, and ample time to make plans and execute them.

And this reality has been true for the adoption of classless address allocations, the adoption of CIDR in BGP, and the extremely widespread use of NAT. But all of these measures were short-term, whereas the longer-term measure, that of the transition to IPv6, was what was intended to come after IPv4. But IPv6 has not been the subject of widespread adoption so far, while the time of anticipated exhaustion of IPv4 has been drawing closer. Given almost two decades of advance warning of IPv4 address exhaustion, and a decade since the first stable implementations of IPv6 were released, we could reasonably expect that this industry—and each actor within this industry—is aware of the problem and the need for a stable and scalable long-term solution as represented by IPv6. We could reasonably anticipate that the industry has already planned the actions it will take with respect to IPv6 transition, and is aware of the triggers that will invoke such actions, and approximately when they will occur.

However, such an expectation appears to be ill-founded when considering the broad extent of the actors in this industry, and there is little in the way of a common commitment as to what will happen after IPv4 address exhaustion, nor even any coherent view of plans that industry actors are making in this area.

This lack of planning makes the exercise of predicting the actions within this industry following address exhaustion somewhat challenging, so instead of immediately describing future scenarios, it may be useful to first describe the original plan for the response of the Internet to IPv4 address exhaustion.

What Was Intended?

The original plan, devised in the early 1990s by the IETF to address the IPv4 address shortfall, was the adoption of CIDR as a short-term measure to slow down the consumption of IPv4 addresses by reducing the inefficiency of the address plan, and the longer-term plan of the specification of a new version of the Internet Protocol that would allow for adoption well before the IPv4 address pool was exhausted.

The industry also adopted the use of NAT as an additional measure to increase the efficiency of address use, although the IETF did not strongly support this protocol. For many years the IETF did not undertake the standardization of NAT behaviors, presumably because NAT was not consistent with the IETF's advocacy of end-to-end coherence of the Internet at the IP level of the protocol stack.

Over the 1990s the IETF undertook the exercise of the specification of a successor IP protocol to Version 4, and the IETF's view of the longer-term response was refined to be advocacy of the adoption of the IPv6 protocol and the use of this protocol as the replacement for IPv4 across all parts of the network.

In terms of what has happened in the past 15 years, the adoption of CIDR was extremely effective, and most parts of the network were transitioned to use CIDR within 2 years, with the transition declared to be complete by the IETF in June 1996. And, as noted already, NAT has been adopted across many, if not most, parts of the network. The most common point of deployment of NAT has not been at an internal point of demarcation between provider networks, but at the administrative boundary between the local customer network and the ISP, so that the common configuration of *Customer Premises Equipment* (CPE) includes NAT functions. Customers effectively own and operate NAT devices as a commonplace aspect of today's deployed Internet.

CIDR and NAT have been around for more than a decade now, and the address consumption rates have been held at very conservative levels in that period, particularly so when considering that the bulk of the population of the Internet was added well after the advent of CIDR and NAT.

The longer-term measure—the transition to IPv6—has not proved to be as effective in terms of adoption in the Internet.

There was never going to be a "flag-day" transition where, in a single day, simultaneously across all parts of every network the IP protocol changed to using IPv6 instead of IPv4. The Internet is too decentralized, too large, too disparate, and too critical for such actions to be orchestrated, let alone completed with any chance of success. A flag day, or any such form of coordinated switchover, was never a realistic option for the Internet.

If there was no possibility of a single, coordinated switchover to IPv6, the problem is that there was never going to be an effective piecemeal switchover either. In other words, there was never going to be a switchover where host by host, and network by network, IPv6 is substituted for IPv4 on a piecemeal and essentially uncoordinated basis. The problem here is that IPv6 is not "backward-compatible" with IPv4. When a host uses IPv6 exclusively, then that host has no direct connectivity to any part of the IPv4 network. If an IPv6-only host is connected to an IPv4-only network, then the host is effectively isolated. This situation does not bode well for a piecemeal switchover, where individual components of the network are switched over from IPv4 to IPv6 on a piecemeal basis. Each host that switches over to IPv6 essentially disconnects itself form the IPv4 Internet at that point.

Given this inability to support backward compatibility, what was planned for the transition to IPv6 was a "dual-stack" transition. Rather than switching over from IPv4 to IPv6 in one operation on both hosts and networks, a two-step process has been proposed: first switching from IPv4 only to a "dual-stack" mode of operation that supports both IPv4 and IPv6 simultaneously, and second—and at a much later date—switching from dual-stack IPv4 and IPv6 to IPv6 only.

During the transition more and more hosts are configured with dual stack. The idea is that dual-stack hosts prefer to use IPv6 to communicate with other dual-stack hosts, and revert to use IPv4 only when an IPv6-based end-to-end conversation is not possible. As more and more of the Internet converts to dual stack, it is anticipated that use of IPv4 will decline, until support for IPv4 is no longer necessary. In this dual-stack transition scenario, no single flag day is required and the dual-stack deployment can be undertaken in a piecemeal fashion. There is no requirement to coordinate hosts with networks, and as dual-stack capability is supported in networks the attached dual-stack hosts can use IPv6. This scenario still makes some optimistic assumptions, particularly relating to the achievement of universal deployment of dual stack, at which point no IPv4 functions are used, and support for IPv4 can be terminated. Knowing when this point is reached is unclear, of course, but in principle there is no particular timetable for the duration of the dual-stack phase of operation.

There are always variations, and in this case it is not necessarily that each host must operate in dual-stack mode for such a transition. A variant of the NAT approach can perform a rudimentary form of protocol translation, where a *Protocol-Translating NAT* (or NAT-PT[6]) essentially transforms an incoming IPv4 packet to an outgoing IPv6 packet, and conversely, using algorithmic binding patterns to map between IPv4 and IPv6 addresses. Although this process relieves the IPv6-only host of some additional complexity of operation at the expense of some added complexity in *Domain Name System* (DNS) transformations and service fragility, the essential property still remains that in order to speak to an IPv4-only remote host, the combination of the local IPv6 host and the NAT-PT have to generate an equivalent IPv4 packet. In this case the complexity of the dual stack is now replaced by complexity in a shared state across the IPv6 host and the NAT-PT unit. Of course this solution does not necessarily operate correctly in the context of all potential application interactions, and concerns with the integrity of operation of NAT-PT devices are significant, a factor that motivated the IETF to deprecate the existing NAT-PT specification[7]. On the other hand, the lack of any practical alternatives has led the IETF to subsequently reopen this work, and once again look at specifying the standard behavior of such devices[8].

The detailed progress of a dual-stack transition is somewhat uncertain, because it involves the individual judgment of many actors as to when it may be appropriate to discontinue all support for IPv4 and rely solely on IPv6 for all connectivity requirements. However, one factor is constant in this envisaged transition scenario, and whether it is dual stack in hosts or dual stack through NAT-PT, or various combinations thereof, the requirement that there are sufficient IPv4 addresses to span the addressing needs of the entire Internet across the complete duration of the dual-stack transition process is consistent.

Under this dual-stack regime every new host on the Internet is envisaged to need access to both IPv6 and IPv4 addresses in order to converse with any other host using IPv6 or IPv4. Of course this approach works as long as there is a continuing supply of IPv4 addresses, implying that the envisioned timing of the transition was meant to have been completed by the time that IPv4 address exhaustion happens.

If this transition were to commence in earnest at the present time, in late 2008, and take an optimistic 5 years to complete, then at the current address consumption rate we will require a further 90 to 100 /8 address blocks to span this 5-year period. A more conservative estimate of a 10-year transition will require a further 200 to 250 /8 address blocks, or the entire IPv4 address space again, assuming that we will use IPv4 addresses in the future in precisely the same manner as we have used them in the past and with precisely the same level of usage efficiency as we have managed to date.

Clearly, waiting for the time of IPv4 unallocated address pool exhaustion to act as the signal to industry to commence the deployment of IPv6 in a dual-stack transition framework is a totally flawed implementation of the original dual-stack transition plan.

Either the entire process of dual-stack transition will need to be undertaken across a far faster time span than has been envisaged, or the manner of use of IPv4 addresses, and, in particular their usage efficiency in the context of dual-stack transition support, will need to differ markedly from the current manner of address use. Numerous forms of response may be required, posing some challenging questions because there is no agreed precise picture of what markedly different and significantly more efficient form of address use is required here. To paraphrase the situation, it is clear that we need to do "something" differently, and do so as a matter of some urgency, but we have no clear agreement on what that something is that we should be doing differently. This situation obviously is not an optimal one.

What was intended as a transition mechanism for IPv6 is still the only feasible approach that we are aware of, but the forthcoming exhaustion of the unallocated IPv4 address pool now calls for novel forms of use of IPv4 addresses within this transitional framework, and these novel forms may well entail the deployment of various forms of address translation technologies that we have not yet defined, let alone standardized. The transition may also call for scaling capabilities from the interdomain routing system that also head into unknown areas of technology and deployment feasibility.

Why?

At this point it may be useful to consider how and why this situation has arisen.

If the industry needed an abundant supply of IPv4 addresses to underpin the entire duration of the dual-stack transition to IPv6, then why didn't the industry follow the lead of the IETF and commence this transition while there was still an abundant supply of IPv4 addresses on hand? If network operators, service providers, equipment vendors, component suppliers, application developers, and every other part of the Internet supply chain were aware of the need to commence a transition to IPv6 well before effective exhaustion of the remaining pool of IPv4 addresses, then why didn't the industry make a move earlier? Why was the only clear signal for a change in Internet operation to commence a dual-stack transition to IPv6 one that has been activated too late to be useful for the industry to act on efficiently?

One possible reason may lie in a perception of the technical immaturity of IPv6 as compared to IPv4. It is certainly the case that many network operators in the Internet are highly risk-adverse and tend to operate their networks in a mainstream path of technologies rather than constantly using leading-edge advance releases of hardware and software solutions. Does IPv6 represent some form of unacceptable technical risk of failure that has prevented its adoption? This reasoning does not appear to be valid in terms of either observed testing or observation of perceptions about the technical capability of IPv6. The IPv6 protocol is functionally complete and internally consistent, and it can be used in almost all contexts where IPv4 is used today. IPv6 works as a platform for all forms of transport protocols, and is fully functional as an internetwork layer protocol that is functionally equivalent to IPv4. IPv6 NAT exists, *Dynamic Host Configuration Protocol Version 6* (DHCPv6) provides dynamic host configuration for IPv6 notes, and the DNS can be completely equipped with IPv6 resource records and operate using IPv6 transport for queries and responses.

Perhaps the only notable difference between the two protocols is the ability to perform host scans in IPv6, where probe packets are sent to successive addresses. In IPv6 the address density is extremely low because the low-order 64-bit interface address of each host is more or less unique, and within a single network the various interface addresses are not clustered sequentially in the number space. The only known use of address probing to date has been in various forms of hostile attack tools, so the lack of such a capability in IPv6 is generally seen as a feature rather than an impediment. IPv6 deployment has been undertaken in a small scale for many years, and although the size of the deployed IPv6 base remains small, the level of experience gained with the technology functions has been significant. It is possible to draw the conclusion that IPv6 is technically capable and this capability has been broadly tested in almost every scenario except that of universal use across the Internet.

It also does not appear that the reason was a lack of information or awareness of IPv6. The efforts to promote IPv6 adoption have been under way in earnest for almost a decade now. All regions and many of the larger economies have instigated programs to promote the adoption of IPv6 and have provided information to local industry actors of the need to commence a dual-stack transition to IPv6 as soon as possible. In many cases these promotional programs have enjoyed broad support from both public and industry funding sources. The coverage of these promotional efforts has been widespread in industry press reports. Indeed, perhaps the only criticism of this effort is possibly too much promotion, with a possible result that the effectiveness of the message has been diluted through constant repetition.

A more likely area to examine in terms of possible reasons why industry has not engaged in dual-stack transition deployment is that of the business landscape of the Internet. The Internet can be viewed as a product of the wave of progressive deregulation in the telecommunications sector in the 1980s and early 1990s. New players in the deregulated industry searching for a competitive edge to unseat the dominant position of the traditional incumbents found the Internet as their competitive lever. The result was perhaps unexpected, because it was not one that replaced one vertically integrated operator with a collection of similarly structured operators whose primary means of competition was in terms of price efficiency across an otherwise undifferentiated service market, as we saw in the mobile telephony industry. In the case of the Internet, the result was not one that attempted to impose convergence on this industry, but one that stressed divergence at all levels, accompanied by branching role specialization at every level in the protocol stack and at every point in the supply chain process. In the framework of the Internet, consumers are exposed to all parts of the supply process, and do not rely on an integrator to package and supply a single, all-embracing solution. Consumers make independent purchases of their platform technology, their software, their applications, their access provider, and their means of advertising their own capabilities to provide goods and services to others, all as independent decisions, all as a result of this direct exposure to the consumer of every element in the supply chain.

What we have today is an industry structure that is highly diverse, broadly distributed, strongly competitive, and intensely focused on meeting specific customer needs in a price-sensitive market, operating on a quarter-by-quarter basis. Bundling and vertical integration of services has been placed under intense competitive pressure, and each part of the network has been exposed to specialized competition in its right. For consumers this situation has generated significant benefits. For the same benchmark price of around US$15 to US$30 per month, or its effective equivalent in purchasing power of a local currency, today's Internet user enjoys multimegabit-per-second access to a richly populated world of goods and services.

The price of this industry restructure has been a certain loss of breadth and depth of the supply side of the market. If consumers do not value a service, or even a particular element of a service, then there is no benefit in incurring marginal additional cost in providing the service. In other words, if the need for a service is not immediate, then it is not provided. For all service providers right through the supply side the focus is on current customer needs, and this focus on current needs, as distinct from continued support of old products or anticipatory support of possible new products, excludes all other considerations.

Why is this change in the form of communications industry operation an important factor in the adoption of IPv6? The relevant question in this context is that of placing IPv6 deployment and dual-stack transition into a viable business model. IPv6 was never intended to be a technology visible to the end user. It offers no additional functions to the end user, nor any direct cost savings to the customer or the supplier. Current customers of ISPs do not need IPv6 today, and neither current nor future customers are aware that they may need it tomorrow. For end users of Internet services, e-mail is e-mail and Web-based delivery of services is just the Web. Nothing will change that perspective in an IPv6 world, so in that respect customers do not have a particular requirement for IPv6, as opposed to a generic requirement for IP access, and will not value such an IPv6-based access service today in addition to an existing IPv4 service. For an existing customer IPv6 and dual stack simply offer no visible value. So if the existing customer base places no value on the deployment of IPv6 and dual stack, then the industry has little incentive to commit to the expenditure to provide it.

Any IPv6 deployment across an existing network is essentially an unfunded expenditure exercise that erodes the revenue margins of the existing IPv4-based product. And as long as sufficient IPv4 address space remains to cover the immediate future needs, looking at this situation on the basis of a quarter-by-quarter business cycle, then the decision to commit to additional expenditure and lower product margins to meet the needs of future customers using IPv6 and dual-stack deployments is a decision that can comfortably be deferred for another quarter. This business structure of today's Internet appears to represent the major reason why the industry has been incapable of making moves on dual-stack transition within a reasonable timeframe as it relates to the timeframe of IPv4 address pool exhaustion.

What of the strident calls for IPv6 deployment? Surely there is substance to the arguments to deploy IPv6 as a contingency plan for the established service providers in the face of impending IPv4 address exhaustion, and if that is the case, why have service providers discounted the value of such contingency motivations? The problem to date is that IPv4 address exhaustion is now not a novel message, and, so far, NAT usage has neutralized the urgency of the message.

The NAT protocol is well-understood, it appears to work reliably, applications work with it, and it has influenced the application environment to such an extent that now no popular application can be fielded unless is can operate across this protocol. For conventional client-server applications, NAT represents no particular problem. For peer-to-peer–based applications, the rendezvous problem with NAT has been addressed through application gateways and rendezvous servers. Even the variability of NAT behavior is not a service provider liability, and it is left to applications to load additional functions to detect specific NAT behavior and make appropriate adjustments to the behavior of the application.

The conventional industry understanding to date is that NAT can work acceptably well within the application and service environment. In addition, NAT usage for an ISP represents an externalized cost, because it is essentially funded and operated by the customer and not the ISP. The service provider's perspective is that considering that this protocol has been so effective in externalizing the costs of IPv4 address scarcity from the ISP for the past 5 years, surely it will continue to be effective for the next quarter. To date the costs of IPv4 address scarcity have been passed to the customer in the form of NAT-equipped CPE devices and to the application in the form of higher complexity in certain forms of application rendezvous. ISPs have not had to absorb these costs into their own costs of operation. From this perspective, IPv6 does not offer any marginal benefits to ISPs. For an ISP today, NATs are purchased and operated by customers as part of their CPE equipment. To say that IPv6 will eliminate NATs and reduce the complexities and vulnerabilities in the NAT service model is not directly relevant to the ISP.

The more general observation is that, for the service provider industry currently, IPv6 has all the negative properties of revenue margin erosion with no immediate positive benefits. This observation lies at the heart of why the service provider industry has been so resistant to the call for widespread deployment of IPv6 services to date.

It appears that the current situation is not the outcome of a lack of information about IPv6, nor a lack of information about the forthcoming exhaustion of the IPv4 unallocated address pool. Nor is it the outcome of concerns over technical shortfalls or uncertainties in IPv6, because there is no evidence of any such technical shortcomings in IPv6 that prevent its deployment in any meaningful fashion. A more likely explanation for the current situation is an inability of a highly competitive deregulated industry to be in a position to factor longer-term requirements into short-term business logistics.

What Next?

Now we consider some questions relating to IPv4 address exhaustion. Will the exhaustion of the current framework that supplies IP addresses to service providers cause all further demand for addresses to cease at that point?

Or will exhaustion increase the demand for addresses in response to various forms of panic and hoarding behaviors in addition to continued demand from growth?

The size and value of the installed base of the Internet using IPv4 is now very much larger than the size and value of incremental growth of the network. In address terms the routed Internet currently (as of 14 August 2008) spans 1,893,725,831 IPv4 addresses, or the equivalent of 112.2 /8 address blocks. Some 12 months ago the routed Internet spanned 1,741,837,080 IPv4 addresses, or the equivalent of 103.8 /8 address blocks, representing a net annual growth of 10 percent in terms of advertised address space.

These facts lead to the observation that, even in the hypothetical scenario where all further growth of the Internet is forced to use IPv6 exclusively while the installed base still uses IPv4, it is highly unlikely that the core value of the Internet will shift away from its predominate IPv4 installed base in the short term.

Moving away from the hypothetical scenario, the implication is that the relative size and value of new Internet deployments will be such that these new deployments may not have sufficient critical mass by virtue of their volume and value as to be in a position to force the installed base to underwrite the incremental cost to deploy IPv6 and convert the existing network assets to dual-stack operation in this timeframe. The corollary of this observation is that new Internet network deployments will need to communicate with a significantly larger and valuable IPv4-only network, at least initially. The fact that IPv6 is not backward-compatible with IPv4 further implies that hosts in these new deployments will need to cause IPv4 packets with public addresses in their packet headers to be sent and received, either by direct deployment of dual stack or by proxies in the form of protocol-translating NATs. In either case the new network will require some form of access to public IPv4 addresses. In other words, after exhaustion of the unallocated address pools, new network deployments will continue to need to use IPv4 addresses.

From this observation it appears highly likely that the demand for IPv4 addresses will continue at rates comparable to current rates across the IPv4 unallocated address pool and after it is exhausted. The exhaustion of the current framework of supply of IPv4 addresses will not trigger an abrupt cessation of demand for IPv4 addresses, and this event will not cause the deployment of IPv6-only networks, at least in the short term of the initial years following IPv4 address pool exhaustion. It is therefore possible to indicate that immediately following this exhaustion event there will be a continuing market need for IPv4 addresses for deployment in new networks.

Although a conventional view is that this market need is likely to occur in a scenario of dual-stacked environments, where the hosts are configured with both IPv4 and IPv6, and the networks are configured to also support the host operation of both protocols, it is also conceivable to envisage the use of deployments where hosts are configured in an IPv6-only mode and network equipment undertakes a protocol-translating NAT function. In either case the common observation is that we apparently will have a continuing need for IPv4 addresses well after the event of IPv4 unallocated pool exhaustion, and IPv6 alone is no longer a sufficient response to this problem.

How?

If demand continues, then what is the source of supply in an environment where the current supply channel, namely the unallocated pool of addresses, is exhausted? The options for the supply of such IPv4 addresses are limited.

In the case of established network operators, some IPv4 addresses may be recovered through the more intensive use of NAT in existing networks. A typical scenario of current deployment for ISPs involves the use of private address space in the customer's network and NAT performed at the interface between the customer network and the service provider infrastructure (the CPE). One option for increasing the IPv4 address usage efficiency could involve the use of a second level of NAT within the service provider's network, or the so-called "carrier-grade" NAT option[9]. This option has some attraction in terms of increasing the port density use of public IPv4 addresses, by effectively sharing the port address space of the public IPv4 address across multiple CPE NAT devices, allowing the same number of public IPv4 addresses to be used across a larger number of end-customer networks.

The potential drawback of this approach is that of added complexity in NAT behavior for applications, given that an application may have to traverse multiple NATs, and the behavior of the compound NAT scenario becomes in effect the behavior of the most conservative of the NATs in the path in terms of binding times and access. Another potential drawback is that some applications have started to use multiple simultaneous transport sessions in order to improve the performance of the download of multipart objects. For single-level CPE NATs with more than 60,000 ports to be used for the customer network, this application behavior had little effect, but the presence of a carrier NAT servicing a large number of CPE NATs may well restrict the number of available ports per connection, in turn affecting the utility of various forms of applications that operate in this highly parallel mode. Allowing for a peak simultaneous demand level of 500 ports per customer provides a potential use factor of some 100 customers per IP address.

Given a large enough common address pool, this factor may be further improved by statistical multiplexing by a factor of 2 or 3, allowing for between 200 and 300 customers per NAT address. Of course such approximations are very coarse, and the engineering requirement to achieve such a high level of NAT usage would be significant. Variations on this engineering approach are possible in terms of the internal engineering of the ISP network and the control interface between the CPE NATs and the ISP equipment, but the maximal ratio of 200 to 300 customers per public IP address appears to be a reasonable upper bound without unduly affecting application behaviors.

Another option is based on the observation that, of the currently allocated addresses, some 42 percent of them, or the equivalent of some 49 /8 address blocks, are not advertised in the interdomain routing table, and are presumed to be either used in purely private contexts, or currently unused. This pool of addresses could also be used as a supply stream for future address requirements, and although it may be overly optimistic to assume that the entirety of this unadvertised address space could be used in the public Internet, it is possible to speculate that a significant amount of this address pool could be used in such a manner, given the appropriate incentives. Speculating even further, if this address pool were used in the context of intensive carrier-grade NATs with an achieved average deployment level of, say, 10 customers per address, an address pool of 40 /8s would be capable of sustaining some 7 billion customer attachments.

Of course, no such recovery option exists for new entrants, and in the absence of any other supply option, this situation will act as an effective barrier to entry into the ISP market. In cases where the barriers to entry effectively shut out new entrants, there is a strong trend for the incumbents to form cartels or monopolies and extract monopoly rentals from their clients. However, it is unlikely that the lack of supply will be absolute, and a more likely scenario is that addresses will change hands in exchange for money. Or, in other words, it is likely that such a situation will encourage the emergence of markets in addresses. Existing holders of addresses have the option to monetize all or part of their held assets, and new entrants, and others, have the option to bid against each other for the right to use these addresses. In such an open market, the most efficient usage application would tend to be able to offer the highest bid, in an environment dominated by scarcity tending to provide strong incentives for deployment scenarios that offer high levels of address usage efficiency.

It would therefore appear that options are available to this industry to increase the usage efficiency of deployed address space, and thereby generate pools of available addresses for new network deployments. However, the motive for so doing will probably not be phrased in terms of altruism or alignment to some perception of the common good. Such motives sit uncomfortably within the commercial world of the deregulated communications sector.

Nor will it be phrased in terms of regulatory impositions. It will take many years to halt and reverse the ponderous process of public policy and its expression in terms of regulatory measures, and the "common-good" objective here transcends the borders of regulatory regimes. This consideration tends to leave this argument with one remaining mechanism that will motivate the industry to significantly increase the address usage efficiency: monetizing addresses and exposing the costs of scarcity of addresses to the address users. The corollary of this approach is the use of markets to perform the address distribution function, creating a natural pricing function based on levels of address supply and demand.

References

[1] TCP/IP Mailing List, Message Thread: "Running out of Internet Addresses," November 1988.
http://www-mice.cs.ucl.ac.uk/multimedia/misc/tcp_ip/8813.mm.www/index.html#121

[2] F. Solenksy, "Internet Growth," Steering Group Report, p. 61, Proceedings of the 18th IETF Meeting, August 1990.
http://www.ietf.org/proceedings/prior29/IETF18.pdf

[3] G. Huston, "The IPv4 Internet Report," August 2008,
http://ipv4.potaroo.net

[4] P. Gross and P. Almquist, "IESG Deliberations on Routing and Addressing," RFC 1380, November 1992.

[5] K. Egevang and P. Francis, "The IP Network Address Translator (NAT)," RFC 1631, May 1994.

[6] G. Tsirtsis and P. Srisuresh, "Network Address Translation – Protocol Translation (NAT-PT)," RFC 2766, February 2000.

[7] C. Aoun and E. Davies, "Reasons to Move the Network Address Translator – Protocol Translator (NAT-PT) to Historic Status," RFC 4966, July 2007.

[8] M. Bagnulo, P. Matthews, and I. van Beijnum, "NAT64/DNS64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers," Internet Draft, work in progress, **draft-bagnulo-behave-nat64-00.txt**, June 2008.

[9] T. Nishitani and S. Miyakawa, "Carrier Grade Network Address Translator (NAT) Behavioral Requirements for Unicast UDP, TCP and ICMP," Internet Draft, work in progress, **draft-nishitani-cgn-00.txt**, July 2008.

[10] Olaf Maennel, Randy Bush, Luca Cittadini, Steven M. Bellovin, "A Better Approach than Carrier-Grade-NAT,"
http://rip.psg.com/~randy/080820.alt-to-cgn.pdf

[11] William Lehr, Tom Vest, Eliot Lear, "Running on Empty: The Challenge of Managing Internet Addresses," to be presented at the 36th Research Conference on Communication, Information and Internet Policy (TPRC), on 27 September 2008.
http://eyeconomics.com/backstage/References_files/Lehr-Vest-Lear-TPRC2008-080915.pdf

[12] Hain, Tony, "A Pragmatic Report on IPv4 Address Space Consumption," *The Internet Protocol Journal,* Volume 8, No. 3, September 2005

[13] **http://icann.org/en/announcements/proposal-ipv4-report-29nov07.htm**
(See also "Fragments" on page 46.)

GEOFF HUSTON is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He graduated from the Australian National University with a B.Sc. and M.Sc. in Computer Science. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of numerous Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005; he served on the Board of Trustees of the Internet Society from 1992 to 2001.
E-mail: **gih@apnic.net**

Letters to the Editor

I sincerely congratulate you for Geoff Huston's excellent article in *The Internet Protocol Journal,* June 2008, on the "Decade of Internet Evolution." The article shows an amazing insight into the Internet as it has recently evolved and deserves as wide an audience as possible.

The only comment I could make is that though Huston hints about separating the IP address from the host name, he does not explicitly mention the *Host Identity Protocol* (HIP)[1]. Previous issues of the Journal have this omission as well.

Note: As we struggle in the IETF and everywhere else in the industry with NAT traversal, mobility, and multihoming, we see countless approaches for each application layer protocol separately. HIP seems to fulfill the promise of solving these problems comprehensively.

Thanks for the privilege to continue reading the Journal; keep such papers coming.

—Henry Sinnreich, Adobe Systems, Inc.
hsinnrei@adobe.com

[1] R. Moskowitz, P. Nikander, P. Jokela, Ed., and T. Henderson, "Host Identity Protocol," RFC 5201, April 2008. See also: **http://www.ietf.org/html.charters/hip-charter.html**

The author responds:

Thank you for your generous comments.

At some point I was toying (dangerously!) with writing an article that attempted to predict the next 10 years, looking at what appears to be important today and what that could mean in the future. There is no doubt that the tight binding of identity and location is one of the assumptions that has made the Internet both simple and effective for the past decade. But where we sit today, in a world dominated by scale, mobility, a dense mesh of interconnectivity, highly capable end devices, dense middleware, and a panoply of specialized requirements, we need to look forward to methods that allow separation of identity and location. Now this separation could be at the level of the Internet Protocol itself, as in HIP or *Site Multihoming by IPv6 Intermediation* (SHIM6); or at the level of the transport session, as exemplified at present by the *Stream Control Transmission Protocol* (SCTP); or even at the application level, where the various offerings related to *Voice over IP* (VoIP) and *Peer-to-Peer* (P2P) have been working at the level of multiparty application rendezvous and application identity that sit on top of an adaptive platform of dynamic discovery of the characteristics of the underlying transport subsystem.

Each approach appears to offer some significant leverage in scaling the network in diverse ways, while at the same time presenting us with some fascinating insights into possible architectures that could address our needs in the next decade. No doubt the next 10 years will present us with some quite novel challenges with the imminent exhaustion of the unallocated IPv4 address pool and the associated observation that the schedule for the update of IPv6 has proceeded so slowly that we will be forced to be remarkably inventive with IPv4. HIP may well be a central part of such invention, but, more generally, I have no doubt that we will examine more generally how we can devise refinements to the networking model that preserve useful notions of identity across a rather fluid sea of shared location tokens.

Regards,

—Geoff Huston, APNIC
gih@apnic.net

Ten Years of IPJ

We received many congratulatory messages in response to our June 2008 Anniversary Issue. The following are some quotes from our readers:

"Compliments and congratulations for the tenth anniversary of this great Journal. It is great because it is making us realize the synergy between what has been and what is to come."

—John Okewole, Lagos, Nigeria

"This week I received the June 2008 issue of IPJ. I have been a subscriber for several years and it has been a great pleasure to find great contents in IPJ, such as the current issue that brings reviews on Internet evolution. I would like to send my congratulations to the IPJ team for 10 years of publication and my best wishes for future success."

—Frederico Fari, Belo Horizonte, Brazil

"I think that IPJ is a great journal. I hope you will not be forced to give up the paper edition because is a beautiful one (and it allows me to read during the evening hours when all computers and children in the house are shut down :–)"

—Andrea Montefusco, Rome, Italy

Book Reviews

Two Books on Cyber Law

Code and Other Laws of Cyberspace

Code and Other Laws of Cyberspace, by Lawrence Lessig, Basic Books, 1999, ISBN 0-465-03913-8. `http://code-is-law.org/`

Code 2.0

Code 2.0, by Lawrence Lessig, Basic Books, 2006, ISBN-10: 0-465-03914-6, ISBN 13: 978-0-465-03914-2. `http://codev2.cc/`

First published in 1999, then Harvard Law School Professor Lawrence Lessig's cautionary tale about the inescapable influence of certain material features of the built Internet has since become a foundational "Internet studies" text in universities and laws schools around the world. Lessig, who now occupies an endowed chair at Stanford Law School, makes a series of troubling observations about the Internet, his chosen sector of focus since setting aside his mid-1990s work on legal and institutional development in post-Soviet societies.

Lessig's key findings from that previous work are that rules matter—especially the sort of rules embodied in "constitutions" and other foundational institutions; that rules are artifacts of contingent human intent and design; and that rules can be changed. Being a "classical liberal" on the model of John Stuart Mill, Lessig advocates the sort of rules that afford maximum liberty for individuals against a triumvirate of coercive influences, including not only governments but also market power and oppressive social mores.

Now however, a fourth challenge to personal liberty has been exposed by the advent of the Internet—or rather, of *cyberspace,* which Lessig describes as the lived experience of participants in the rich application space that has been built atop the Internet. This new constraining factor is "architecture," which Lessig defines as "the built environment," or "the way the world is," that is, the cumulative result of all of the contingent historical events and decisions that have shaped the material circumstances confronting Internet users (or *cyberspace denizens*) today. *Code* is Lessig's term for the instruction sets (that is, programs, applications, etc.) that are the building blocks of the architecture of cyberspace; it is the stuff that emerges from the decision making of a relatively few (the *code writers*), which accretes over time into the less-malleable architecture that shapes the everyday choices and possibilities of everyone else whom the Internet or cyberspace touches.

New Code Means New Power(s)

According to Lessig, the code that defines cyberspace—which he calls "West Coast Code"—demands particular attention, both because of its omnipresence and because of how it differs from the other, more familiar factors that can impinge on individual liberty.

Like the canons of law (also known as "East Coast Code"), code is basically a collection of rules written with human goals and objectives in mind. However, in its effects code more closely resembles the laws of nature, because it requires neither the awareness nor the consent of its subjects in order to be effective. Although this claim sounds suspiciously like a variant, or perhaps an illustration of Arthur C. Clarke's *Third Law of Prediction* (which states that any sufficiently advanced technology will be indistinguishable from the supernatural), there is purpose behind Lessig's observation. The self-enforcing character of code is doubly problematic in the case of cyberspace, he suggests, because unlike the law, code affords no appeal, no recourse, and no formal, institutional review and interpretation of the kind that lawyers and judges exercise in legal matters. Without such expert oversight, code might come to be used as a tool to subvert individual liberties or public values, for either commercial or political gain, without anyone's being the wiser. In fact, he implies, the lack of transparency of code almost invites such abuses.

At this point some might be tempted to dismiss Lessig's program as just "sour grapes" from a high-profile industry spokesman sensing this erosion of the traditional prominence and centrality of his profession in a new code-centric world. Lessig believes passionately in the exercise of law and judicial review as master tools for keeping other important forces—government power, market power, and social norms—broadly aligned with "important public values." He extols the relationships among the rule of law, democracy, and politics, the latter of which invests law with legitimacy to raise or lower the cost of particular individual actions (for example, by taxing, criminalizing, valorizing, or subsidizing them) to encourage conformity with publicly chosen goals and values. He observes that "architecture is a kind of law" and that "code codifies values, and yet, oddly, most people speak as if code were just a question of engineering." It takes no great leap of imagination to conclude that code too should be subject to the same kind of legal and judicial oversight that keeps the rest of society running smoothly. Eliminating any doubt, Lessig asserts that:

> Technology is plastic. It can be remade to do things differently. We should expect—and demand—that it can be made to reflect any set of values that we think important. The burden should be on the technologists to show us why that demand can't be met.

However, such a dismissal would indeed be too easy, for Lessig also expresses misgivings about the professionalization and segregation of "constitutional thinking" within the legal sector. "Constitutional thought has been the domain of lawyers and judges for too long," Lessig writes, and as a result everyone else has grown less comfortable—and also less competent—in engaging in fruitful conversation about fundamental, "constitutional" values.

And yet Lessig suggests that this skill has also atrophied within the legal community, as more and more jurists have embraced an "originalist" interpretive philosophy that holds that the U.S. Constitution provides no guidance for how to resolve conflicts between old values—what Lessig calls *latent ambiguities*—or how to address wholly novel concerns raised by technologies such as the Internet. Originalists (Lessig mentions U.S. Supreme Court Justice Antonin Scalia) assert that in such cases the only recourse is the political and legislative processes—where, one assumes, limited experience with both technology and constitutional debate make the prospects for success even dimmer. Lessig writes that "We (legal scholars) have been trapped by a mode of reasoning that pretends that all the important questions have already been answered," but that "the constitutional discourse of our present Congress is far below the level at which it must be to address the questions about constitutional values that will be raised by cyberspace."

Diagnosis from a Distance

Lessig is without question eminently qualified to make such observations about his home-turf legal and political spheres. However, it is less clear that his blanket charge of deliberative incompetence is equally valid across the full range of Internet and cyberspace stakeholders. Neither is it clear that the architecture of cyberspace is as uniquely problematic as he suggests, compared to the architecture of other, more familiar domains. Finally, Lessig's own admittedly limited technical expertise may lead him to misapprehend the boundary between cyberspace and the Internet, and to underestimate the radicalness of his proposed cyberspace fix.

Taking these ideas in reverse order, Lessig's conception of the structural and functional distinction between the Internet and cyberspace merits closer scrutiny. As explained later, Lessig advocates profound technical changes to bring the functions of code under the rule of law (or laws, because Lessig wishes to accommodate subsidiary jurisdictions as well as sovereign differences in law). However, he envisions this intervention affecting only the "code" domain, not the "Internet's core protocols":

> When I speak about regulating the code, I'm not talking about changing these core TCP/IP protocols...In my view these components of the network are fixed. If you required them to be different, you'd break the Internet. Thus rather than imagining the government changing the core, the question I want to consider is how the government might either (1) complement the core with technology that adds regulability, or (2) regulate applications that connect to the core.

Lessig's specific ideas for achieving this function while preserving the core are not fully detailed in this context until *Code 2.0* (2006), which Lessig describes as an update rather than a full rewrite, albeit one with new relevance to match a "radically different time." The central idea involves the introduction of an "identity layer" that permits authoritative in-band querying and signaling of the jurisdiction(s) to which every would-be Internet user is subject. The deployment of this system would be accompanied by the development of a comprehensive distributed database of Internet usage restrictions mandated by every legally recognized jurisdiction around the world. Together, these components would operate as a kind of "domain interdiction system" that would automatically black-hole all Internet resource queries that are legally impermissible to individuals based on their jurisdiction(s) of origin, regardless of their actual location.

This proposal is clearly vulnerable to criticism of many kinds—technical, ethical, practical, etc.—and to be fair Lessig anticipates and preemptively responds to several of the most obvious ones. Space limitations preclude any review of those arguments here, but it is impossible to resist a few short observations. First, it is not clear why Lessig imagines that his proposed system would be anything less than a fundamental intervention in the core function and protocols of the Internet. Today several different high-profile technical developments that could plausibly be described as changing TCP/IP are under way, but they (hopefully) will not break the Internet. At the same time, TCP/IP is not the only technology that is essential to the Internet "core." The system that Lessig advocates is clearly inspired by the *Domain Name System* (DNS), it would of necessity be similarly global and ubiquitous in scope and scale, and it would likely function by selectively blocking some DNS responses based on the initiator's identity. Although some once regarded the DNS as a mere application (for example, shortly after it was invented), few today would categorize it as anything other than a core protocol. Also, given the degree to which any implementation of the proposed identity system would preempt many "normative" features that are associated with the Internet core (for example, the principles behind the *end-to-end* arguments), it is unclear what would remain "unbroken" therein that might still warrant any special consideration or separate treatment. We can only hope that Lessig's optimism on this question is justified, because looming developments in certain wireless standards as well as in the management of IP addressing may provide for more concrete—and less revisable—answers in the very near future.

Objects in View May Be Closer Than They Appear

Then there is the question of how much code really makes the architecture of cyberspace different from the architecture of other domains. Many of Lessig's claims on this point date back to the first version of the book, when Internet exceptionalism was still new enough for deflationary counterarguments to seem provocative.

Although the revolutionary potential of the Internet continues to inspire many (this reviewer included), the past decade of booms, busts, compromises, and indictments have done much to temper that faith. It is not that Lessig's concerns about the opaque nature of cyberspace architecture, about the substantial influence that code writers and network owners command, and about the vulnerability of the whole system to a crisis-induced authoritarian turn aren't reasonably well-founded. But they are equally apropos to most other important spheres of life. The phrase "possession is nine-tenths of the law" has multiple meanings, and was coined many decades before the Internet was invented. The inexplicability of many current "real-world" legislative and judicial outcomes without recourse to some cynical theory of unacknowledged interests and unobservable influence certainly raises many questions about the architecture of the space beyond cyberspace. And Lessig's warnings about national security fears precipitating a sudden loss of freedoms (taken from Jonathan Zittrain's *Z-Theory*) now seem prophetic—albeit less for the Internet than for the earliest and largest host society of the Internet. One might observe that Lessig is guilty of his own kind of exceptionalism—one that, ironically, may obscure the degree to which constitutional challenges in the real and virtual worlds are more or less the same. In fact, Lessig's subsequent shift of priorities from code to intellectual property law recently ended with a return to his original home turf of law and politics—perhaps in belated recognition that sometimes, even when you have a good story, East Coast Code is still the only durable recourse.

Finally, there is the question of constitutional acumen. This question is the critical one for Lessig (he uses some form of the term *constitution* more than 250 times in the main text), because for him the term evokes nothing less than "an architecture... a way of life that structures and constrains social and legal power, to the end of protecting fundamental values." In this sense, he adds, constitutions are built rather than found. Moreover, they have been built in different (albeit sometimes overlapping) places by different institutions and societies, many with quite different conceptions of which fundamental values to uphold. From whence will the architecture of values of cyberspace emerge? Who will be its authors? Lessig never quite gives a final answer, even for his own home jurisdiction, but he does help to winnow out several likely suspects. As noted previously, he invests little faith in the current U.S. legislative branch. He also has reservations about many members of his own legal profession, although the need to preserve backward compatibility with the primary U.S. Constitution and to reconcile newly revealed "latent ambiguities" therein obviously recommends some legal training at the very least. Government and industry represent the most likely perpetrators of liberty-undermining code, Lessig claims, so he looks for no help from those quarters.

In the end Lessig provides some oblique advice for judges (abandon formalism), hackers (open source), and voters (educate yourself, and don't give up hope), but ultimately concludes with a call for more lawyerly deliberation: if only our leaders could act more like lawyers, telling stories that persuade "not by hiding the truth or exciting the emotion, but by using reason," and our fellow citizens could act like juries, resisting the fleeting passions of the mob and making decisions based on the facts alone, then perhaps we could overcome the architectural challenges of both cyberspace and physical space.

Story Boards and Internet Constitutions

Notwithstanding its solipsistic aspects, advice like that discussed in the last section is hard to find fault with. Professor Lessig is unquestionably a person of good conscience, and has a long, distinguished, and very well-documented record of putting this advice into practice in a wide range of good causes, including many that are wholly unrelated to code or cyberspace. However, one could argue (perhaps with equal solipsism) that many of the behaviors and virtues that he commends are now regularly on display in the mailing lists, message boards, and other deliberative records of the Regional Internet Registries, the IETF, and the IAB—in particular in discussions on the form that IPv4 and IPv6 address-allocation policies should take, in the design of future routing systems that balance scalability with the freedom to choose between competing providers, and in the reconciliation of traditional policies and their beneficiaries with the changing realities of Internet resource stewardship. Closer scrutiny of these records reveals that successful consensus policies are almost invariably borne of good, well-reasoned stories, the vast majority of which are offered by individuals who are affiliated neither with government agencies nor with any of the largest and most powerful ISPs. Many of the storytellers are old hands, but new voices regularly emerge and command attention based on nothing more than the strength of their reasoning. Participating in these discussions, one can *occasionally* experience the same feeling that inspires Lessig in the courtroom, where "some, for the first time in their lives, see power constrained by reason. Not by votes, not by wealth, not by who someone knows—but by an argument that persuades."

That this "architectural" work has gone largely unrecognized to date in law schools, university humanities and social science departments, and even in some civil society-oriented Internet governance fora is not entirely unexpected, because the context and terminology of those discussions is invariably technical, even if many participants recognize that the underlying principles are essentially "constitutional" in nature. No doubt a more complete conversation between code writers and constitutionalists is inevitable over time, and with luck more cross-fertilization will lead to better protocols, better policies, and better architecture.

However, this rapprochement is unlikely to be initiated by technologists seeking to take up the study and application of legal principles. Lessig, whose own intellectual project builds substantially on the antiformalist, "legal realist" school of thought, should understand this reality better than most. In the crudest of forms, legal realism holds that "the Law is whatever lawyers happen to say it is." Stated as neither a boast nor a claim of entitlement but rather as a practical observation of the challenges that lawyers face in applying ambiguous old laws to incommensurable new circumstances, this maxim nevertheless clearly conveys a sense of both the great responsibility and the great power that lawyers command. Perhaps it is time that Mr. Lessig and his counterparts consider the possibility that a similar school of thought may inform (consciously or unconsciously) the perspectives of network builders and code writers. Being of no less good conscience, perhaps code writers and other "cyberspace realists" are merely waiting for the moment when the Law and lawyers come calling with a good story, under the banner of reason rather than power. So long as the story now unfolding continues to make sense and satisfy the ever-expanding audience, we needn't fear either.

Code may not be *that* particular story, but it's an excellent read, and an important contribution to a dialogue that must be engaged.

—Tom Vest
tvest@eyeconomics.com

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

Global Policy Proposal for Remaining IPv4 Address Space

Global Internet Number Resource Policies are defined by the *Address Supporting Organization* (ASO) MoU[1]—between the *Internet Corporation for Assigned Names and Numbers* (ICANN) and the *Number Resource Organization* (NRO)—as "Internet number resource policies that have the agreement of all RIRs according to their policy development processes and ICANN, and require specific actions or outcomes on the part of the *Internet Assigned Numbers Authority* (IANA) or any other external ICANN-related body in order to be implemented." Attachment A of this MoU describes the *Development Process of Global Internet Number Resource Policies,* including the adoption by every *Regional Internet Registry* (RIR) of a global policy to be forwarded to the ICANN Board by the ASO, as well as its ratification by the ICANN Board. In this context, the ICANN Board adopted its own Procedures[2] for the Review of Internet Number Resource Policies Forwarded by the ASO for Ratification.

Among other features, these Procedures state that the Board will decide, as and when appropriate, that ICANN staff should follow the development of a particular global policy, undertaking an "early awareness" tracking of proposals in the addressing community. To this end, staff should issue background reports periodically, forwarded to the Board, to all ICANN Supporting Organizations and Advisory Committees and posted at the ICANN Web site.

At its meeting on 20 November 2007, the Board resolved to request tracking of the development of a global policy proposal for allocation of remaining IPv4 address space, under discussion in the Regional Internet Registries. The status overview presented below is compiled in response to this request and will be further updated as developments proceed, for information to ICANN entities and the wider community. This is the fifth issue of the tracking of this policy.

Originally, two slightly different global policy proposals were introduced for allocation of the remaining IPv4 address space:

- A version (1) "Global Policy for the Allocation of the Remaining IPv4 Address Space," first presented at LACNIC X in May 2007
- A version (2) "End Policy for IANA IPv4 allocations to RIRs," first presented at APNIC 24 in September 2007

Both featured the same approach, distribution of an equal number *N* of /8 IPv4 address blocks to each RIR when the IANA free pool would reach the threshold value of $5 \times N$, but differed in the proposed value of *N*, notably 2 or 1, respectively. The proposals were discussed in parallel in the RIRs and regarded essentially as one proposal, with a view to converging on a value for *N*. In February 2008, agreement was reached for a unified proposal (3).

The current proposal is thus:

- Version (3) "Global Policy for the Allocation of the Remaining IPv4 Address Space," first presented at APNIC 25 in February 2008.

The proposal was introduced at the subsequent meetings of all other RIRs. It has now been adopted in ARIN, AfriNIC, LACNIC and RIPE, and is in final call in APNIC. If adopted by all the RIRs, the proposal will subsequently be handled by the NRO Executive Council and the ASO Advisory Council according to their procedures before being submitted to the ICANN Board for ratification. A table[3] can be found on the ICANN Website that indicates the status within each RIR for the current proposal. Hyperlinks are included for easy access.

It should be noted that other policy proposals have been put forward and are being discussed regarding IPv4 address space exhaustion, although only those mentioned above have been scoped as global policy proposals in the sense of the ASO MoU, that is, focusing on address allocation from IANA to the RIRs, and recognized by the ASO AC as global policy proposals in that meaning.

[1] **http://aso.icann.org/docs/aso-mou2004.html**

[2] **http://icann.org/en/general/review-procedures-pgp.html**

[3] **http://www.icann.org/en/announcements/proposal-ipv4-report-29nov07.htm**

Upcoming Events

The *Internet Engineering Task Force* (IETF) will meet in Minneapolis, Minnesota, November 16 – 21, 2008. In 2009, IETF meetings are scheduled for San Francisco, California (March 22 – 27), Stockholm, Sweden (July 26 – 31) and Hiroshima, Japan (November 8 – 13). For more information see **http://www.ietf.org/**

The *North American Network Operators' Group* (NANOG) will meet in Los Angeles, California, October 12 – 14. Immediately following the NANOG meeting, the *American Registry for Internet Numbers* (ARIN) will meet in the same location, October 15 – 17. See **http://nanog.org** and **http://arin.net**

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in Cairo, Egypt, November 2 – 7, 2008. For more information see: **http://icann.org**

The *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will be held in Manila, Philippines, February 18 – 27, 2009. See: **http://www.apricot2009.net/**

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

December 2008 Volume 11, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

ISSN 1944-1134

From The Editor

Response to our use of a new printing paper has been very positive, so we will continue to use the uncoated and recycled Exact® paper introduced with our September 2008 issue. We are still interested in hearing your feedback on the paper, as well as any other aspect of this journal. Send your comments to: **ipj@cisco.com**

The last decade has seen many developments in the area of *wireless* networking technologies. Wireless Internet access is now available in thousands of locations ranging from private homes to hotels, trains, airplanes, ships at sea, and even entire cities. Wireless systems, specifically Bluetooth, are also used for short-range device connectivity such as between a mobile phone and a headset, while WiMAX systems are being deployed for larger area coverage. In our first article, T. Sridhar gives an overview of Wi-Fi, Bluetooth, and WiMAX.

As stated in our previous issue, the topic of IP Version 4 address exhaustion and migration to IP Version 6 is being debated in many Internet-related organizations, including the IETF, *Internet Corporation for Assigned Names and Numbers* (ICANN), and the *Regional Internet Registries* (RIRs). In our last issue, Geoff Huston outlined the history of IPv4 address depletion. This time we bring you the first in a two-part series of articles entitled "The End of Eternity." The article is by Niall Murphy and David Wilson. Part Two will follow in our March 2009 issue. As you will see from our "Letters to the Editor," views on the right way to tackle the address exhaustion and protocol migration challenge abound, and I predict we will carry yet more articles on this topic in future issues.

Just over 10 years ago, Jonathan B. Postel, Internet pioneer and a key player in many core Internet activities, passed away. In this issue we bring you a remembrance article written by another Internet pioneer, Vint Cerf. In connection with this anniversary, special events were held in Minneapolis in conjunction with the 73rd meeting of the IETF. The *Jonathan B. Postel Service Award* for 2008 was awarded to EsLaRed of Venezuela by a committee of former award winners. You will find more information about the award in our "Fragments" section on page 42.

Remember to let us know if your mailing address changes and to visit our online companion, *The Internet Protocol Forum,* where you will find additional articles and other material: **http://ipjforum.org**

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Wi-Fi, Bluetooth and WiMAX—Technology and Implementation

by T. Sridhar, Flextronics

Wireless networks can be classified broadly as *Wireless Personal-Area Networks* (WPAN), *Wireless LANs* (WLANs), and *Wireless Wide-Area Networks* (WWANs). WPANs operate in the range of a few feet, whereas WLANs operate in the range of a few hundred feet and WWANs beyond that. In fact, wireless WANs can operate in a wide range—a metropolitan area, cellular hierarchy, or even on intercity links through microwave relays.

This article examines wireless technologies for the WLAN, WPAN, and WWAN areas, with specific focus on the IEEE 802.11 WLAN (often known as Wi-Fi®), *Bluetooth* (BT) in the WPAN, and WiMAX for WWAN as representative technologies. It discusses key aspects of the technology—medium access and connectivity to the wired network—and concludes by listing some common (mis)perceptions about wireless technology.

WLANs

The *Institute of Electrical and Electronic Engineers* (IEEE) defined three major WLAN types in 802.11–802.11 b and g, which operate in the 2.4-GHz frequency band, and 802.11a, which operates in the 5-GHz band. The 2.4- and 5-GHz bands used here are in the license-free part of the electromagnetic spectrum, and portions are designated for use in *Industrial, Scientific, and Medical* (ISM) applications—so these portions are often called ISM bands. More recently, a high-speed 802.11 WLAN has been proposed—the 802.11n WLAN, which operates in both the 2.4- and 5-GHz bands.

The 2.4-GHz frequency band used for 802.11 is the band between 2.4 and 2.485 GHz for a total bandwidth of 85 MHz, with 3 separate nonoverlapping 20-MHz channels. In the 5-GHz band, there are a total of 12 channels in 3 separate subbands—5.15 to 5.25 GHz (100 MHz), 5.25 to 5.35 GHz (100 MHz), and 5.725 to 5.825 GHz (100 MHz).

The more common mode of operation in 802.11 is the *infrastructure* mode, where the stations communicate with other wireless stations and wired networks (Ethernet typically) through an *access point*. The other mode is the *ad-hoc* mode, where the stations can communicate directly with each other without the need for an access point; we will not discuss this mode in this article. The access point bridges traffic between wireless stations through a lookup of the destination address in the 802.11 frame (see Figure 1a).

The *Media Access Control* (MAC) header of 802.11 has four addresses. Depending upon the value of a *FromDS* (from access point), or a *ToDS* (to access point) bits in the header (see Figure 1b), the addresses have different connotations. The first two addresses are for the receiver and transmitter, respectively.

Figure 1a: WLAN Network with Ethernet Connectivity

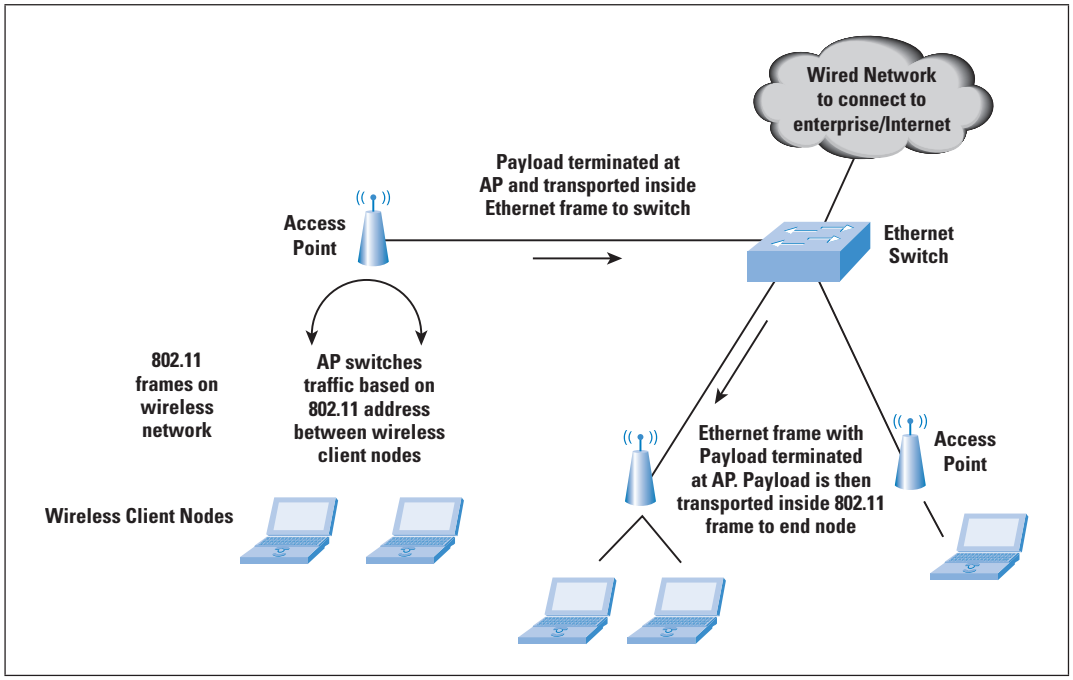
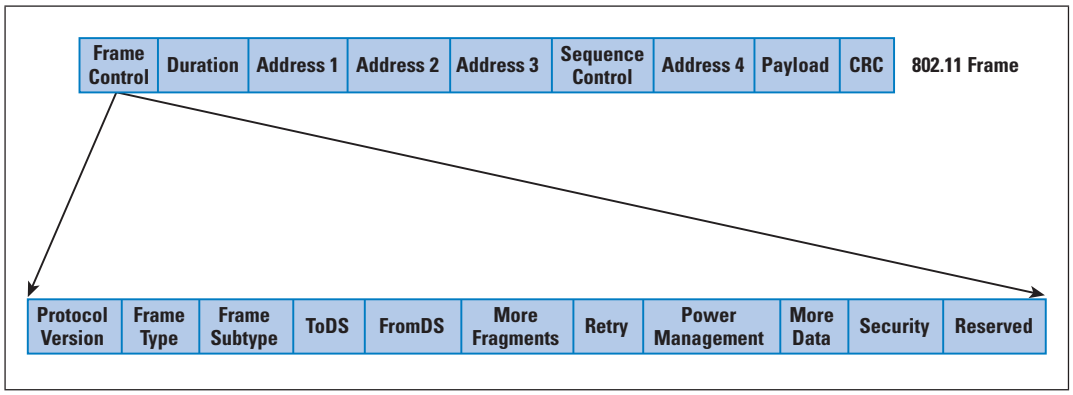


Figure 1b: 802.11 Frame Format



Address 4 is not used except when both FromDS and ToDS are set to 1—it is for a special mode of communication for access point-to-access point traffic, whence addresses 3 and 4 refer to the source- and destination-station MAC addresses, respectively, whereas addresses 1 and 2 refer to the access point addresses (that is, the transmitter and receiver on this inter-access point channel). When FromDS is set to 1, address 1 is the destination-station MAC address, address 2 is the access point address, and address 3 is the source-station MAC address. When ToDS is set to 1, address 1 is the access point MAC address, address 2 is the transmitting-station MAC address, and address 3 is the destination-station MAC address.

Although earlier versions of 802.11 LANs used *Frequency Hopping Spread Spectrum* (FHSS), 802.11b typically uses *Direct Sequence Spread Spectrum* (DSSS) for 1-, 2-, 5.5-, and 11-Mbps speeds. Both schemes involve transmission of a narrowband signal over a wider frequency range to mitigate the possibility of interference at any one frequency. The nodes and access points typically transmit at the highest data rate possible based on the current signal-to-noise ratio.

At the MAC level, 802.11 LANs involve the use of *Carrier Sense Multiple Access/Collision Avoidance* (CSMA/CA). Stations back off if they detect that another station is transmitting on that channel. The station then waits for a random period after the end of the transmission before it attempts to transmit on that channel. In addition, control frames such as *Request to Send* (RTS) and *Clear to Send* (CTS) are used to facilitate the actual data transfer. The CTS control frame has the duration for which the transmitting node is allowed to transmit. Other stations sense this frame and back off for at least the specified duration before sensing the radio link again.

When the access points are connected through a LAN, the entire system is known as a *Distribution System*. The access points perform an integration function—that is, bridging between wired and wireless LANs. In this scenario, (see Figure 1a) the wireless control and data frames are terminated at the access point or tunneled from the access point to a centralized controller over Ethernet. When terminated at the access point, the payload is transmitted from the access point to the network over Ethernet. This transmission is done in the following manner:

The source and destination addresses are set to the station and access point addresses, respectively. At the access point, the payload is stripped from the 802.11 data frame and sent as part of an Ethernet packet either as a broadcast packet or to a specific destination. If the packet sizes (when reassembled) are larger than the Ethernet frame size, they are discarded. In the reverse direction, the Ethernet frame can be directly encapsulated into an 802.11 frame for transmission from the access point to the end node. At the WLAN end node, the complete Ethernet frame shows up at the driver level as though it were a frame received on a pseudo Ethernet interface.

The most common 802.11b WLAN speed is 11 Mbps. However, based on the interframe spacing, preamble, header encapsulation, and acknowledgements for frames required, the actual throughput for user data would be about 50 percent of the actual speeds. This throughput of 50 percent of actual link speed is a common theme on 802.11g and 802.11a also.

Stations connect to the access point through a scanning process. Scanning can be passive or active. In the passive mode, the station searches for access points to find the best access point signal (which contains the *Service Set Identifier* [SSID], data rates, and so on).

The access point frame that the stations look for is a management frame known as the *beacon frame*. In the active mode, the station initiates the process by broadcasting a probe frame. All access points that receive the probe send back a probe response, helping the station build up the list of available access points. The sequence of a station "connecting" to an access point involves two steps. The first is *authentication,* where the station sends an authentication request frame to the access point. Depending upon the authentication through 802.1X or internal configuration, the access point can accept or reject the request with an authentication response. The second step is *association,* which is required to determine the data rates supported between the access point and the station. At the end of the association phase, the station is allowed to transmit and receive data frames.

Power Concerns in 802.11

Although it is not a part of the standard, the access points might adjust their transmitting power based on the environment they are in (they do have maximum limits based on regional restrictions). If they do not perform this adjustment, all the stations might connect to the access point with the highest transmitting power, even if the access point is far away. The other concern is, of course, the interference between access points. The power adjustment is usually done through configuration and, in some cases, through a monitoring function on the network. In the latter case, the monitoring function reports the information to a central controller.

A new initiative within the IEEE (802.11k) has been started to improve traffic distribution within the network. Specifically, it addresses the problem of access point overloading so that stations can connect to underused access points for a more efficient use of network resources.

With respect to power management on the client side, a station can indicate that it is going into a "sleep" or low-power state to the access point through a status bit in a frame header (refer to Figure 1b). The access point then buffers packets for the station instead of forwarding them to the station as soon as they are received. The sleeping station periodically wakes up to receive beacons from the access point. The beacons include information about whether frames are being buffered for the station. The station then sends a request to the access point to send the buffered frames. After receiving the frames, the station can go back to sleep.

802.11a/g Technology—Orthogonal Frequency-Division Multiplexing

Sometimes called *discrete multitone* (DMT) in the *Digital Subscriber Line* (DSL) world, *Orthogonal Frequency-Division Multiplexing* (OFDM) is used as the underlying technology in 802.11g and 802.11a. OFDM is a form of *Frequency-Division Multiplexing* (FDM); normally, FDM uses multiple frequency channels to carry the information of different users. OFDM uses multicarrier communications, but only between one pair of users—that is, a single transmitter and a single receiver.

Multicarrier communications splits a signal into multiple signals and modulates each of the signals over its own frequency carrier, and then combines multiple frequency carriers through FDM. OFDM uses an approach whereby the carriers are totally independent of (orthogonal to) each other. Note that the total bandwidth consumed with OFDM is the same as with single carrier systems even though multiple carriers are used—because the original signal is split into multiple signals. OFDM is more effective at handling narrowband interference and problems related to multipath fading, simplifying the building of receiver systems.

We can illustrate this process with a simple example—one often used in discussions about OFDM. For a "normal" transmission at 1 Mbps, each bit can take 1 microsecond to send. Consider bit 1 and bit 2 sent with a gap of 1 microsecond. If two copies of bit 1 are received at the destination, one of them is the reflected or delayed copy. If the delay is around 1 microsecond, this delayed copy of bit 1 can interfere with bit 2 as it is received at the destination because they arrive at approximately the same time. Now consider an OFDM transmission rate of 100 kbps, that is, the bits are sent "slower" but over multiple frequencies. A multipath delay of around 1 microsecond will not affect bit 2, because bit 2 is now arriving much slower (around 10 microseconds). The delay in bit arrival (1 microsecond in our example) is not a function of the transmission—rather it is due to the various paths taken by the signal.

Orthogonal Frequency-Division Multiple Access (OFDMA) superimposes the multiple-access mechanism on OFDM channels, so that multiple users can be supported through subsets of the subcarriers assigned to different users. Note that 802.16-2004 ("Fixed" WiMAX) uses OFDM, whereas 802.16e-2005 ("Mobile" WiMAX) uses OFDMA.

MIMO and 802.11n

Multiple Input Multiple Output (MIMO) antennas are the basis for the 802.11n wireless LAN standard, currently in draft form but on the way to final standardization. Signals often reflect off objects and are received at different times and strengths at the receiver, resulting in a phenomenon called *multipath distortion*. (Note: 802.11n in this article implies the draft 802.11n standard at the time of writing.) MIMO actually takes advantage of this distortion by sending a single data stream split into multiple parts to be transmitted from multiple antennas (typically 3 in 802.11n) and letting the reflected signals be processed at the receiver (through multiple antennas). The transmission of multiple data streams over different spatial channels, sometimes known as *Space Division Multiplexing* (SDM), also allows a larger amount of data to be sent over the air. Through advances in the *Digital Signal Processing* (DSP)-based processing, the receiver can process the signals, cross-correlate them, and reconstitute them accurately despite interference. Also, because of the multiple signals received over multiple paths, link reliability is increased.

The 802.11n standard uses three antennas and also supports two radios (for the 2.4- and 5-GHz bands where 802.11n can operate). It can also use 40-MHz channels through *channel bonding*—that is, two adjacent 20-MHz channels are combined into a single 40-MHz channel, possibly resulting in a data rate of up to 150 Mbps of effective throughput.

One concern with 802.11n that is starting to gain attention is the power requirement of 802.11n access points. With radios in both bands and the use of MIMO, 802.11n access points tend to consume more power than the 802.11 a/b/g access points, leading to problems when the access point is powered by *Power over Ethernet* (PoE) power-sourcing equipment. The 802.3af standard permits a maximum of 12.95W per Ethernet port, which is often less than the power that most 802.11n APs need. The IEEE 802.3at working group is working toward a higher-power PoE standard. This initiative, commonly called *PoE Plus*, will peak at 25W per Ethernet port (on Category 5 Ethernet cable).

Ethernet Backhaul

The access point has two primary functions—connecting wireless clients to each other as well as connecting wireless and wired clients. In the latter, the access point can act as an Ethernet bridge by passing Layer 2 frames between the wired and wireless networks, or as a router, terminating WLAN and Ethernet Layer 2 frames and performing IP-level forwarding. The Layer 3 routing model is less popular and we will not consider it here.

The access point typically terminates WLAN management and control frames. However, there is another model of a *thin access point* wherein these frames can be backhauled to a WLAN switch for processing. The access point connection to the wired network is typically an Ethernet link to a dedicated Ethernet switch port at 100-Mbps or Gigabit Ethernet speeds. With the advent of 802.11g and 802.11a WLANs, 10-Mbps links are not sufficient because these WLANs can operate at close to 27-Mbps throughput over the wireless network.

When considering 802.11n, we find that 100-Mbps backhaul links to the switch are insufficient for the 802.11n throughput of 150, or even 300 Mbps with channel bonding. Gigabit Ethernet links are often considered for connectivity between the 802.11n access point and the Ethernet switch. The next speed for Ethernet connectivity is 10 Gbps, which is well-established in the enterprise for data center and core Ethernet network applications. Work is ongoing in the IEEE for 40- and 100-Gbps Ethernet, so that should cover advances in wireless speeds for efficient backhaul to the wired network.

Bluetooth

Bluetooth started as a "wire-replacement" protocol for operation at short distances. A typical example is the connection of a phone to a PC, which, in turn, uses the phone as a modem (see Figure 2). The technology operates in the unlicensed 2.4-GHz ISM band. The standard uses FHSS technology. There are 79 hops in BT displaced by 1 MHz, starting at 2.402 GHz and ending at 2.480 GHz.

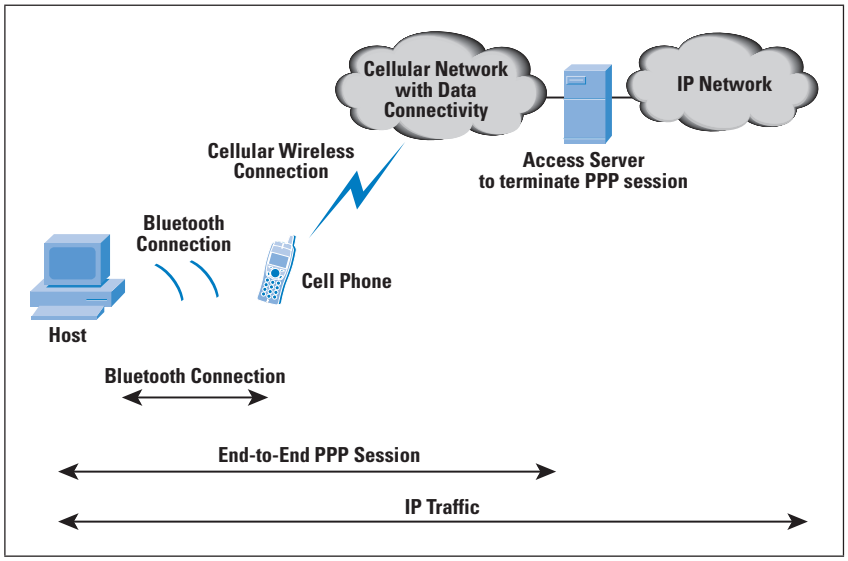


Figure 2: Typical Use of a Bluetooth enabled phone as a data modem for a PC

Bluetooth belongs to a category of *Short-Range Wireless* (SRW) technologies originally intended to replace the cables connecting portable and fixed electronic devices. It is typically used in mobile phones, cordless handsets, and hands-free headsets (though it is not limited to these applications). The specifications detail operation in three different power classes—for distances of 100 meters (long range), 10 meters (ordinary range), and 10 cm (short range).

Bluetooth operates in the unlicensed ISM band at 2.4 GHz (similar to 802.11 b/g wireless), but it is most efficient at short distances and in noisy frequency environments. It uses FHSS technology—that is, it avoids interference from other signals by hopping to a new frequency after transmitting and receiving a packet. Specifically, 79 hops are displaced by 1 MHz, starting at 2.402 GHz and finishing at 2.480 GHz.

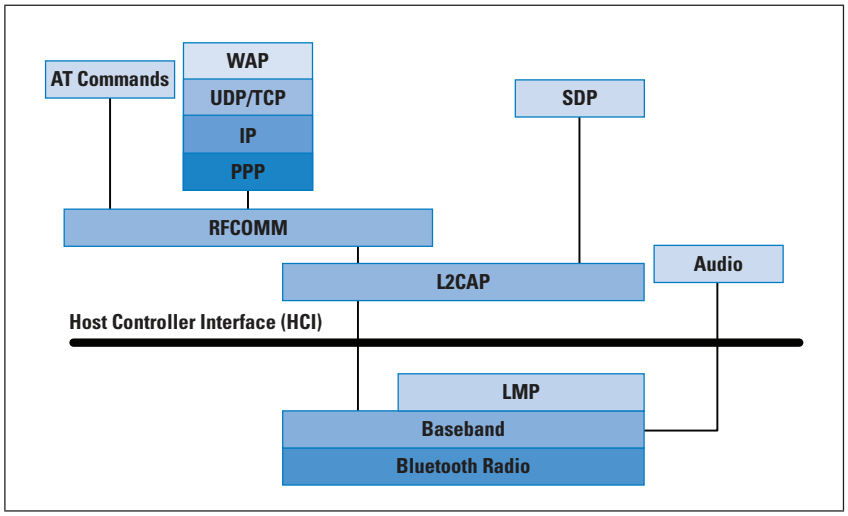
Bluetooth can operate in both point-to-point and logical point-to-multipoint modes. Devices using the same BT channel are part of a *piconet* that includes one master and one or more slaves. The master BT address determines the frequency hopping sequence of the slaves. The channel is also divided into time slots, each 625 microseconds in duration. The master starts its transmission in even-numbered time slots, whereas the slave starts its transmission in odd-numbered slots.

BT specifies two types of links, a *Synchronous Connection-Oriented* (SCO) link and an *Asynchronous Connectionless Link* (ACL). The SCO link is a symmetric point-to-point link between a master and a single slave in the piconet, whereas the ACL link is a point-to-multi-point link between the master and all the slaves participating in the piconet. Only a single ACL link can exist in the piconet, as compared to several individual SCO links.

Bluetooth Stack

Other than the radio and baseband components (the physical layer of Bluetooth that manages physical channels and links), the Bluetooth stack (see Figure 3) includes a *Link Manager Protocol* (LMP) used for link management between the endpoints, a *Logical Link Control and Adaptation Protocol* (L2CAP) for the data link, a *Radio Frequency Communication* (RFCOMM) protocol to provide emulation of serial ports over L2CAP, and a *Service Discovery Protocol* (SDP) for the dynamic discovery of services—because the set of services changes dynamically based on the RF proximity of the devices. In addition, the *Host Controller Interface* (HCI) provides a uniform command interface to the baseband controller and the link manager to have access to the hardware registers.

Figure 3: Key Elements of the Bluetooth Stack



LMP is required for authentication, encryption, switching of roles between master and slave, power control, and so on. L2CAP provides both connection-oriented and connectionless data services functions, including protocol multiplexing, segmentation and reassembly, and piconet-based group abstraction. As part of the multiplexing function, L2CAP uses the concept of channels, with a channel ID representing a logical channel endpoint on a BT device. L2CAP offers services to the higher layers for connection setup, disconnect, data reading and writing, pinging the endpoint, and so on.

RFCOMM, which provides emulation of serial ports on the BT link, can support up to 60 simultaneous connections between two BT devices. The most common emulation is of the RS-232 interface, which includes emulation of the various signals of this interface such as *Request To Send* (RTS), *Clear To Send* (CTS), *Data Terminal Ready* (DTR), and so on. RFCOMM is used with two types of BT devices—endpoints such as printers and computers and intermediate devices such as modems. In Figure 3, the IP stack over *Point-to-Point Protocol* (PPP) over RFCOMM emulates the mode of operation over a dialup or dedicated serial link. Because the various BT devices in a piconet may offer or require a different set of services, the *Service Discovery Protocol* (SDP) is used to determine the nature of the services available on the other nodes. SDP uses a request-response packet scheme for its operation.

Bluetooth Profiles

BT includes multiple profiles that correlate to the type of services that are available from BT nodes. For example, the BT headset profile is used between an audio source and a headset, both connecting wirelessly through BT—it involves a subset of the well-known AT commands used with modems. The audio source (typically a cell phone or cordless phone) implements the BT audio gateway profile for communicating with the device implementing the headset profile. Other profiles include a basic printing profile (often used for printing between a PC and a BT-enabled printer), dialup networking profile, fax profile, cordless telephony profile, *Human Interface Device* (HID) profile, and so on. The last profile is used for BT-enabled keyboards and mice—it is based on the HID protocol defined for USB.

The Bluetooth dialup networking profile is interesting from an IP perspective; as shown in Figures 2 and 3, it involves the IP stack running over RFCOMM to provide the appearance of a serial port running PPP, which is very similar to dialup networking over a basic telephone service line.

Bluetooth Frame Format and Speeds

The frame format in BT consists of a 72-bit field for the access code (including a 4-bit preamble, 64-bit synchronization field, and 4 bits of trailer), followed by a 54-bit header field that includes information about the frame type, flow control, acknowledgement indication, sequence number, and header error check. Following the header field is the actual payload, which can be up to 2745 bits. In all, the frame length can be a maximum of 2871 bits. Whereas synchronous BT traffic has periodic reserved slots, asynchronous traffic can be carried on the other slots.

BT ranges can vary from a low-power range of 1 meter (1 mW) for Class 3 devices, 10 meters (2.5 mW) for Class 2 devices, to 100 meters (100 mW) for Class 1 devices. BT Version 1.2 offers a data rate of 1 Mbps, and BT Version 2.0 with *Enhanced Data Rate* (EDR) supports a data rate of 3 Mbps. BT Version 1.1 was ratified as the IEEE Standard 802.15.1 in 2002.

Bluetooth versus Wi-Fi

A few years ago, some marketing literature tried to emphasize BT and Wi-Fi as competing technologies. Though both operate in the ISM spectrum, they were invented for different reasons. Whereas Wi-Fi was often seen as a "wireless Ethernet," BT was initially seen purely as a cable- or wire-replacement technology. Uses such as dialup networking and wireless headsets fit right into this usage model. Recently, the discussion has focused more on coexistence instead of competition because they serve primarily different purposes. There are still some concerns related to their coexistence because they operate over the same 2.4-GHz ISM band.

To recapitulate, the Bluetooth physical layer uses FHSS with a 1-MHz-wide channel at 1600 hops/second (that is, 625 microseconds in every frequency channel). Bluetooth uses 79 different channels. Standard 802.11b/g uses DSSS with 20-MHz-wide channels—it can use any of the 11 20-MHz-wide channels across the allocated 83.5 MHz of the 2.4-GHz frequency band. Interference can occur either when the Wi-Fi receiver senses a BT signal at the same time that a Wi-Fi signal is being sent to it (this happens when the BT signal is within the 22-MHz-wide Wi-Fi channel) or when the BT receiver senses a Wi-Fi signal.

BT 1.2 has made some enhancements to enable coexistence, including *Adaptive Frequency Hopping* (AFH) and optimizations such as Extended SCO channels for voice transmission within BT. With AFH, a BT device can indicate to the other devices in its piconet about the noisy channels to avoid. Wi-Fi optimization includes techniques such as dynamic channel selection to skip those channels that BT transmitters are using. Access points skip these channels by determining which channels to operate over based on the signal strength of the interferers in the band. Adaptive fragmentation is another technique that is often used to aid optimization. Here, the level of fragmentation of the data packets is increased or reduced in the presence of interference. For example, in a noisy environment, the size of the fragment can be reduced to reduce the probability of interference.

Another way to implement coexistence is through intelligent transmit power control. If the two communicating (802.11 or Wi-Fi) devices are close to each other, they can reduce the transmit power, thus lowering the probability of interference with other transmitters.

WiBree to Low-Energy Bluetooth

WiBree is a technology first proposed by Nokia to enable low power communication over the 2.4-GHz band for button cell (or equivalent) battery-powered devices. A consequence of the low power requirement is the need for the wireless function to perform a very small set of operations when active and go back to the sleep or to standby mode when inactive.

The WiBree technology has been adapted by the *Bluetooth Special Interest Group* (SIG) as part of the lower-power BT initiative—also known as *Low Energy* (LE) BT technology. The LE standard is expected to be finalized sometime in 2009. When this standardization is completed, three types of BT devices will be available: traditional BT, LE BT, and a mixed or dual-mode BT. A mixed-mode device can operate in low power mode when communicating with other LE devices (for example, sensors) and traditional BT mode when communicating with BT devices, implying the presence of both a BT stack and an LE stack on the same device.

WiMAX

WiMAX stands for *Worldwide Interoperability for Microwave Access* and is defined under the IEEE 802.16 working group. Two standards exist for WiMAX—802.16d-2004 for fixed access, and 802.16e-2005 for mobile stations[9]. The WiMAX forum certifies systems for compatibility under these two standards and also defines network architecture for implementing WiMAX-based networks.

WiMAX can be classified as a last-mile access technology similar to DSL, with a typical range of 3 to 10 kilometers and speeds of up to 5 Mbps per user with non-line of sight coverage. WiMAX access networks can operate over licensed or unlicensed spectra in various regions or countries—though licensed spectrum implementations are more common. WiMAX operation is defined over frequencies between 2 and 66 GHz, parts of which may be unlicensed spectrum deployments in some countries. The lower frequencies can operate over longer ranges and penetrate obstacles, so initial network rollouts are in this part of the spectrum—with 2.3-, 2. 5-, and 3.5-GHz frequency bands being common. Channel sizes vary from 3.5, 5, 7, and 10 MHz for 802.16d-2004 and 5, 8.75, and 10 MHz for 802.16e-2005. WiMAX networks are often used to backhaul data from Wi-Fi access points. In fact, they are often envisaged as replacements for the current implementation of metro Wi-Fi networks that use 802.11b/g for client access and 802.11a for backhaul to connect to the other parts of the network.

Technology

The 802.6d-2004 standard uses OFDM similar to 802.16a and 802.16g, whereas 802.16e-2005 uses a technology called *Scalable Orthogonal Frequency Division Multiplexed Access* (S-OFDMA). This technology is more suited to mobile systems because it uses subcarriers that enable the mobile nodes to concentrate the power on the subcarriers with the best propagation characteristics (because a mobile environment has more dynamic variables). Likewise, the 802.16e radio and signal processing is more complex.

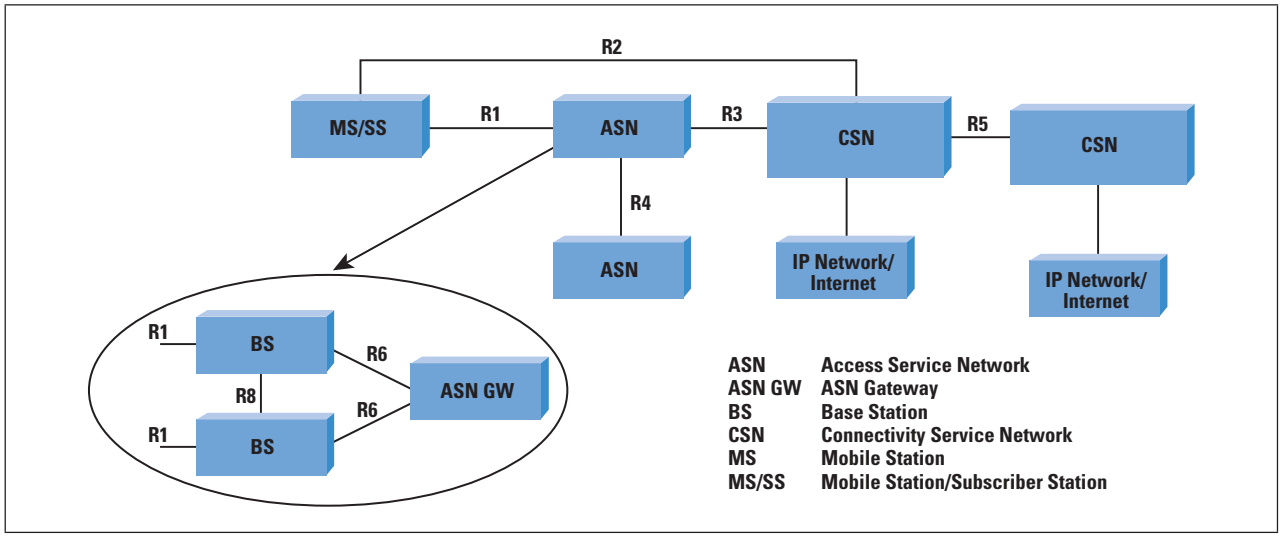
Unlike 802.11, which supports only *Time-Division Duplexing* (TDD)—where transmit and receive functions occur on the same channel but at different times), 802.16 offers TDD, *Frequency-Division Duplexing* (FDD) (transmit and receive on different frequencies, which could also be at different times). Another innovation in WiMAX is similar to the scheme in *Code Division Multiple Access* (CDMA)—subscriber stations are able to adjust their power based on the distance from the base station, unlike the case of client stations in an 802.11 network.

WiMAX base stations use a scheduling algorithm for medium access by the subscriber stations. This access is through an access slot that can be enlarged or contracted (to more or fewer slots) that is assigned to the subscriber stations. *Quality-of-Service* (QoS) parameters can be controlled through balance of the time-slot assignments among the base stations. The base-station scheduling types can be unsolicited grant service, real-time polling service, non-real time polling service, and best effort. Depending upon the time of traffic and service requested, one of these scheduling types can be used.

WiMAX Network Architecture

The WiMAX network architecture is specified through functional entities (see Figure 4), so you can combine more than one functional entity to reside on a network element. The *Mobile Station* (MS) connects the *Access Service Network* (ASN) through the R1 interface—which is based on 802.16d/e. The ASN is composed of one or more *base stations* (BSs) with one or more ASN gateways to connect to other ASNs and to the *Connectivity Service Network* (CSN). The CSN provides IP connectivity for WiMAX subscribers and performs functions such as *Authentication, Authorization, and Accounting* (AAA)[10,11], ASN-CSN tunneling, inter-CSN tunneling for roaming stations, and so on. A critical tenet of the WiMAX Forum network architecture is that the CSN must be independent of the protocols related to the radio protocols of 802.16.

Figure 4: WiMAX Forum Network Architecture Functional Blocks and Interface Points



The R3 interface (reference point) is used for the control-plane protocols and bearer traffic between the ASN and CSN for authentication, policy enforcement, and mobility management. The base station connects to an ASN gateway to provide the MS with external network access. The R6 interface between the BS and ASN-GW could be open or closed based on the profile—in fact, you could have a co-located base station and *ASN gateway* (ASN-GW), depending upon the network implementation. The ASN gateway uses the R3 interface to communicate with the AAA services in the visited CSN (that is, the CSN "corresponding" to the ASN). The servers in the visited CSN can communicate with the home CSN (that is, the CSN corresponding to the "home" network of the MS). In the simplest case multiple ASNs (WiMAX networks) connect through ASN gateways to the public Internet (that is, there is only one *Network Service Provider* (NSP) and the visited and home CSNs are the same). Note that you could implement a WiMAX network with just one ASN and one CSN—in that case, the R3 interface would be completely internal and not exposed.

Three profiles are identified to map ASN functions into ASN-GW and BS functions. These profiles are considered an implementation guideline for how you would build the various devices implementing these functions. Profile A is a strict separation of the BS and ASN-GW functions, where the ASN-GW controls and manages radio resources that are located on the BS and also provides the handover and data-path functions. The R6 interface is exposed in this profile.

Profile B is a more integrated function, where the BS has more functions than in profile A; in fact, the BS might even integrate most of the ASN functions. The R6 interface is a closed interface in this profile. The third profile is profile C, which is similar to profile A except that the base stations to incorporate more functions, including radio resource management and control as well as hand-offs.

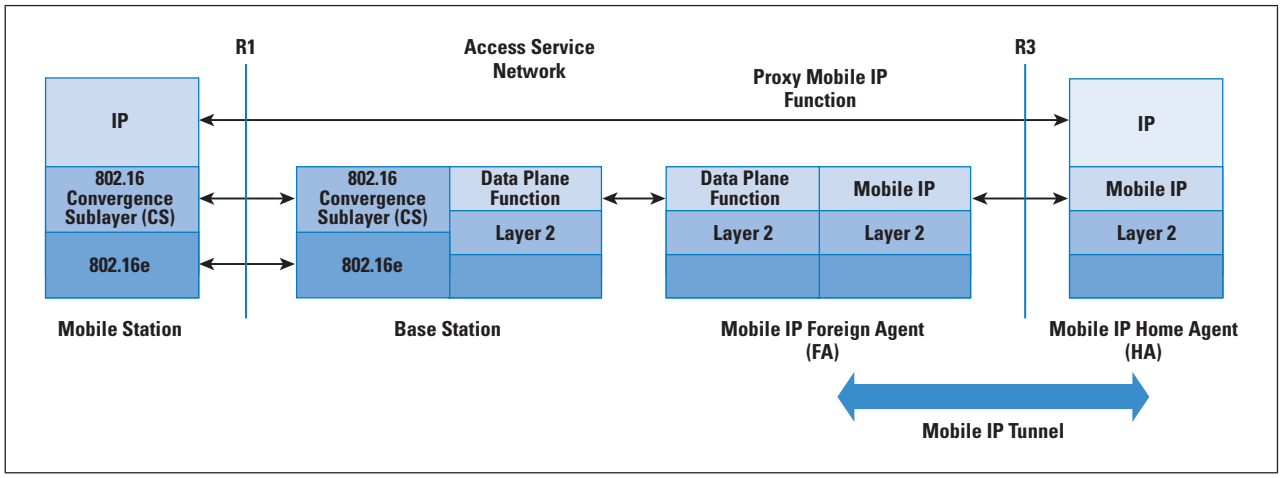
IP Connectivity and Data Transfer

The MS can be a fixed IP gateway (think of an 802.11 access point that provides connectivity to users in a coffee shop and connects to the IP network of the service provider through WiMAX) or a mobile end node (for example, a laptop with WiMAX connectivity). The IP address used by the gateway on the connection to the WiMAX network is known as the *Point of Attachment* (PoA) address. A third type of access is nomadic access, where the IP gateway can be moved from one location to another but connects to the network only after it has been relocated.

When the station is mobile, the WiMAX Forum specifies that the *Mobile IP* (MIP) architecture and protocols should be used. There are two types of Mobile IP possible: *Client Mobile IP* (CMIP) and *Proxy Mobile IP* (PMIP). The former involves changes to the MS protocol stack, but the latter does not.

The architecture can support both models. In the P-MIP scenario (see Figure 5), the ASN implements the Foreign Agent (see William Stallings' article in IPJ on Mobile IP[8]), and terminates Mobile IP tunnels for the various mobile stations in the same ASN.

Figure 5: Data Transport and Proxy Mobile IP in WiMAX



In the figure, the MS has an address at the point of attachment that is used to forward packets from the MIP Foreign Agent inside the ASN. Because the ASN acts as a proxy of the attached MS, this implementation is known as a *Proxy MIP* implementation—also, there is no need for the MS to be aware of the MIP function being performed by the network.

Perspective on WiMAX versus Cellular Services

The WiMAX Forum has specified that the *Network Working Group* (NWG) architecture should be capable of supporting voice, multimedia services, and priority services such as emergency voice calls. It also supports interfacing with interworking and media gateways. Also, the service permits more than one voice session per subscriber, as well as simultaneous voice and data sessions. Support of IP Broadcast and Multicast services over WiMAX networks is also included. The architecture is also expected to support differentiated QoS levels at a per-MS or -user level (coarse grained) and at a per-service flow (fine-grained) level. It shall also support admission control and bandwidth management.

Initially, WiMAX was touted by some as a replacement for cellular services. An important consideration was using *Voice over IP* (VoIP) for voice calls—that is, where voice was another service over the data network. This model was in contrast to the existing cellular service where data was an adjunct to the basic service of TDM-based voice. More recently, WiMAX is being positioned as a data-connectivity option for remote locations, especially where it would be difficult to lay new copper or optical cable. Not surprisingly, these options are being pursued aggressively in developing countries.

Common Misperceptions About Wi-Fi, BT, and WiMAX Technologies

We have considered the key aspects of the three technologies—Wi-Fi, BT, and WiMAX—and their position in IP networks. In this section, we will outline and clarify some common perceptions and misperceptions about these technologies.

1. *BT and Wi-Fi are competing technologies*—Actually, they address a different set of requirements despite operating in the same 2.4-GHz space. BT is a "wire replacement" usually for short distances. Wi-Fi is typically used for data, voice, and video traffic over distances up to 300 meters.
2. *WiMAX is Wi-Fi on steroids*—To clarify, this statement is an oversimplification used often in the trade press. WiMAX operates in licensed spectra and uses a different network architecture as compared to Wi-Fi, which is in the unlicensed spectrum and uses a simple access point to wired Ethernet architecture. One overlapping function is for backhauling Wi-Fi traffic, which can be done by Wi-Fi (typically 802.11a) or WiMAX.
3. *Unlike BT, Wi-Fi cannot be used for voice*—This perception is not true because you can send multimedia traffic over Wi-Fi networks implementing 802.11e QoS functions that rely on the access point and stations implementing priority-based traffic transmission and scheduling.
4. *Wireless networks are not secure*—Although there is some validity to this argument because it is easier to eavesdrop on wireless networks, implementation of security schemes such as *Wi-Fi Protected Access* (WPA/WPA2) will help alleviate this problem.
5. *Wireless and radio technologies consume more power*—This statement is often true if the devices transmit continuously or have to increase their power because of the distance between the transmitter and receiver. Noisy channels contribute to this power use also. However, with careful engineering of the wireless implementation and techniques such as power save (in Wi-Fi) and short duty cycle transmissions, the power requirement can be lowered.

Summary

In this article, we have provided a flavor for IEEE 802.11 WLAN, Bluetooth, and WiMAX technologies and their implementation—specifically, how the nodes on these networks connect to an IP network. These technologies often serve complementary functions for end-to-end connectivity.

For Further Reading

[1] IEEE 802.11 Standard, **http://standards.ieee.org/getieee802/download/802.11-2007.pdf**

[2] Edgar Danielyan, "IEEE 802.11," *The Internet Protocol Journal,* Volume 5, No. 1, March 2002.

[3] T. Sridhar, "Wireless LAN Switches—Functions and Deployment," *The Internet Protocol Journal,* Volume 9, No. 3, September 2006.

[4] IEEE 802.16-2004 IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems.

[5] IEEE 802.163-2005 IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, IEEE, **http://standards.ieee.org/getieee802/802.16.html**

[6] Bluetooth Special Interest Group Publications, **http://www.bluetooth.com/Bluetooth/Technology/**

[7] **http://www.wimaxforum.org/technology/documents/WiMAX_Forum_Network_Architecture_Stage_2-3_Rel_1v1.2.zip**

[8] William Stallings "Mobile IP," *The Internet Protocol Journal,* Volume 4, No. 2, June 2001.

[9] Jarno Pinola and Kostas Pentikousis "Mobile WiMAX," *The Internet Protocol Journal,* Volume 11, No. 2 , June 2008.

[10] Convery, S., "Network Authentication, Authorization, and Accounting – Part One: Concepts, Elements, and Approaches," *The Internet Protocol Journal,* Volume 10, No. 1, March 2007.

[11] Convery, S., "Network Authentication, Authorization, and Accounting – Part Two: Protocols, Applications, and the Future of AAA," *The Internet Protocol Journal,* Volume 10, No. 2, June 2007.

T. SRIDHAR is Vice President of Technology at Flextronics in San Jose, California. He received his BE in Electronics and Communications Engineering from the College of Engineering, Guindy, Anna University, Madras, India, and his Master of Science in Electrical and Computer Engineering from the University of Texas at Austin. He can be reached at **T.Sridhar@flextronics.com**

The End of Eternity
Part One: IPv4 Address Exhaustion and Consequences

by Niall Murphy, Google, and David Wilson, HEAnet

"Eternity is a very long time, especially towards the end," said Woody Allen[22,23], and he was mostly right. The eternity that the 32 bits of IPv4 address space promised is now almost at an end, and we are faced with the task of deciding what to do after the "end of eternity."

The size of the problem of IPv4 exhaustion is, unfortunately, also proportional to its longevity[1,2,3]. Although the next-generation (IPng) effort[4] kick-started the development of IPv6 partially in response to concern about the IPv4 consumption rate, the industry as a whole largely ignored the problem after *Classless Inter-Domain Routing* (CIDR) and the *Regional Internet Registries* (RIR) system contained the depletion problem to a manageable horizon. More recently, after Geoff Huston's[5] work showing that the expected depletion time was sooner than many organizations had expected, the concern has received considerable attention in address-allocation policy circles.

In this article, we examine IPv4 exhaustion in more detail. We talk about what exactly exhaustion will mean and what we can do about it, and then present a vision for the postexhausted world. Those familiar with our RIPE-55 talk[6] will find much that is familiar, but the arguments have been expanded for a more general audience. The authors, as in that talk, are speaking only for themselves, and not their organizations.

What Does Exhaustion Mean?

Trivially, the point of IPv4 exhaustion is the point at which the guaranteed-free-and-unused pool runs out and the current allocation mechanism comes to an end. Although the depletion of the free pool defines the technical point of exhaustion, it is not the depletion itself that is of primary importance. After all, if it were, we could simply declare a moratorium on allocations with immediate effect, to preserve the resource for some notional future requirements. Rather, it is the effect on the practices and procedures, within the RIRs and within the *Local Internet Registries* (LIRs), administrative and technical, that will practically define exhaustion. These practices, which have grown to fit around the current behavior of the addressing system, the free pool, and so on, will require urgent reform after exhaustion, as indeed will the RIR system in general.

Currently organizations use and require new addresses for essentially every IP-related additional deployment (for example, adding customers to a publicly numbered DSL service, adding extra *Secure Sockets Layer* (SSL)-enabled websites to a Web hosting service, and adding extra publicly reachable servers to almost any service).

It has been emphasized that this problem affects only the *growth* of organizations performing IP deployments[7]. Although it is important to acknowledge the partial correctness of this statement, much about the postexhaustion state could undermine the stability of well-established advertisements and routes unless the transition is well-handled. It seems intuitively correct that those who received allocations before exhaustion will be unaffected by exhaustion turmoil[8], but we regard this premise as optimistic, as you will see later.

Along those lines, one less well-examined consequence of exhaustion is the erosion of the consensus model of Internet governance. There is potential for wide divisions to open up at the local and regional level unless this consensus is carefully conserved. No clear successor to the current model as yet exists; the RIRs appeared to be heading toward a spectrum of positions on, for example, the allocation of the last portions of the IPv4 free pool[9, 10] until quite recently[24].

The erosion of this model of governance as a consequence of exhaustion has been neither widely examined nor expected in the Internet community. Partially, this situation arose because of the useful and well-executed role that the RIRs have historically filled in providing sensible and stable conditions for decision making; some proportion of the membership of the RIRs might well feel that IPv4 exhaustion is a problem like any other, which the RIRs themselves are in the perfect position to resolve. However, although the atmosphere of mutual cooperation fostered by the RIRs has produced many useful service-related outputs (for example, the *Test Traffic Measurement* service of *Réseaux IP Européens* [RIPE][25]), one of the major nonobvious benefits they have brought is to provide a centralized focus for discussion with governments and regulatory agencies. Not only is it more efficient and therefore less time-wasting to centralize through one representative organization, it has also created expectations that similar matters can be dealt with in the same coordinated way—a very valuable expectation, which has helped to increase the credibility of industry self-regulation. This credibility allowed, for example, the *Number Resource Organization* (NRO) to help forestall a proposal to allocate IPv6 according to geographical boundaries[27, 28].

Indeed, without credible industry self-regulation, it is not at all clear that this community could have grown as fast as it did. Although it seems clear today that the RIRs are the correct place for this kind of activity to go on (witness RIPE's "enhanced cooperation" task force[26]), if they had not been around, government would either have had to deal with an organization with less of a pedigree or one with more inherent bias, or multiple organizations with competing biases, all of which could compel them to distrust the results of their liaisons. Unfortunately, in this respect the RIRs have been a victim of their own success.

Just as the consensus model in domains broke down when top- and second-level domains became monetized, so it is likely that the inherent win or loss for any given holder in any policy changes will undermine attempts to build consensus for address policy in a monetized IPv4 world. Absent this consensus, many of the RIR services that we rely upon will be undermined—not least the veracity of the WHOIS database and subsequent reliability of our routing filters, but also the RIR and *Internet Corporation for Assigned Names and Numbers* (ICANN) representations toward governments.

What Are the Problems with Exhaustion?

The biggest problem is the simplest one: existing organizations whose business model or operations are *solely* predicated on an ongoing flow of IPv4 addresses will fail. This premise would seem an extreme, even theoretical, characterization, but the size of this category in the real world is larger than you might think. Numerous organizations are also in trouble, perhaps less predicated upon IPv4 than the others, but that—for example—might have financial or operational difficulty in making the postexhaustion transition happen internally. They would also be placed at risk. Finally, there are those organizations that might rely on others to perform their transition correctly in order for them to continue effective operations: less directly at risk, but still probably affected.

Those who deal with the operation of the Internet on a daily basis are well-aware of the workarounds available that could save organizations from the doomsday scenario. It is unfortunate, then, that many of us have looked to the simplest cases in our immediate experience in order to form our opinions of the scale of the problem. It is indeed true that, in the short term, the client-side problem has largely been solved—provided that your customers or developers never have expectations in line with an end-to-end Internet. (It would seem that address-space pressure is likely to erode whatever end-to-end expectations still remain in today's Internet.)

However, the server-side problem (for example, SSL Website hosting, *IP Security* [IPsec] VPN endpoints, ...) remains unsolved. Workarounds exist[11], but whether they will be ready and deployed in time remains an open question. There are, therefore, organizations operating at this moment that depend upon the continued availability of IPv4 addresses. Adequate workarounds have yet to be developed—never mind proven—for these businesses.

The situation becomes more complicated when we consider the candidate solutions. For example, such organizations as described previously cannot solve their problem by deploying IPv6 alone prior to the end of the transition, because they require universal reachability. Without universal reachability, support costs will rise, the quality of the user experience will decrease, and the credibility of Internet governance will be threatened. The only available evidence shows our position on the IPv6 transition curve being at the very beginning[12].

Therefore it is difficult to emphasize this enough—new entrants providing Internet services *cannot expect to compete equally with existing operations*—because they have a very high barrier to entry formed not by the natural action and development of competitors, but by the resource scarcity of new addresses. Without new addresses, they cannot have an IPv4 *Default-Free Zone* (DFZ) routing-table entry; without a DFZ entry, they cannot be multihomed; without multihoming, they cannot offer sufficiently redundant Internet service; and without sufficiently redundant Internet service, they cannot meaningfully compete with existing operators.

A variety of poor-quality "fudges" are possible, of course: they could use the address space of their upstream operators (and run the risk of having that address space pulled or charged for), or they could outsource any address-requiring services to another organization (and be unable to control their service quality, as well as dependent upon their continuing operation), or they could host through some kind of public proxy network that redirects to their back-end servers through various hard-coded means (and create a fragile, difficult-to-operate network with higher running costs per unit customer than their competitors).

We will examine the other negative consequences of exhaustion in more detail later in the discussion; meanwhile, let us assume that the scenario described previously is undesirable enough for us to ask whether we can actually do anything to forestall it.

Can We Practically Defer Exhaustion?

What we would ideally like is some policy or algorithm that would give us more time—how much time is open to question—without producing its own set of ill effects. (We can certainly defer exhaustion by ceasing to allocate new IPv4 addresses tomorrow, but that solution is hardly practical.) Unfortunately, this problem is very difficult to resolve. Such direct precedents that appear clearly related to the current situation provide no useful guidance. Many resource-exhaustion problems have been faced before, but ultimately the solutions for those can be categorized into three kinds:

- *Make the resource renewable:* In this case, the resource is in danger of running out, but can be replenished by some means. Often this replenishment involves constraining production predicated on the resource to some smaller value, particularly when there is a natural rate of renewal—for example, fishing stocks. In the case of IPv4, it is fundamentally nonrenewable in that the resource is of a finite size. (As we discussed previously, current reclamation efforts[13], although worthy of pursuit as a low-overhead task, cannot be a solution.)
- *Move to another resource:* This solution is already under way in the sense that we are engaged in the transition to IPv6. However, adoption of IPv6 will not happen fast enough to prevent the negative consequences of exhaustion.

- *Divide the resource more fairly:* This solution is useful primarily in the case where hoarding is taking place, causing resource problems for some significant proportion of a resource-using population. We are dividing the resource fairly as it is, and certainly since the emergence of the RIRs. For reasons discussed later, husbanding the resource more carefully is unlikely to actually be a solution.

We have faced other abstract exhaustion problems before as well: for example, phone-number depletion is somewhat similar to our current problem. However, phone-number depletion admits of a simpler solution—the creation of extra digits in the number space—because of the centralization of network knowledge in a comparatively small number of switches. For the Internet, where every deployed host would have to be informed about changes to the number space, such an approach is not operationally feasible. Furthermore, adding extra digits to the number code is not in fact simple, and telecommunications companies have experienced a wide range of problems with such approaches in the past, to say nothing of the loss of revenue and the failure of calls to connect because of customer confusion[14, 15]. We see no historical situation that provides a clear precedent and a clear way forward.

SimLIR

Accordingly, to help answer the question posed in the preceding section, we wrote a tool, *SimLIR*, to explore exhaustion and post-exhaustion scenarios. Rather than being a tool influenced primarily by computations based on growth curves, a "top-down" approach, it is a modeling tool that examines how changes in behavior affect relative consumption rates. Roughly 6,000 lines of Python, the tool is due to be open-sourced at its Google Code page[16] shortly after this article is available. The tool models the whole *Internet Assigned Numbers Authority* (IANA)—>RIR—>LIR hierarchy, and currently maps LIRs to countries; it uses the same publicly available data as Geoff's work. We would appeal to the community to help improve the program, because more research is desperately needed in this area.

Running the tool under various scenarios has produced preliminary results indicating that we cannot meaningfully defer exhaustion, given our current growth rates. It can be used to compare the effect of policy adjustments on known historical and simulated behavior. For example, one simple policy adjustment that has been informally suggested is to decrease the initial allocation size for new LIRs. Modeling this allocation with the tool, we halve the size the LIRs receive at the time of initial membership. If we allow this scenario to run to completion, we have seen that it allows us to defer exhaustion by less than a week. Intuitively, we might expect this assumption to be realistic because startup activity, although important, is relatively small in terms of proportion of allocations. New LIRs numbered approximately 500 in 2006[17], and any scheme that attempted to defer exhaustion based on such a small proportion of overall operations could not practically succeed.

The question then arises whether any other scheme based upon treating some partition of the request-space differently could have a significant positive effect. However, such a scheme necessarily assumes that some set of requests are oversized, and can in fact be shrunk with no ill effects. Even if they are oversized, identifying them without inducing either unworkable bureaucracy or a chilling effect on the operations of the organization would be a significant task, not lightly undertaken. Furthermore, it would be in the self-interest of the current RIR membership not to agree to such a change in policy. With any such scheme, there would be a non-zero chance of their own requests being deemed faulty in some respect, thus leading to significant risk to their own operations. All of this process would of course be happening in the approach to exhaustion, where it would be more critical than ever to receive enough numbering resources! We can assume, therefore, that no such scheme would ever make it past the policy-making apparatus of bottom-up-influenced RIRs. Ironically, the easiest changes to enact are changes governing allocations to startup organizations; the affected organizations are not in the room at the time of policy formation, because they are not members yet. But such changes are highly unlikely to have a positive effect.

Finally, partitioning schemes are similar to other schemes proposed to rework the *End Game* for IPv4 allocation[18, 19, 20] or retain a certain proportion of the free pool for as-yet-unknown future needs, in that we put RIRs in the awkward situation of having to decide that some requests are more legitimate than others, at a time when these requests are likely to be particularly urgent. RIRs should not be in the business of deciding who gets to have new customers, and partitioning the request space invites the possibility of preferential treatment. We can be sure that any preferential treatment at this crucial time, accidental or otherwise, would attract lawsuits. Judicial involvement in the allocation process close to the time of exhaustion would benefit almost nobody.

It is important to note that these risks are mainly specific to partitioning the request space from the RIR to the LIR; in other words, imposing criteria at the time of request. Partitioning the remaining pool per RIR, that is, imposing criteria at the time of division, such as proposed by the $n = 1$ policy[24], does not suffer from "favoritism." Indeed, even if there were blatantly iniquitous division at the IANA-to-RIR level, although various checks and balances exist to ensure there is not, it would be unlikely to affect those with resources sufficient to possess an office in the region in question, or to open one up; it is patently clear that the requests will follow where the space is, and it is highly unlikely that any single RIR with a large amount of space left after others have been exhausted would be in any kind of position to pass a discriminatory policy.

We make these points to highlight that any scheme based upon LIR partitioning presents immense difficulties of principle. Even if these difficulties are worked out, they seem unlikely to meaningfully defer exhaustion: the current run rate for IPv4 address space will exhaust the space within a 5-year timeframe anyway, even if all practically possible measures are taken.

The Consequences of Scarcity

Suppose for the moment that at the time of exhaustion, Internet-connected organizations have to fend for themselves, with no particularly well-defined industry strategy in place. We would then expect to see a broad movement within the industry to conserve precious public IPv4 address space. One obvious way for an organization to obtain more usable IPv4 space is to move previously publicly-numbered resources behind *Network Address Translation* (NAT) gateways. Other, less-legitimate sources of new addresses will probably also be explored, and these actions, combined with the generally uncoordinated changes, may well trigger the following negative consequences:

- *Inability to measure clients, and difficulty of supporting them:* As we see more layers of NAT within networks, it becomes gradually more difficult to establish who is actually connecting to you, and what problems they are having. Cookies are a partial solution for only one important protocol. Measurement becoming harder means that support costs will rise.
- *Address-space hijacking:* As organizations become more desperate for space, it is entirely feasible that they will begin to cast around for space not explicitly unavailable in order to meet their business needs. How widespread this practice would be remains an open question, but effective barriers to this behavior are not currently available. We would expect a general deterioration in the quality of routing.
- *WHOIS database quality down:* Coupled with layers of NAT hiding more and more networks from direct sight, transfers of address space (legitimate or otherwise) will cause the WHOIS database to become gradually less and less accurate, leading to...
- *Distributed denial-of-service (DDoS) tracking trouble:* Problems tracking DDoS attacks and abuse origins of all kinds make law enforcement and network operators equally unhappy.
- *Connection quality down:* Connection quality, in terms of connections that complete successfully and have tolerable latency, will go down as a function of client growth behind gateways.
- *RIR billing model under pressure:* The RIRs will need to find a new way to pay their costs or go out of business—gradually, but inevitably. Of course the RIRs, like every other organization, must serve a need, but they currently provide a large number of ancillary services not directly related to IP allocation, and those services would also be under threat.

- *Consensus undermined:* This consequence is possibly the most dangerous of them all. If a chaotic state of affairs is allowed to continue for too long, our very ability to make decisions as a community will be undermined as organizations abandon the RIR model that has failed them. We will have squandered, in a way, the foundation of trust that allows such ethical codes as we have developed in Internet operations to persist. That foundation will not be easily recovered.

(Note that all of these are effects that are likely to emerge to varying degrees with the onset of scarcity, however it takes place; in other words, if the RIRs engage in a program of scarcity management by partitioning requests, it is highly likely that the scenario described previously will happen no matter what is left in the free pool.)

In any large shock such as we describe, there will be operational turmoil. Organizations will attempt to employ the technologies they need to dig themselves out of trouble, or bend the rules to the same end. There will be financial turmoil as the ability of each business to scale in the new regime is tested. Turmoil for existing businesses and new entrants will no doubt attract increased attention from governmental and quasigovernmental agencies of all kinds. Turnover in the routing table will increase as uncoordinated deaggregation of prefixes takes place. Unwelcome as all these consequences are, we will probably be far too preoccupied with our own individual problems to take care of the broader picture.

Postexhaustion Vision

Although we hope it is clear, given the previous discussion—that IPv4 addresses will still be required after exhaustion—our highest aspiration cannot be an Internet confined in perpetuity to IPv4 alone. If we are to continue in a manner resembling our current operations, we require continued address plenty, even by today's rather restricted standards. The End Game, therefore, is an IPv6 Internet, or at least enough of one to keep off address scarcity for a workable subset of the industry.

So, the problem can then be characterized as the transition toward this state of affairs—the gap between the end of the old allocation model and the emergence of an adequate replacement. Any solution will have to either make the gap shorter, by bringing users to the IPv6 Internet sooner, or make it less painful, by helping IPv4-dependent organizations survive. (Note that a solution that makes the gap less painful may well cause it to lengthen.)

With the problem stated this way, we can evaluate possible solutions in this context. A hurried, stimulated transition of popular services to IPv6 will quite likely shorten the gap, although a mass transition is also likely to be an unstable one and so rather painful.

A voluntary release of unused addresses may help reduce the pain, but is unlikely to service the run rate adequately, given its voluntary nature, and in any event will prolong dependence on IPv4, thus lengthening the gap. Tweaking policies to make remaining IPv4 addresses arbitrarily difficult to get merely introduces the effects of scarcity still sooner, helping neither goal.

That said, our initial examination of the problems of exhaustion indicate that there will be a group of people who will require IPv4 addresses after the exhaustion point, and it is also clear that there are those who have addresses, such as the lucky recipients of class A addresses in the early days, but no particular incentive to give them up. We do not actually want to recycle these prefixes indefinitely, however; that just sustains the current model. Optimally, we should provide whatever opportunity we can to those who require IPv4 addresses, to get them (and us) toward the End Game of an adequate global IPv6 deployment.

We do not require an unlimited IPv4 supply to accomplish this goal. We do, however, require liquidity: the ability to transfer, with incentives to transfer. Although it is very difficult for a centralized system (such as an RIR) to reclaim adequate space, the effort/reward ratio is much more favorable for an individual organization that knows its own network. So we must provide some stimulus for them to increase liquidity, while imposing some realistic restriction on demand. It must of course be scrupulously fair.

Stated in this way, a market-based trading exchange is not just one way of attempting to solve the problem—such an exchange, properly regulated, is arguably the most neutral and fairest way to manage the problem of scarcity.

In the next article we will explore how such a market system should work, discuss what new problems it is likely to create, and consider the potential effect on the routing table.

References

[1] ftp://ftp.ietf.org/ietf-online-proceedings/94dec/area.and.wg.reports/ipng/ale/ale-minutes-94dec.txt

[2] http://tools.ietf.org/html/rfc2008

[3] Hain, Tony, "A Pragmatic Report on IPv4 Address Space Consumption," *The Internet Protocol Journal*, Volume 8, No. 3, September 2005

[4] http://playground.sun.com/ipv6/doc/history.html

[5] http://ipv4.potaroo.net

[6] http://www.ripe.net/ripe/meetings/ripe-55/presentations/murphy-simlir.pdf

[7] http://www.isoc.org/educpillar/resources/ipv6_faq.shtml

[8] http://www.ietf.org/internet-drafts/draft-narten-ipv6-statement-00.txt

[9] http://www.apnic.net/meetings/24/program/sigs/policy/presentations/el-nakhal-prop-051.pdf

[10] http://www.ripe.net/ripe/policies/proposals/2007-06.html

[11] http://www.switch.ch/pki/meetings/2007-01/name-based_ssl_virtualhosts.pdf

[12] For example, http://h.root-servers.org/128.63.2.53_2.html versus http://h.root-servers.org/h2_5.html

[13] http://www.ripe.net/ripe/meetings/ripe-55/presentations/vegoda-reclaiming-our.pdf

[14] A "smooth and convenient" dialing plan for India. http://www.mycoordinates.org/indias-phone-june-06

[15] http://en.wikipedia.org/wiki/UK_telephone_code_misconceptions

[16] http://code.google.com/p/simlir/

[17] http://www.ripe.net/docs/ripe-407.html#membership

[18] http://www.ripe.net/ripe/policies/proposals/2007-03.html

[19] http://www.ripe.net/ripe/policies/proposals/2007-06.html

[20] **http://www.ripe.net/ripe/policies/proposals/2007-07.html**

[21] **http://kuznets.fas.harvard.edu/~aroth/alroth.html**

[22] Woody Allen, "Side Effects," 1980.

[23] Woody Allen through (most famously) Stephen Hawking, **http://www.cnn.com/2006/WORLD/asiapcf/07/04/talkasia.hawking.script/index.html**

[24] **http://icann.org/en/announcements/proposal-ipv4-report-29nov07.htm**

[25] **http://www.ripe.net/ttm/**

[26] **http://www.ripe.net/ripe/tf/enhanced-cooperation/index.html**

[27] **http://www.nro.net/documents/nro18.html**

[28] **http://www.ripe.net/maillists/ncc-archives/im-support/2004/index.html**

[29] Huston, G., "The Changing Foundation of the Internet: Confronting IPv4 Address Exhaustion," *The Internet Protocol Journal*, Volume 11, No. 3, September 2008.

NIALL MURPHY holds a B.Sc. in Computer Science and Mathematics from University College Dublin. While in university, he founded the UCD Internet Society, which provided Internet access to approximately 5,000 students. He went on to work for (and found) various organizations: the **.IE** domain registry, Club Internet (now Magnet Entertainment), Ireland On-Line, Enigma Consulting, Bitbuzz, and Amazon.com. He is currently in Site Reliability Engineering at Google. He is the coauthor of numerous articles, some RFCs, the O'Reilly book *IPv6 Network Administration*, and is a published poet and keen amateur landscape photographer. E-mail: **niallm@avernus.net**

DAVE WILSON holds a B.Sc. in Computer Science from University College Dublin, not coincidentally from around the same time as Niall. He has worked at HEAnet, the Irish National Research & Education Network, for more than 10 years, maintaining an involvement with RIPE and with the pan-European research network Géant. Dave is a member of the ICANN Address Supporting Organization Address Council; he helped to found the Irish IPv6 task force, which has the support of the national government there. E-mail: **dave.wilson@heanet.ie**

Remembering Jon: Looking Beyond the Decade

by Vint Cerf, Google

A decade has passed since Jon Postel left us.[0] It seems timely to look back beyond that decade and to look forward beyond a decade hence. It seems ironic that a man who took special joy in natural surroundings, who hiked the Muir Trail and spent precious time in the high Sierras, was also deeply involved in that most artificial of enterprises, the Internet. As the *Internet Assigned Numbers Authority* (IANA)[1] and the *Request for Comments* (RFC) editor, Jon could hardly have chosen more polar interests. Perhaps the business of the artificial world was precisely what stimulated his interest in the natural one.

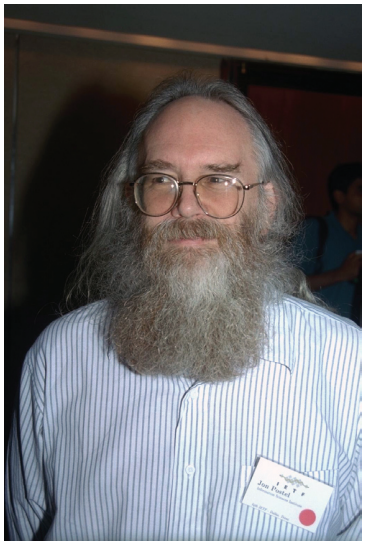


Photo: Peter Löthberg

As a graduate student at UCLA in the late 1960s, Jon was deeply involved in the ARPANET project, becoming the first custodian of the RFC note series inaugurated by Stephen D. Crocker. He also undertook to serve as the "Numbers Czar," tracking domain names, Internet addresses, and all the parameters, numeric and otherwise, that were critical to the successful functioning of the burgeoning ARPANET and, later, Internet protocols. His career took him to the east and west coasts of the United States but ultimately led him to the University of Southern California's *Information Sciences Institute* (ISI), where he joined his colleagues, Danny Cohen, Joyce K. Reynolds, Daniel Lynch, Paul Mockapetris, and Robert Braden, among many others, who were themselves to play important roles in the evolution of the Internet.

It was at ISI that Jon served longest and as the end of the 20th cen-tury approached, began to fashion an institutional home for the work he had so passionately and effectively carried out in support of the Internet. In consultation with many colleagues, but par ticularly with Joseph Sims of the Jones Day law firm and Ira Magaziner, then at the Clinton administration White House, Jon worked to design an institution to assume the IANA re-sponsibilities. Although the path to its creation was rocky, the *Internet Corporation for Assigned Names and Numbers* (ICANN)[2] was officially created in early October 1998, just two weeks before Jon's untimely death on October 16.

In 1998 an estimated 30 million computers and 70 million users were on the Internet. In the ensuing decade, the user population has grown to almost 1.5 billion and the number of servers on the Internet now exceeds 500 million (not counting episodically connected laptops, *personal digital assistants* [PDAs], and other such devices). As this decade comes to a close, the *Domain Name System* (DNS) is undergoing a major change to accommodate the use of non-Latin character sets in recognition that the world's languages are not exclusively expressible in one script[7]. A tidal wave of newly Internet-enabled devices as well as the increasing penetration of Internet access in the world's population is consuming what remains of the current IPv4 address space, accelerating the need to adopt the much larger IPv6 address space in parallel with the older one. More than three billion mobile devices are in use, roughly 15 percent of which are already Internet-enabled.

Jon would take considerable satisfaction knowing that the institution he worked hard to create has survived and contributed materially to the stability of the Internet. Not only has ICANN managed to meet the serious demands of Internet growth and importance in all aspects of society, but it has become a worked example of a new kind of international body that embraces and perhaps even defines a multi-stakeholder model of policy making. Governments, civil society, the private sector, and the technical community are accommodated in the ICANN policy development process. By no means a perfect and frictionless process, it nonetheless has managed to take decisions and adapt to the changing demands and new business developments rooted in the spread of the Internet around the globe.

Always a strong believer in the open and bottom-up style of the Internet, Jon would also be pleased to see that the management of the Internet address space has become regionalized and that five *Regional Internet Registries* (RIRs)[3] now cooperate on global policy, serving and adapting to regional needs as they evolve. He would be equally relieved to find that the loose collaboration of DNS root zone operators has withstood the test of time and the demands of a much larger Internet, showing that their commitment has served the Internet community well. Jon put this strong belief into practice as he founded and served as ex-officio trustee of the *American Registry for Internet Numbers* (ARIN)[4].

As the first individual member of the Internet Society he helped to found in 1992, Jon would certainly be pleased that it has become a primary contributor to the support of the Internet protocol standards process, as intended. The Internet Architecture Board and Internet Engineering and Research Task Forces, as well as the RFC editing functions, all receive substantial support from the Internet Society.

He might be surprised and pleased to discover that much of this support is derived from the Internet Society's creation of the *Public Interest Registry* (PIR)[5, 6] to operate the `.org` top-level domain registry. The Internet Society's scope has increased significantly as a consequence of this stable support, and it contributes to global education and training about the Internet as well as to the broad policy developments needed for effective use of this new communication infrastructure.

As a computer scientist and naturalist, Jon would also be fascinated and excited by the development of an interplanetary extension of the Internet to support manned and robotic exploration of the Solar System. In October 2008, the Jet Propulsion Laboratory began testing of an interplanetary protocol using the Deep Impact spacecraft now in eccentric orbit around the sun. This project began almost exactly 10 years ago and is reaching a major milestone as the first decade of the 21st century comes to an end.

It is probable that Jon would not agree with all the various choices and decisions that have been made regarding the Internet in the last 10 years, and it is worth remembering his philosophical view: "Be conservative in what you send and liberal in what you receive."

Of course he meant this idea in the context of detailed protocols, but it also serves as a reminder that in a multi-stakeholder world, accommodation and understanding can go a long way toward reaching consensus or, failing that, at least toleration of choices that might not be at the top of everyone's list.

No one, not even someone of Jon's vision, can predict where the Internet will be decades hence. It is certain, however, that it will evolve and that this evolution will come, in large measure, from its users. Virtually all the most interesting new applications of the Internet have come not from the providers of various Internet-based services, but from ordinary users with extraordinary ideas and the skills to experiment. That they are able to experiment is a consequence of the largely open and nondiscriminatory access to the Internet that has prevailed over the past decade. Maintaining this spirit of open access is the key to further development, and it seems a reasonable speculation that if Jon were still with us, he would be in the forefront of the Internet community in vocal and articulate support of that view.

A 10-year toast seems in order. Here's to Jonathan B. Postel, a man who went about his work diligently and humbly, who served all who wished to partake of the Internet and to contribute to it, and who did so asking nothing in return but the satisfaction of a job well done and a world open to new ideas.

References

[0] Vint Cerf, "I Remember IANA," *The Internet Protocol Journal,* Volume 1, No. 3, December 1998. Also published as RFC 2468, October 1998.

[1] **http://www.iana.org**

[2] Vint Cerf, "Looking Toward the Future," *The Internet Protocol Journal,* Volume 10, No. 4, December 2007.

[3] Daniel Karrenberg, Gerard Ross, Paul Wilson, and Leslie Nobile, "Development of the Regional Internet Registry System," *The Internet Protocol Journal,* Volume 4, No. 4, December 2001.

[4] **http://www.arin.net**

[5] **http://www.pir.org**

[6] **http://www.isoc.org**

[7] Huston, G., "Internationalizing the Domain Name System," The *Internet Protocol Journal,* Volume 11, No. 1, March 2008.

VINTON G. CERF is vice president and chief Internet evangelist for Google. Cerf served as a senior vice president of MCI from 1994 through 2005. Widely known as one of the "Fathers of the Internet," Cerf is the co-designer of the TCP/IP protocols and the architecture of the Internet. He received the U.S. National Medal of Technology in 1997 and the 2004 ACM Alan M. Turing award. In November 2005, he was awarded the Presidential Medal of Freedom. Cerf served as chairman of the board of the *Internet Corporation for Assigned Names and Numbers* (ICANN) from 2000 through 2007 and was founding president of the Internet Society. He is a Fellow of the IEEE, ACM, the American Association for the Advancement of Science, the American Academy of Arts and Sciences, the International Engineering Consortium, the Computer History Museum, and the National Academy of Engineering. He is an honorary Freeman of the City of London. Cerf holds a Bachelor of Science degree in Mathematics from Stanford University and Master of Science and Ph.D. degrees in Computer Science from UCLA. E-mail: **vint@google.com**

Letters to the Editor

IPv4 Address Exhaustion

I read with interest your article in *The Internet Protocol Journal* (Volume 11, No. 3, September 2008) regarding the IPv4 address exhaustion problem. It occurs to me that two approaches for encouraging the public and *Internet Service Provider* (ISP) community to migrate to IPv6 are being dismissed somewhat, but used creatively together might offer some hope for pushing us in that direction: government regulation and changing the fact that there isn't a public interest in IPv6.

What if government regulation forced a new or currently existing common service to use IPv6? One obvious possibility is video content. Since the broadcast industry is already regulated by the FCC, further regulation providing for governance of this type of application isn't too much of a stretch. Consumer demand is likely to increase in this area as broadband continues to be widely deployed, and if the public were required to run in dual-stack mode to access it, the likelihood of adoption would be much greater. It would also incent the ISPs to provide connectivity to the IPv6 address space, possibly even with a revenue-generating model behind it.

I reluctantly bring up the pornography industry as another type of content that could be relegated to the IPv6 address space. It is my understanding that this type of traffic as a percentage of the total is quite large. Based on this assumption, it would have the same effect of forcing the large portions of the public and ISPs to provide connectivity to the IPv6 address space. Again, I mention this industry reluctantly, but from a political perspective regulation of this industry and its content is likely to be an easier proposal for the public to support since you could use the "value" of disconnected portions of the Internet to best advantage.

I realize that the global nature of the Internet makes regulation and the subsequent enforcement extremely difficult. But, I also assume that even if our enforcement were controlled only at the perimeter of the U.S. traffic it would have a strong effect on the behavior of the public and ISPs.

Best regards,

—*John Newell, INX Inc.*
jcnewell@gmail.com

The author responds:

Thanks for your response. It is true to say that various efforts have been undertaken across many years to find a "killer-app" for IPv6, if I may be permitted to use that overabused and by now very tired term. To date these efforts have not been successful. That's not because of any lack of trying.

There have been some really quite innovative ideas for IPv6 over the years, and so far most of them have been retrofitted into IPv4 one way or another. From one perspective this retrofit is entirely logical, given that good ideas tend to thrive in locations where audiences are receptive, and today's IPv4 Internet is still a very fertile place for good ideas to flourish.

The other part of the problem is that service providers tend to create innovative services with existing markets in minds, so these days the novel applications and services that appear to gain the attention of significant parts of the user base tend to operate in the IPv4 network, and by necessity such applications and services account for *Network Address Translation* (NAT) devices and various forms of filters and firewalls.

These observations indicate that a certain reinforcing cycle exists that cements the existing role of the IPv4 Internet, and tends to work against the widespread deployment of innovative services that are feasible only in the IPv6 environment.

So if the adoption of IPv6 is a carrot or stick affair, our efforts to find some tempting carrots have, so far, not been overly successful. We've been unable to identify particular goods or services for which there is a compelling case of consumer demand coupled with a set of technology constraints that imply that the service is feasible only across a deployed IPv6 infrastructure with IPv6 endpoints. So if the field we are working in is bereft of carrots, are there any available sticks that we can use instead? In this case there is the same old stick that originally motivated iPv6 in the first place: We are running out of IPv4 addresses. If we believe that there is more to do in the Internet, more people to connect, more devices to add, more conversations to have, more services to deploy, more ideas to realize, and more objectives to achieve, then IPv4 cannot in and of itself sustain that vision for the Internet. The threat here is that the growth of the IPv4 Internet may well cease when the supply of further IPv4 addresses is exhausted.

Is this threat of network stagnation going to be enough to propel us into an IPv6 Internet? Will it be an adequate motivator to encourage the necessary investment in network infrastructure and in the provision of goods and services that first operate in a transitional dual-stack environment, and ultimately in an IPv6 world? I hope that the answers are "yes," as do many others I'm sure.

But I'm also worried that it may not be enough and that we may spin off into an entirely different trajectory that ultimately dismantles most of the attributes of today's Internet. I worry that instead of an open network that fosters innovation and creativity we might end up with "vertical integration" and "transparent convergence" and a network that actively resists new services and applications.

So for me, and I hope many others, IPv6 needs no new "killer-app." IPv6 does not need television or pornography to succeed. IPv6 is an imperative for the Internet simply because the alternatives to IPv6 appear to offer us a leap backward in technology and a leap backward in the elastic ways we've been able to use networks—and in the process we are going to destroy the Internet as we know it!

Regards,

—Geoff Huston, APNIC
gih@apnic.net

Dear Ole,

In his latest IPJ article (Volume 11, No. 3), Geoff Huston highlights the significance of NAT as a mechanism enabling service providers to externalize the costs and risks arising from IPv4 address scarcity. While acknowledging the increased burden and uncertainty borne by end users and NAT-traversing applications, Geoff speculates that the success of this mechanism is likely to inspire the deployment of yet another level of ("carrier grade") address translation, to further prolong if not absolutely preclude the incorporation of IPv6 by incumbent service providers. While entirely plausible, such a move would create the same kind of "double blind" conditions for Internet service delivery that prevailed in financial markets when debt securitization was coupled with the externalization of asset depreciation risks in the form of *Credit Default Swaps* In such cases, the second layer of indirection tends to make it all too easy to maintain self-serving assumptions (and/or plausible deniability) about the true nature and purpose of the first layer, and thus to fuel the perpetuation of unsustainable industry practices unto the point of industry collapse. Given the now inescapable lessons of the recent financial sector collapse, it would be nice if we didn't have to learn this particular one again the hard way.

—Tom Vest
tvest@eyeconomics.com

On Paper

I just received the September issue (Volume 11, No. 3) of IPJ and wanted to make a quick comment about the paper change. Upon reading the section on the change I quickly dug up the previous copy of IPJ and compared the two. I personally like the new paper much better. The main reason I like it is because it is much easier on the eyes, I think mostly because it no longer has a glare from overhead lighting reflecting like the old paper type did. It's a welcomed change from my take.

—David Swafford,
Network Engineer for CareSource, Dayton, OH, US
david@davidswafford.com

Book Reviews

A Dictionary and a Handbook

Hundreds of telecom books are published each year, but it is unusual to find a really good one. There must have been a blue moon (I'll have to check my almanac) this month, for I found two new and quite remarkable books by the same author, Ray Horak. One is a dictionary and the other an encyclopedic work, both covering the full range of voice, data, fax, video, and multimedia technologies and applications that comprise contemporary telecommunications. Further, they do so in such a plain-English, commonsense manner that you don't need to be a serious telecom student or professional to benefit from them—any layperson with a serious need to know will find them to be of great value. Finally (and this is rare in a technical book), both are actually relatively easy and certainly interesting reads, with liberal doses of fascinating historical context. In fact, they are even strong on entertainment value, with humorous observations and quotations sprinkled throughout. Horak has written each book in a different style for a different purpose, so they are best acquired together—as a set.

Webster's New World Telecom Dictionary

Webster's New World Telecom Dictionary, by Ray Horak, ISBN-10: 047177457X, ISBN-13 978-0471774570, Wiley Publishing Inc., 2007.

In order to communicate effectively in a contemporary telecom conversation, one must speak a special language rife with technical terminology, much of which is in the form of abbreviations, acronyms, contractions, initialisms and portmanteaux. To add to the confusion, many terms have multiple very precise—and occasionally imprecise—meanings, depending on the context. Writing a telecom dictionary must be a formidable task, one which only either the very brave or very foolhardy would even attempt. I'm not sure into which category Ray Horak falls, but his *Webster's New World Telecom Dictionary* is an excellent piece of work.

Organization

Dictionaries are in alphabetical order, of course, with chapters thrown in for symbols and numbers. Because the introduction of symbols requires special treatment, within each of the 28 chapters Horak organizes the approximately 4,600 definitions in ASCII order, perhaps as an accommodation for the binarians among us. The book includes an appendix of standards organizations and special interest groups, which can be useful if you need more information on a subject or need to know exactly to whom to complain about a *standard* or *specification,* both of which terms are defined clearly in the dictionary, of course.

Comparisons: Comprehensive and Correct

In my opinion, the best telecom dictionary ever written, aside from *Webster's,* is the *Communications Standard Dictionary,* by Martik H. Weik. That book unfortunately is out of print, with the final 3rd edition dated 1996. At 1095 pages, it is a bit overwritten and way too technical for most purposes, reading much like an IEEE dictionary. At this point, it certainly is out-of-date.

A handful of other telecom dictionaries and encyclopedias are currently in print, by far the most popular of which is *Newton's Telecom Dictionary.* Because *Newton's* dominates the market and has done so for many years, any telecom dictionary or encyclopedia is inevitably compared to that work. *Webster's New World Telecom Dictionary* is no exception, particularly because Ray Horak was the contributing editor to Newton's from the 12th through the 22nd editions.

Although *Webster's* defines only 4,600 terms in comparison to Newton's highly dubious claim of some 24,500 terms, *Webster's* definitions are much better researched, much more precise, and much more efficiently worded (that is, there is much less "fluff"). Even if *Webster's* almost certainly will gain in bulk as future editions expand the coverage of the telecom domain, it contains all of the essential telecom and IT terms, and defines them clearly and concisely. *Webster's* includes many humorous definitions but, unlike *Newton's,* they are all relevant and meaningful. For example, Horak lists three types of standards—*de jure, de facto,* and *du jour.* According to him, a *du jour* standard is defined as follows:

> "From French, meaning *of the day.* The popular standard of the day. One day 10 years ago, ATM was really hot and a lot of people made a lot of money talking about ATM and selling products based on ATM. It seemed like only the next day that IP was really cool. (I made this one up.)"

Other humorous definitions include analogue, endianess, Hellenologophobia, hoot 'n' holler, OCD, PC, and WMBTOTCITB-WTNTALI. All of these, and more, serve to lighten the load, so to speak, but none of this humor detracts from what is a serious book on a serious subject. *Newton's,* on the other hand, is so full of personal observations and anecdotes, irrelevant humor (?), and inaccurate definitions as to make you wonder why bother to make the comparison at all. Horak states that he wrote *Webster's* partly to atone for his sins in contributing to *Newton's,* but mostly to put an authoritative reference book in his own hands, and those of others involved in litigation support. He apparently does a fair amount of work as an expert witness in intellectual property (the other IP) cases and on innumerable occasions has been asked to define and opine on terms such as link, circuit, channel, call, connection, switch, router, and PSTN. Now he can testify in court with one hand on the Good Book and the other on *Webster's*.

Recommended

Webster's New World Telecom Dictionary is an excellent piece of work. Ray Horak and his technical editor, Bill Flanagan, have collaborated to create a well-written, authoritative work that clearly sets a new standard for telecom dictionaries. I highly recommend it to anyone serious about telecom.

Telecommunications and Data Communications Handbook

Telecommunications and Data Communications Handbook, by Ray Horak, ISBN-10: 0470041412, ISBN-13: 978-0470041413, John Wiley & Sons, 2007.

Unless you have really big hands, you may wonder how it is that a tome of 791 pages that weighs more than 3 pounds could possibly be called a handbook. Well, the term "handbook" actually is fairly imprecise, but Ray Horak's *Telecommunications and Data Communications Handbook* certainly is not. Actually, it is about as compact as it can be, given its encyclopedic nature, and it is very precise, indeed. The book covers the entire telecom landscape, from wireline to wireless, from copper to radio and fiber, from electrical to optical, and from the customer premises to the cloud. It discusses voice, data, fax, video and multimedia technologies, systems, and applications in great detail, and in the LAN, MAN, and WAN domains. The handbook explores every relevant technology, standard, and application in the telecom and datacom space.

Horak is a well-known telecom consultant, author, writer, columnist, and lecturer. The *Telecommunications and Data Communications Handbook* is based on his best-selling *Communications Systems and Networks* (1997, 2000, 2002), but is considerably more technical and broader in scope. It is exceptionally well-written in Horak's plain-English, commonsense style, making it just as helpful to the neophyte and layperson as to the serious student or seasoned IT professional. Horak makes liberal use of well-constructed graphics to illustrate system and network architectures, topologies, and applications.

Organization

The Handbook begins with an excellent table of contents (20 pages) and ends with an excellent index (29 pages), both of which are crucial to a good book. After all, it doesn't make any difference how good the information is if you can't find it. The book is logically organized into 15 chapters and 2 appendixes.

Chapter 1 is devoted to fundamental concepts and definitions, thereby building a firm foundation of concepts and terminology upon which subsequent chapters build. Terms such as two-wire, four-wire, circuit, link, channel, switch, and router are clearly defined, compared, and contrasted. Chapter 2 explores the full range of transmission systems, including twisted pair (UTP, STP, and ScTP), coaxial, microwave, satellite, *Free Space Optics* (FSO), fiber-optics, *powerline carrier* (PLC), and hybrid systems.

Chapter 3 examines voice communications systems: KTS, PBX, Centrex, and ACD. Chapter 4 discusses messaging systems in detail, including facsimile (fax), voice processing, and e-mail and instant messaging, concluding with a detailed discussion of unified messaging and unified communications. Chapter 5 is dedicated to the *Public Switched Telephone Network* (PSTN) and addresses *Numbering Plan Administration* (NPA), regulatory domains, rates and tariffs, signaling and control systems, and network services. Chapter 6 returns to fundamentals, this time in the data communications domain, with detailed explanations of *Data Communications Equipment* (DCE) such as modems, codecs, CSUs, and DCUs, and then moves on to protocol basics, code sets, data formats, error control, compression techniques, network architectures, and security mechanisms.

Chapter 7 deals with conventional digital and data networks such as DDS, Switched 56, VPNs, T/E-carrier, X.25, and ISDN. Chapter 8 treats *Local-Area Networks* (LANs) and *Storage Area Networks* (SANs) exhaustively, including transmission media, topologies, broadband vs. baseband, equipment, operating systems, and standards. This chapter covers 802.3, 802.11, HiperLAN, Bluetooth, IEEE 1394, Fibre Channel, and iSCSI in considerable detail. Chapter 9 is devoted to broadband network infrastructure, including both access technologies (for example, xDSL, CATV, WLL, PON, and BPL) and transport technologies (for example, SONET/SDH and RPR). Chapter 10 offers an exhaustive study of broadband network services, including Frame Relay, ATM, Metropolitan Ethernet, B-ISDN, and AINs.

Chapter 11 discusses wireless, with an emphasis on mobility, covering both broad concepts and technical specifics of *Specialized Mobile Radio* (SMR), paging, cellular (1G, 2G, 2.5G, 3G, and beyond), packet data radio networks, and mobile satellite networks (GEOs, MEOs, and LEOs). Chapter 12 thoroughly treats video and multimedia networking, including a detailed discussion of video and multimedia standards (for example JPEG, MPEG, and H.320), *Session Initiation Protocol* (SIP), and IPTV. Chapter 13 exhaustively and insightfully explores the Internet and *World Wide Web* (WWW), including a thorough discussion of the IP protocol suite. Chapter 14 briefly examines convergence, and Chapter 15 examines telecom regulation, with a focus on the United States.

Appendix A is something of a decoder for abbreviations, acronyms, contractions, initialisms, and symbols. Appendix B gives a complete listing of relevant standards organizations and special interest groups, including full contact information, in case you need more information or want to offer comments on a particular subject.

Comparisons

It is hard to make a valid direct comparison to this book. *The Irwin Handbook of Telecommunications,* by James Harry Green, is good, but less complete, less technical, and drier, if such a combination is possible. The most recently published 5th edition also is apparently out of print. The *Voice & Data Communications Handbook,* by Regis "Bud" Bates, is written at a lower level; and, the *Essential Guide to Telecommunications,* by Annabel Dodd, at a much lower level. These latter two books are breezy reads and appeal more to a mass market than to a serious student or professional.

The *Telecommunications and Data Communications Handbook* compares more correctly to some of the more seminal works of Gilbert Held or James Martin, but covers a much wider range of subject matter and is a much easier and more pleasant read.

Recommended

The *Telecommunications and Data Communications Handbook* is written for the academic and professional community, but is just as relevant to anyone who needs to understand telecommunications system and network technologies and their meaningful applications. It is an exceptional work that should be on every IT professional's bookshelf…when not in his or her hands.

—John R. Vacca,
jvacca@frognet.net

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

Itojun Service Award Launched

A new award, providing recognition and support for those progressing IPv6 development on the Internet, was announced in November. The *Itojun Service Award* honors the memory of Dr. Jun-ichiro "Itojun" Hagino, who passed away in 2007, aged just 37[1]. The award, established by the friends of Itojun and administered by the *Internet Society* (ISOC), recognizes and commemorates the extraordinary dedication exercised by Itojun over the course of IPv6 development. Itojun worked as a Senior Researcher at the *Internet Initiative Japan* (IIJ), was a member of the board of the *Widely Integrated Distributed Environment* (WIDE) Project, and from 1998 to 2006 served on the groundbreaking KAME project in Japan as the "IPv6 Samurai." He was also a member of the *Internet Architecture Board* (IAB) from 2003 to 2005.

At the time of his passing, Russ Housley, *Internet Engineering Task Force* (IETF) Chair, and Olaf Kolkman, IAB Chair, issued a joint statement, praising Itojun's service to IPv6 developments, saying that he had "inspired many and will be missed."

The Itojun Service Award will run for 10 years, presented annually to an individual who has made outstanding contributions in service to the IPv6 community. The award includes a presentation crystal, a US$3,000 honorarium, and a travel grant. The Award will honor an individual who has provided sustained and substantial technical contributions, service to the community, and leadership. With respect to leadership, the selection committee will place particular emphasis on candidates who have supported and enabled others in addition to their own specific actions.

The selection committee members for the Itojun Service Award are: Jun Murai, Hiroshi Esaki, Ole Jacobsen, Bob Hinden, Randy Bush, Bill Manning, Tatuya Jinmei, Kazu Yamamoto, and Kenjiro Cho.

Memorial donations to the Itojun Service Award Fund are welcomed and the Internet Society has established an account for donations. Details of the fund, as well as more information about Jun-ichiro "Itojun" Hagino and the Itojun Service Award are available on the ISOC Web site: `http://www.isoc.org/awards/itojun/`

The WIDE Project has also established a Japanese bank account to collect donations in Japanese Yen, the details of which are available here: `http://www.wide.ad.jp/itojun-award`

[1] Hinden, Bob, "Remembering Itojun: The IPv6 Samurai," *The Internet Protocol Journal,* Volume 10, No. 4, December 2007.

EsLaRed Receives 10th Annual Postel Service Award

ISOC awarded the *Jonathan B. Postel Service Award* for 2008 to *La Fundación Escuela Latinoamericana de Redes* (EsLaRed) of Venezuela for its significant contributions to promote information technologies in Latin America and the Caribbean.

It is now ten years since the passing of Internet pioneer Jonathan B. Postel, the inspiration for this prestigious award. To mark this event in a special way, ISOC formed a *10th Anniversary Award Committee* including all the past award recipients, which has formally recognised EsLaRed for "its sustained efforts to bring scientific, technical, and social progress in Latin America and the Caribbean through education, research, and development activities on technology transfer."

ISOC presented the award, including a US$20,000 honorarium and a crystal engraved globe, in November during the 73th meeting of the IETF in Minneapolis, USA.

Accepting the award for EsLaRed was its President, Professor Ermanno Pietrosemoli. "We're very excited to be honored in this way," said Professor Pietrosemoli. "In the developing world, having access to the Internet, which gives us access to things like scientific journals and medical information, is not easy and it is not taken for granted. It is wonderful for us to be able to help people improve their conditions and to see first hand how the Internet can change people's lives," he said.

"On behalf of the ISOC community, it is my great pleasure to congratulate Professor Pietrosemoli and his dedicated colleagues at EsLaRed for their achievements over the years," said Lynn St. Amour, President and CEO of ISOC. "EsLaRed's commitment to the Internet has been at the forefront of regional development and their leadership has been an instrumental element in forming today's dynamic Latin American and Caribbean Internet community," said Ms St. Amour. For more information about this year's recipient see:
`http://www.isoc.org/awards/postel/eslared.shtml`

The Postel Service Award was established by ISOC to honor individuals or organisations that, like Jon Postel, have made outstanding contributions in service to the data communications community. The award is focused on sustained and substantial technical contributions, service to the community, and leadership. Previous recipients of the Postel Award include Jon himself (posthumously and accepted by his mother), Scott Bradner, Daniel Karrenberg, Stephen Wolff, Peter Kirstein, Phill Gross, Jun Murai, Bob Braden and Joyce K. Reynolds (jointly), and Nii Quaynor. The award consists of an engraved crystal globe and a US$20,000 honorarium. For more information see:
`http://www.isoc.org/awards/postel/`

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal

March 2009 Volume 12, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

IP Version 4 address exhaustion and migration to IP Version 6 continues to be the focus of many Internet-related organizations and events. The *Regional Internet Registries* (RIRs), still debating what will happen as the IPv4 address pool runs out, are developing policies for how to manage address-block transfers between address holders. One potential result of the address shortage is that a *market* (official or otherwise) will develop for the buying and selling of IPv4 addresses. In our last issue, we brought you the first in a two-part series of articles entitled "The End of Eternity," by Niall Murphy and David Wilson. Part Two, included in this issue, discusses what a market-based IP trading exchange might look like.

IP address allocation, transfers, and even the potential trading market for addresses is ultimately dependent on a reliable and trusted registry for this information. The RIRs have been working on a way to ensure that information about *IP Number Resources* (that is, IPv4 addresses, IPv6 addresses, and *Autonomous System* [AS] numbers) are securely stored and distributed so that users of such information can be assured that it is authentic. The underlying technology is a *Resource Certificate Public Key Infrastructure* (RPKI), and it is described in our second article by Geoff Huston.

The Internet technical community is discussing the so-called *identifier/locator split* as a major change to the Internet architecture. The IETF is developing several proposals, including the *Locator Identifier Separation Protocol* (LISP) discussed in our March 2008 issue. In this issue we look at another proposal, the *Host Identity Protocol* (HIP). The article is by Andrei Gurtov, Miika Komu, and Robert Moskowitz.

You will notice that our back cover has a new look. This layout is not the result of any creative design urges, but rather a change in U.S. Postal Service regulations regarding the placement of the subscriber address label. I guess the Internet isn't the only place where addressing is a major topic.

As always, your comments, suggestions, and contributions are welcome, including Letters to the Editor, Book Reviews, and of course full-length articles. Our Call for Papers is included on page 35. Contact us by e-mail at ipj@cisco.com

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

The End of Eternity
Part Two: Address Space Trading and the Routing Table

by Niall Murphy, Google, and David Wilson, HEAnet

In our last article[0], we wrote about the onset of scarcity and the problems that are likely to ensue as a result. We characterized the problem we face as the *gap*, the length of time between the end of IPv4 plenty and the beginning of a universally reachable IPv6 Internet. Noting that any solution should either make the gap shorter, by bringing forward full IPv6 deployment, or make it less painful, by reducing the pressure of IPv4 scarcity, we propose that the fairest, most neutral way to encourage networks out of IPv4 while providing help for those who need it is to introduce a market-based IP address trading exchange. Let us explore now how such a system could work.

Possible Market Structures: Advantages and Drawbacks

An exchange could be set up and operated in many ways. Our preference, however, is for such a service to be run by the existing, trusted, and stable *Regional Internet Registries* (RIRs). Not only are they experienced in maintaining the values that the community as a whole wants to see maintained—fairness and neutrality, transparency, etc.—the RIRs are also in an excellent position to establish the *quality* of prefixes traded in an exchange, having excellent service contracts and history with members. Furthermore, the RIRs are unlikely to be made available for onward sale or transfer to other organizations with "different values," and would maintain their traditionally community-focused policy-making apparatus. They would also be in a position to act quickly to coordinate and assume responsibility if given sufficient authority by the membership.

It does not have to be an RIR, of course: we *could* set up another industry body, but it would take valuable time and require a new governance model. We could also outsource the whole thing to any professionally run auction-handling site, but for such a fundamental change in how we do things, it seems wise to keep it under direct control. Finally, the psychology of continuity is important; if organizations are used to dealing with the RIRs, it provides an important perception of stability to keep them as the interface to getting new addresses.

As with our previous article, we emphasize again that the RIRs have provided excellent service in focusing the consensus of the community in a form that can be passed back to governments and other stakeholders, both external and internal.

The shield provided by the RIRs, protecting the members from the outside and protecting the members from themselves, has worked well for three reasons:

- First, RIR consensus is widely seen to broadly reflect the wishes of their communities as a whole because of the extremely low barrier to representation—in essence anyone who cares can attempt to influence policy, and no formal attempt is made to weigh one set of opinions over another. As a result, RIR policy is a lowest common denominator that is in general free from many of the more partisan stances usually found in the telecommunications arena, leading to greater credibility outside the RIR system, and greater credibility within, because the oppression of a minority by the majority within the context of policy formation is very difficult.
- Secondly, possessing that credibility has led to repeated success for the RIRs in the arena of disseminating and explaining policies outward, and they have therefore reinforced the confidence their members have in them.
- Finally, the RIRs are also comparatively financially easy to run; in the *Réseaux IP Européens* (RIPE) region, fees are by no means excessive given the ratio of customers to addresses; they are observed and validated by RIPE *Network Coordination Centre* (NCC) members, and any competing industry body would have to duplicate not only all the previously mentioned activities, but also the large working surplus that allows the RIRs to ensure stability through more turbulent times. Or to put it another way, "it's open, it works, and it's cheap." We would recommend that any significant extension to the RIR authority, such as running an exchange as proposed, should endeavour to preserve as many of these properties as possible.

So if RIRs are to be the point of contact and policy making, how might such an exchange operate? We have a few guidelines from a relatively new field of economics, called *Market Design Theory*[21], that might help to inform our choices. Firstly, we must have *thickness*: we must have enough traders (both buyers and sellers) entering the market, such that the populace at large can be assured that if they need to perform a transaction, the exchange is the place to do it, rather than private trades. (Private trades, although they enable liquidity, have the disadvantages that the WHOIS database is not maintained, that policy cannot be centralized, that prefix de-aggregation can occur arbitrarily, and so on.) We should avoid *congestion:* so many participants that it becomes difficult to trade. Finally, we must have *safety:* the assurance that if a transaction is engaged in, it will complete, and buyers will receive what they want.

Although other properties exist, those are the main ones required for the exchange to operate successfully. On thickness, we think it is clear that attracting buyers in a time of scarcity will not be a problem. The problem will be attracting sellers from such constituencies as have them available (old *Internet Assigned Numbers Authority* [IANA]-allocation holders, dot-com failures, and so on). An open question is whether the exchange can do more to attract sellers than the monetary reward for selling would do on its own; more meaningful incentives for them are difficult to determine. Overall, congestion does not seem likely to be a concern, given that the RIR model most usefully supports only membership-based participation initially. (Furthermore, our guess is that the "product" will be quite homogenous, so performing trades will presumably be mostly a matter of determining price.)

Let us return to the question of prefix *quality*. The single most important measure of quality of a prefix, the attribute without which the prefix is useless, is *uniqueness*. One must be assured that the prefix one holds is acknowledged as being held by oneself, and that *Internet Service Providers* (ISPs) will accept its announcement from *no other* parties.

From a plentiful pool, where prefixes have no cost other than the service charge of the registry, ensuring uniqueness is perhaps not a simple task, but it is a relatively uncontroversial one. When scarce, prefixes become valuable and will be given a cash value, either officially or by other means. ISPs will then have a business reason to break with consensus on routing filters, as we discuss later in more detail; but regardless, prefixes allocated from the IANA free pool generally have an impeccable heritage and do not vary greatly in usability. There are, of course, the natural delays in having new /8s incorporated into routing filters across the world. Those delays do have real effects, but the recipient of these prefixes usually has good reason to believe that a) these problems will be corrected over time, and b) everyone else in the same /8 will have the same problem.

In the new paradigm, each prefix must be carefully examined by the recipient to test that it is uniquely held by the proffering organization, and the recipient will presumably have a further interest in its routability and membership in blacklists. The quality problem arises in both private and public trades; if the RIRs implemented a quality test, that would be yet another advantage of centralization to the benefit of everyone.

Closely associated with prefix quality is the question of *safety*. Again the RIRs are in an excellent position to provide the necessary support for good-faith transactions, certification of prefixes being the primary mechanism, although various other possibilities (such as membership controls) might also exist.

More pertinently, pricing of the goods traded in such an exchange is an important question. Various natural calculations might support the calculation of address costs, including but not limited to average revenue per address, operational costs averaged over all addresses held, and so on. Our primary contention here is that the RIRs should not engage in price setting directly. Doing so would at the very least invite regulation. There may be a case for placing caps on trades as an antispeculation measure, but that requires further analysis.

What exactly the "goods" are in this case also needs consideration. Our preference is that what is traded is the right to use a prefix, rather than a prefix itself. Quite apart from the inherent oddness in selling a 32-bit integer (with 5-bit netmask), we should avoid the land registry model, where all the previous history of a prefix must be checked before sale. We need the RIR to intermediate itself and provide quality evaluation services rather than leaving it up to the end buyer. We should also not be selling rights to use prefixes of fixed sizes. The exchange needs to offer a spread of lengths in order to meet the needs of all potential customers.

You Say You Want a Revolution

To be sure, a change in the perceptual or legal status of IP addresses is a revolution in how we do things. The ramifications of IP addresses becoming property, or even acquiring intermediate states with property-like title rights, are manifold and they involve sweeping changes. Suddenly things that had no value have a clear public worth. Will organizations then be compelled to list addresses on their books as an asset? Could they then be taxed on them? What would such a tax rate be? Could organizations not actually using the asset (say, the RIRs) avoid this charge? Would transfers entail a taxable operation? These questions are significant and difficult. The right thing for the community is almost undoubtedly that IP addresses do not become simple property, but rather have (at a minimum) transfer and sale rights associated with them. In this way we could enable liquidity without complications, and avoid introducing extra complications at a difficult time. But it is unclear whether regulatory authorities will see it this way without the correct guidance.

The change in legal status of IP addresses is not the only violent change that could be unleashed by exhaustion. Consider, for example, the potential for litigation led by both new entrants unable to acquire an allocation to fulfill their business plan and incumbents seeking to either cause confusion (as an anticompetitive measure against just about anyone) or to try to disrupt any fragile consensus about how the last allocations play out. Leaving aside the question of whether simple prudence would recommend or deprecate such a move, there is a very clear risk of attempted litigation affecting the outcome of the end game.

However, one of the major benefits of a market is that it allows the RIRs to maintain a hands-off approach while still making it at least theoretically possible for an organization to get an independent allocation. The community can be doing all that it realistically can to continue the flow of IPv4, in terms of creating conditions fostering its dissemination, while being seen to be doing such, rather than simply running out of ideas and giving up. It could, of course, be seen—not unfairly—that participating in the transition to a market mechanism might amount to the effective transference of title to those who happened to be in the room at the time of exhaustion, an effective "insider privatization."

Yet, if a market does not emerge, it is hard to see how any new entrants can have a business plan not directly dependent on incumbents. Although there are plenty of incumbents who would value having more address space to continue their business over the cash value of their addresses, so rendering entrance to the market impossible, there are plenty of other organizations that have only ever used a portion of their first allocation and would theoretically be well motivated to disburse these addresses accordingly.

To avoid exceptional attention from regulatory authorities, and to prevent the exchange from failing, we should design the exchange to deter in a systematic way the misbehavior of markets: speculation, hoarding, cartels, price fixing, and regional disadvantage should all be made as difficult as possible within the context of running a limited-membership market.

If we define *speculation* as short-term dealing with no expectation of use, we may be able to limit this kind of behavior naturally as a consequence of the membership-based participation inherent in the RIR model, and as a function of the periodic nature of routing filter generation. Increasing the price with short-term speculation disincentivizes the end purchaser with a use expectation from actually buying the prefix, because there will be a time delay before it can be used; therefore the purchaser with no use expectation will find it more difficult to find a buyer if the price rises to unreasonably high levels.

Hoarding, defined as long-term speculation with no use expectation, is bad for the exchange in that thickness is reduced, but also bad for the hoarder because the long-term value of the asset should decrease, in line with the increase in deployment of IPv6.

The formation of *cartels* would actually be quite a practical difficulty, especially under the closer attention likely to be paid to the exchange by competition authorities. Notwithstanding the coordination difficulties, we are inclined to say again that enough buyers should help to control this problem sufficiently to make the exchange work.

Regional disadvantage is, however we look at this situation, a problem. If scarcity is likely to lead to some monetary value being placed on address space, we face a vista where regional disadvantage can only be reduced, not eliminated. The inequality is, ultimately, one of the most compelling reasons to minimize the length of the transition period, and it would benefit us all to do so. Some measures go part way toward alleviating the problem. For instance, regional cooperation can help—in a market, if buyers cooperate and bulk buy, the threshold for organizations that would otherwise be facing a prohibitive barrier to entry would be reduced.

If we do not have a globally accessible exchange, it does not necessarily mean that the organizations will simply fail, entrenching the regional inequality, but they may respond by trying to fulfill their customer requirements by means of private, uncoordinated trading, with all the problems that entails.

We note that it is probably best to structure the actual trades as *auctions,* rather than facilitated marketplace transactions. When quality is asserted, one prefix is much like another—at least compared to prefixes of a similar size—and treating them as a commodity in this way facilitates the enforcement of policies on a centralized basis.

Drawbacks of a Market

Many cautionary tales about the operation of markets exist. Irrational exuberance, long-lasting depressions, fraudulent or exploitative behavior of all kinds—all of these effects, either enabled or supported by market mechanisms, are well known. Do we have any reason to believe either that these consequences will be not serious in our particular domain or that we have any new way of preventing them from happening?

In truth, we have no particular reason to believe that they won't happen, but there is a structural reason to believe that they might not matter to the exclusion of all else: the worse the situation becomes in the IPv4 marketplace, the more incentive there is to move to IPv6. To that extent, the market might be considered as providing a somewhat self-regulating reason for transition. Of course, we can put various mechanisms in place to help mitigate unstable behavior, as we suggested previously, but ultimately this is a fundamentally new way of doing things that we are ill equipped to understand the full consequences of.

Perhaps the largest drawback, outside of the practical difficulties in getting IPv4 addresses to organizations, is the philosophical impediments that come inherent with switching to a market-based model for allocation. Although a market cannot be said to rule out the consensus model that has turned out well for the Internet community, it also cannot be said to fully support it. This change may be a cultural one we find difficult to reverse, and it might undermine any future attempt by the community to try to differentiate itself on governance model.

Even though we have proposed the market model in good faith, as an attempt to meet the needs of new entrants and existing organizations—and as a boost to the faster deployment of IPv6—if it proves to be a failure in meeting those needs, there may be no more credible strategies left if governments insist on action. That in itself might represent even larger, more unpredictable change for the industry.

Effects on the Routing Table

Another inescapably important question is what will happen to the *Default Free Zone* (DFZ) routing table. A world in which address blocks transfer without the aggregating procedures of the RIRs is naturally a cause for concern, and when needs-based allocation comes to an end, a change in the rate of growth does seem inevitable. We can, however, make some observations that might reassure us, to some extent, that the rate of growth will not be calamitous.

First, as we go from a time of address plenty to address scarcity, one can assume that the ongoing fulfilled demand for address space will be no greater than it is now. Hence, the future growth in the number of prefixes in the routing table—regardless of prefix length—would seem to have an upper limit consistent with the number of allocations by RIRs to *Local Internet Registries* (LIRs) at the moment. This limit is still a multiple of the current curve, because we lose the benefit of the aggregation function performed by LIRs, but it suggests that we will at least not face an order-of-magnitude step change as a result of a disorderly competition.

Then there is the question of the routability of smaller prefixes. There is, at the moment, a *de facto* longest prefix size of around /24 that has close to universal reachability on the general Internet. One might assume that this prefix size will grow inexorably during and after exhaustion, as existing space is broken up into smaller and smaller blocks. Implicit in that assumption is the notion that such block sizes will be adequate for users and worthwhile for ISPs to route; we should probably not rely on networks "making do" with smaller and smaller chunks of address space.

Simultaneously, inexorably growing prefix lengths in the DFZ can only come about because of operator action. In particular, although there is a rough consensus in DFZ operators at the moment that /24 is routable and /25 is not, this policy is not a consensus-approved policy of the RIRs or the IETF. Each operator makes its own decision, based on its own customer needs, its own network, and the expectation of routability with other networks.

Reachability, therefore, depends on ISPs cooperating, and universal reachability depends on ISPs cooperating universally. An ISP may well choose to carry smaller prefixes on behalf of its customers, but unless this practice becomes widespread, no expectation can be made of universal reachability, and the practice will remain a minority one conducted by cooperating ISPs, as occasionally happens from time to time today, and this situation will little affect the size of the routing table for those involved.

Is there a competitive advantage to the largest of the ISPs in investing in very large routers that can carry many millions of prefixes, more than the smaller ISPs can support? If there were, it could perhaps lead to a concentration of power in the tier-one providers (who, as inevitable parts of any lengthy path across the Internet, have the greatest influence on the *de facto* longest routable prefix.) This situation could perhaps be true if routers are price-limited by the supportable number of prefixes, but this characteristic is typically a secondary one at worst. Routers are grouped by the bandwidth they can support, and priced accordingly; a 100-Mbps router that can support a million prefixes will certainly be more expensive than a 100-Mbps router that can support only ten thousand, but there is an order of magnitude step from either router to a router that can support 10 Gbps.

Inaction Leads to Harm

In fact the argument that the effect on the routing table will be unsustainable is opposed to the argument that there may not be adequate liquidity to sustain the market. It is true that we could find ourselves in the latter position, and so the effect of this system on reducing the problem (characterized as "the gap") will be smaller than we might like—but, as a best-effort scenario, not negligible, particularly in regard to showing good stewardship of the resource to potential outside influence. Compared to any other proposal, and particularly compared to voluntary release or a locking down of the address space, we think that this way is the best way to assure that we make available what liquidity there is.

It is difficult to see any model—even an idealized one—that could possibly service the run rate while maintaining aggregatability. The sparse allocation model used by the RIRs is dependent upon the continued availability of large, clean blocks of space, that is, /8s from IANA. With this address plenty comes freedom in our choice of policies, and with that freedom comes relatively quick consensus.

Post-exhaustion, the space will not be plentiful, and regardless of whether a monetary cost is attached, it will no longer be free. At this point, the legitimacy of the consensus of the RIR fora becomes critical. It is a fiercely defended bottom-up process. As the legitimacy of policies in the *Domain Name System* (DNS) world comes from consensus to abide by a single `root.cache`, so the legitimacy of policies in routing comes from general agreement on route filters and the authenticity of data in the RIR WHOIS databases.

We have also learned from the DNS world what happens to operational consensus when the resource becomes in some way valuable. Although the current RIR meetings are able to come to decisions that roughly reflect the consensus of the operational Internet, the necessarily tougher decisions forced upon us will challenge those who participate directly in policy making to reach conclusions that will satisfy operators who are not present. In principle it should not be necessary to account for those who do not represent themselves, but when the legitimacy of our policies is derived from their operational choices, the burden rests on us to ensure that our processes are truly representative.

If we are unsuccessful in doing so, or indeed if we choose to maintain the status quo, we cannot assume that the policies implemented on the operational Internet will themselves remain static. It is already the case that ISPs will work together, as is their entitlement, to agree to route prefixes for the benefit of their mutual customers. It is not unusual for one ISP to accept the announcement from a customer of a subnet of another ISP's address space. This decision is one for those ISPs to make about their own operational environments.

If we choose not to endorse a particular short-term solution to depletion, it falls upon ISPs themselves to find a way to continue their business operations, and resolve their customers' problems. If they cannot get address space from themselves, it will be their *duty* to their customers to get routable address space from somewhere—by negotiating, if necessary, with their peers and upstream providers to change the definition of "routable address space." Ultimately we may assume that if we do not provide a solution to the industry, the industry will invent one—or several competing ones.

Because we assert that the solution that best solves this problem is an address space trading exchange, we may well end up getting one—but one (or more) that is private, and out of sight of our existing policy-making structure. Worse still, competing exchanges would not have access to the RIRs data, and so would not be in a position to assure the quality of a prefix—a situation that could threaten all transactions.

Without exaggerating, it is likely that what we do in response to this crisis will determine the architecture of the Internet for a long while to come. Although we are reminded of Woody Allen's quote wherein he "... hope[s] mankind has the wisdom to choose correctly... between utter hopelessness and total extinction[22, 23]," there are, as we have outlined, measures we can take to survive the coming storm. They are not beautiful solutions. They are not how we have traditionally done things, or even how we would like to do things. Adopting them will almost certainly result in someone being worse off than if we had simply done nothing. But they represent, to our minds, the best, most realistic chance to avoid widespread difficulties and the loss of many of the principles we in the networking community hold dear, to ourselves and in our institutions. Let us begin this process now.

Acknowledgements

The authors would like to gratefully acknowledge help and support from Léan Ní Chuilleanáin, Emma Apted, and David Malone for diligent editing.

References

[0] Murphy, Niall and Wilson, David, "The End of Eternity Part One: IPv4 Address Exhaustion and Consequences," *The Internet Protocol Journal,* Volume 11, No. 4, December 2008.

[1] **ftp://ftp.ietf.org/ietf-online-proceedings/94dec/area.and.wg.reports/ipng/ale/ale-minutes-94dec.txt**

[2] **http://tools.ietf.org/html/rfc2008**

[3] Hain, Tony, "A Pragmatic Report on IPv4 Address Space Consumption," *The Internet Protocol Journal,* Volume 8, No. 3, September 2005.

[4] **http://playground.sun.com/ipv6/doc/history.html**

[5] **http://ipv4.potaroo.net**

[6] **http://www.ripe.net/ripe/meetings/ripe-55/presentations/murphy-simlir.pdf**

[7] **http://www.isoc.org/educpillar/resources/ipv6_faq.shtml**

[8] **http://www.ietf.org/internet-drafts/draft-narten-ipv6-statement-00.txt**

[9] **http://www.apnic.net/meetings/24/program/sigs/policy/presentations/el-nakhal-prop-051.pdf**

[10] **http://www.ripe.net/ripe/policies/proposals/2007-06.html**

[11] **http://www.switch.ch/pki/meetings/2007-01/namebased_ssl_virtualhosts.pdf**

[12] For example, **http://h.root-servers.org/128.63.2.53_2.html versus http://h.root-servers.org/h2_5.html**

[13] **http://www.ripe.net/ripe/meetings/ripe-55/presentations/vegoda-reclaiming-our.pdf**

[14] A "smooth and convenient" dialing plan for India. **http://www.mycoordinates.org/indias-phone-june-06**

[15] **http://en.wikipedia.org/wiki/UK_telephone_code_misconceptions**

[16] **http://code.google.com/p/simlir/**

[17] **http://www.ripe.net/docs/ripe-407.html#membership**

[18] **http://www.ripe.net/ripe/policies/proposals/2007-03.html**

[19] **http://www.ripe.net/ripe/policies/proposals/2007-06.html**

[20] **http://www.ripe.net/ripe/policies/proposals/2007-07.html**

[21] **http://kuznets.fas.harvard.edu/~aroth/alroth.html**

[22] Woody Allen, "Side Effects," 1980.

[23] Woody Allen through (most famously) Stephen Hawking, **http://www.cnn.com/2006/WORLD/asiapcf/07/04/talkasia.hawking.script/index.html**

[24] **http://icann.org/en/announcements/proposal-ipv4-report-29nov07.htm**

[25] **http://www.ripe.net/ttm/**

[26] **http://www.ripe.net/ripe/tf/enhanced-cooperation/index.html**

[27] **http://www.nro.net/documents/nro18.html**

[28] **http://www.ripe.net/maillists/ncc-archives/im-support/2004/index.html**

NIALL MURPHY holds a B.Sc. in Computer Science and Mathematics from University College Dublin. While in university, he founded the UCD Internet Society, which provided Internet access to approximately 5000 students. He went on to work for (and found) various organizations: the .IE domain registry, Club Internet (now Magnet Entertainment), Ireland On-Line, Enigma Consulting, Bitbuzz, and Amazon.com. He is currently in Site Reliability Engineering at Google. He is the coauthor of numerous articles, some RFCs, the O'Reilly book *IPv6 Network Administration,* and is a published poet and keen amateur landscape photographer. E-mail: **niallm@avernus.net**

DAVE WILSON holds a B.Sc. in Computer Science from University College Dublin, not coincidentally from around the same time as Niall. He has worked at HEAnet, the Irish National Research & Education Network, for more than 10 years, maintaining an involvement with RIPE and with the pan-European research network Géant. Dave is a member of the ICANN Address Supporting Organization Address Council; he helped to found the Irish IPv6 task force, which has the support of the national government there. E-mail: **dave.wilson@heanet.ie**

Resource Certification

by Geoff Huston, APNIC

Opinions vary as to what aspect of the Internet infrastructure represents the greatest common vulnerability to the security and safety of Internet users, but it is generally regarded that attacks that are directed at the network infrastructure are the most insidious, and in that case the choice is probably between the *Domain Name System* (DNS) and the interdomain routing system.

The question of how to improve the robustness of these functions has been a longstanding topic of study. For the DNS it appears that there is convergence on *Domain Name System Security Extensions* (DNSSEC) as the technical solution to securing DNS resolution operations, and the focus of attention in this space has shifted from technical behavior to topics relating to operational deployment. It has been a difficult time for DNSSEC and to say that there is an end in sight may well be premature at this stage, but there are definite signs of progress in this space. The same cannot be said of progress with securing routing, and particularly in securing interdomain routing. Here much remains to be done in order to achieve reasonable consensus on what technical measures to adopt, let alone the second step of study of how such measures could be deployed across the Internet.

The IETF's approach to addressing the topic of securing interdomain routing has followed a conventional IETF path. The first step has been to consider the nature of various vulnerabilities that exist within today's interdomain routing system and then develop a set of requirements that should be addressed in any solution space, without necessarily defining what such a solution may be. When the enumeration of requirements achieves a suitable level of consensus from the community, it is then possible to commence work on standardizing solutions. In the case of securing interdomain routing, the first steps were undertaken in *Birds of a Feather* (BOF) sessions and in the subsequently formed *Routing Protocol Security Requirements* (RPSEC) Working Group. This work is almost complete, and apart from some definitive statement relating to a requirement for securing the *Autonomous System* (AS) Path attribute in *Border Gateway Protocol* (BGP), the set of requirements for securing interdomain routing is now in an almost final state[1]. The task of the *Securing Inter-Domain Routing* (SIDR) Working Group is to standardize technologies that can meet these requirements.

So where does "Resource Certification" fit in?

Public Key Cryptography

One commonly used security technology is *Public Key Cryptography,* a technique that is easily explained. The approach uses a pair of keys, A and B. Anything enciphered with key A can be deciphered only with key B, and conversely, and knowledge of the value of one key does not lead to discovery of the value of the other key. Key A is kept as a closely guarded secret, whereas key B is openly published. If I want to send you a message that only you can decipher and read, I should encrypt it using your public key. If I want to send you a message that only I could have sent (nonrepudiation), then I will generate a digital signature of the message using my private key. That way any attempts to alter the message will also be detectable.

This latter approach, of using keys to generate digital signatures of messages, lies at the heart of DNSSEC, because DNSSEC adds public keys and digital signatures to the DNS. A DNS query can generate a response that lists both the DNS answer and the digital signature of that answer. The DNS can also be queried to retrieve the public key used to sign all the components of that zone, so that the digital signature can be verified and the query agent can be assured that the response is a genuine one. But how can the key itself be verified? In DNSSEC the hierarchical nature of the DNS itself is exploited by having each zone "parent" sign the keys of its delegated "children." So the zone key can be verified by retrieving the parent's signature across that zone key, and so on to the root of the DNS. As long as the query agent knows beforehand the value of the public key used to sign the root zone of the DNS, and as long as DNSSEC is used universally, all DNS responses can be verified in DNSSEC.

Although this approach works in the interlocked hierarchical structure of the DNS, when we turn our attention to securing the use of IP addresses and AS numbers in the context of interdomain routing, there is no comparable hierarchy to exploit. In such cases a common solution is to turn to *Digital Certificates*.

Digital Certificates are digitally signed public attestations by a certification authority that associate a subject's public key value with some attribute of the subject. A typical application is in identity certification, where the certification authority is attesting that the holder of the private key whose matching public key is provided in the certificate has met the authority's certification criteria to be identified by a particular name. Digital certificates are useful in that they can reduce the number of trust points in a security domain, so that each member of the domain does not have to validate identity and exchange public keys with every other member of the domain, but can undertake a single transaction with a certification authority that is trusted by all the members of the domain. As long as every member of the domain carries the public key of the certification authority and can access all issued digital certificates, then the members of the domain can verify each other's attestations and digital signatures.

Of course digital certificates are used for far more than attestations of identity, and can encompass the authority to perform specific tasks, undertake particular roles, or grant permissions and right-of-use authorities. It is this latter use case that is relevant to resource certification.

Resource Certificates

A Resource Certificate is a conventional X.509 certificate that conforms to the *Public Key Infrastructure Working Group* (PKIX) profile (RFC 5280) with one critical component, namely a certificate extension that lists a collection of IP number resources (IPv4 addresses, IPv6 addresses, and AS numbers)[17].

These certificates attest that the certificate issuer has granted to the entity represented by the certificate subject a unique "right-of-use" of the associated set of IP number resources listed in the certificate extension, by virtue of an associated resource allocation. The unique "right-of-use" concept mirrors the resource allocation framework, where the certificate provides a means of third-party validation of assertions related to resource allocations[2].

By coupling the issuance of a certificate by a parent *Certification Authority* (CA) to the corresponding resource allocation, a test of the validity of a certificate, including the IP number resource extension, can also be interpreted as validation of that resource allocation. Signing operations that descend from that certificate can therefore be held to be testable, under the corresponding hierarchy of allocation. In other words, if you received your address block from a particular *Regional Internet Registry* (RIR), then only that RIR can issue a Resource Certificate for you that includes your public key and the allocated number resources. Anything you sign using your private key can be verified through the RIR's issued certificate.

Unlike certificates that relate to attestations of identity, Resource Certificates are not necessarily long-lived. When an additional allocation action occurs, the associated Resource Certificate is reissued with an IP number resource extension that matches the new allocation state. In the case of a reduction in allocated resources, the previously issued certificates are explicitly revoked when the new certificate is issued. In other cases there is no explicit revocation of the older certificates.

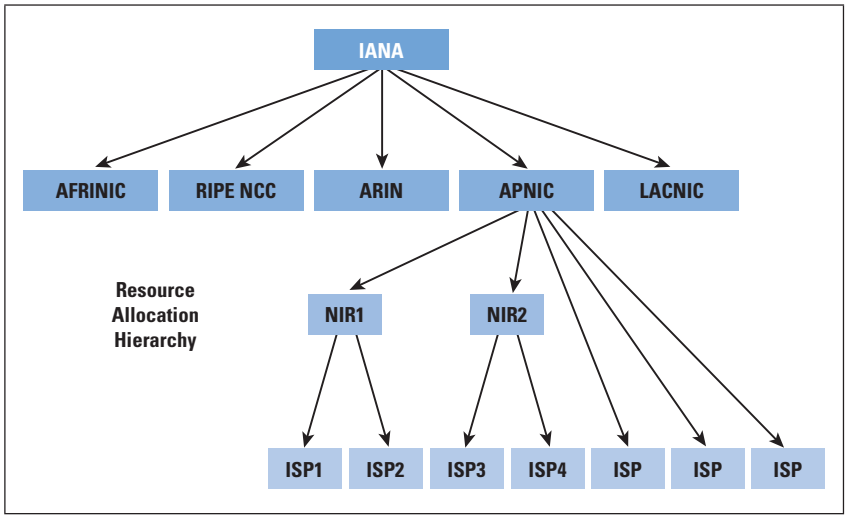
The intention here is that any instrument signed by the subject's private key that relates to an assertion of resource control, whether it is a protocol message in a routing protocol or an administrative request to an *Internet Service Provider* (ISP) to route a prefix or as assertion of title over the "right-of-use" of a number resource, can be validated through the matching public key contained in the certificate and the IP number resource that is enumerated in this certificate. The Resource Certificate itself can be verified in the context of a Resource Certificate *Public Key Infrastructure* (PKI).

The Resource Certificate Public Key Infrastructure

The *Resource Certificate Public Key Infrastructure* (RPKI) describes the structure of the certification framework used by Resource Certificates. The intent of the RPKI is to construct a robust hierarchy of X.509 certificates that allows relying parties to validate assertions about IP addresses and AS numbers, and their use.

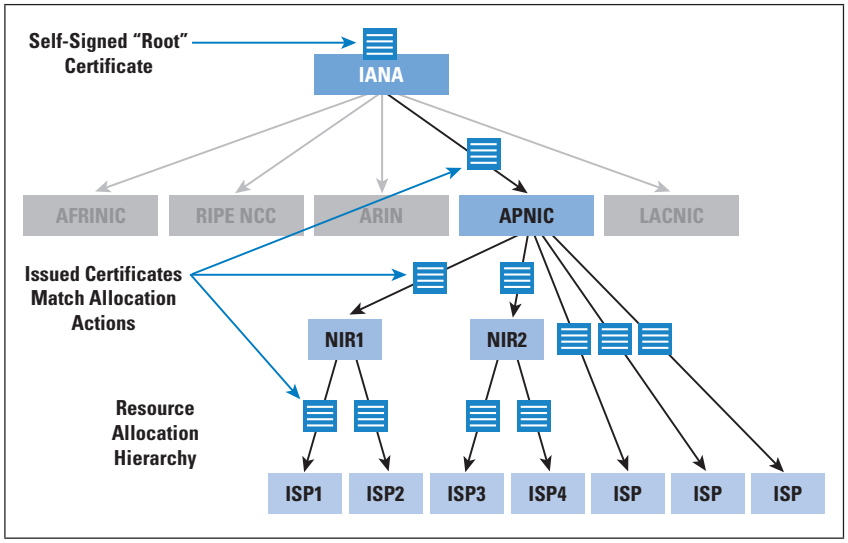
The structure of the RPKI as it relates to public use of IP number resources is designed to precisely mirror the structure of the distribution of addresses and ASs in the Internet, so a brief description of this distribution structure is appropriate. The *Internet Assigned Numbers Authority* (IANA) manages the central pool of number resources. The IANA publishes a registry of all current allocations. The IANA does not make direct allocations of number resources to end users or *Local Internet Registries* (LIRs), and instead allocates blocks of number resources to the RIRs. The RIRs perform the next level of distribution, allocating number resources to LIRs, *National Internet Registries* (NIRs), and end users. NIRs perform allocations to LIRs and end users, and LIRs allocate resources to end users (Figure 1).

Figure 1: Address Distribution Hierarchy for the Internet



The RPKI mirrors this allocation hierarchy. One interpretation of this model would send the IANA manager a root RPKI key, and using this key the IANA would issue a self-signed "root" certificate, and also issue subordinate certificates to each of the RIRs, describing in the resource extension to the certificate the complete set of number resources that have been allocated to that RIR at the time of issuance. The certificate would also hold the public key of the RIR and would be signed by the private key of the IANA. Each RIR would issue certificates that correspond to allocations made by that RIR, where the resource extension to those certificates lists all the allocated resources, and the certificate includes the public key of the recipient of the resource allocation, signed with the private key of the RIR. If the recipient of the resource allocation is an LIR or an NIR, then it too would also similarly issue resources certificates (Figure 2).

Figure 2: RPKI Resource Certificate Hierarchy



The common constraint within this certificate structure is that an issued certificate must contain a resource extension that contains a subset of the resources that are described in the resource extension of the issuing authority's certificate. This requirement corresponds to the allocation constraint than a registry cannot allocate resources that were not allocated to the registry in the first place. One implication of this constraint is that if any party holds resources allocated from two or more registries, then it will hold two or more Resource Certificates in order to describe the complete set of its resource holdings.

Validation of a certificate within this RPKI is similar to conventional certificate validation within any PKI, namely establishing a chain of valid certificates that are linked by issuer and subject from a nominated trust anchor CA to the certificate in question. The only additional constraints in the RPKI are that every certificate in this validation path must be a valid Resource Certificate, and the IP number of resources described in each certificate must be a subset of the resources described in the issuing authority's certificate.

Within this RPKI all Resource Certificates must have the IP addresses and AS resources present, and marked as critical extensions. The contents of these extensions correspond exactly to the current state of IP address and AS number allocations from the issuer to the subject.

Any holder of a resource who can make further allocations of resources to other parties must be able to issue Resource Certificates that correspond to these allocations. Similarly, any holder who wishes to use the RPKI to digitally sign an attestation needs to be able to issue an *End Entity* (EE) certificate to perform the digital signing operation.

For this reason all issued certificates that correspond to allocations are certificates with the CA capability enabled, and each CA certificate is capable of issuing subordinate CA certificates that correspond to further sub-allocations and subordinate EE certificates that correspond to a generation of digital signatures on attestations.

The RPKI makes conventional use of *Certificate Revocation Lists* (CRLs) to control the validity of issued certificates, and every CA certificate in the RPKI must issue a CRL according to the nominated CRL update cycle of the CA. A CA certificate may be revoked by an issuing authority for numerous reasons, including key rollover, the reduction in the resource set associated with the subject of the certificate, or termination of the resource allocation. To invalidate the authority or attestation that was signed by a given EE certificate, the CA issuing authority that issued the EE certificate simply revokes the EE certificate.

Resource Certificates are intended to be public documents, and all certificates and objects in the RPKI are published in openly accessible repositories. The set of all such repositories forms a complete information space, and it is fundamental to the model of securing the public Internet interdomain routing system that the entire RPKI information space is available. Other uses of the RPKI might permit use of subsets, such as the single chain from a given end-entity certificate to a trust anchor, but routing security is considered against all known publicly routable addresses and AS numbers, so all known resource certification outcomes must be available. In other words the intended use of the RPKI in routing contexts is not a case where each relying party may make specific requests for RPKI objects in order to validate a single object, but one where each relying party will perform a regular sweep across the entire set of RPKI objects in order to ensure that the relying party has a complete picture of the RPKI information space.

This aspect of the RPKI represents some interesting challenges, in that rather than having a single CA publish all the certificates produced in a security application at a single point, the RPKI permits the use of many publication points in a widely distributed fashion. Each CA can issue RPKI objects and publish them using a locally managed publication point. It is incumbent upon relying parties to synchronize a locally managed cache of the entire RPKI information space at regular and relatively frequent intervals.

For this reason the RPKI has introduced an additional mechanism in its publication framework, namely the use of a "manifest" to allow relying parties to determine whether they have been able to retrieve the entire set of RPKI published objects from each RPKI repository publication point, or if there has been some attempt to disrupt the relying party's access to the entire RPKI information set.

It also implies that the RPKI publication point access protocols should support the efficient function of a synchronization comparison, so that a locally managed cache of the RPKI need only call for the uploading of those objects that have been altered since the previous synchronization operation.

Signed Attestations and Authorities

The underlying intent of digital certificates, and Resource Certificates in particular, is in terms of supporting a transitive trust relationship that allows a relying party to verify the authenticity of a signed artefact through verification of the signer's key using the PKI. So the obvious question is: what artefacts are useful to sign?

Much of the motivation for Resource Certificates has come from a desire to underpin efforts in securing aspects of interdomain routing. This effort goes well beyond securing the individual point-to-point connection used between BGP speakers, and refers to the matter of verifying the authenticity of the payload of the BGP protocol exchange. The specific question that may be posed is: how can a BGP speaker validate the authenticity of the route object being presented to it?

The approach being studied by the SIDR Working Group is to use structured attestations, where, like the digital certificate itself, the attestation is structured in an ASN.1 digital object, and this object is signed using a signing formation that is itself a piece of structured ASN.1, namely the *Cryptographic Message Syntax* (CMS)[18].

The first of these attestations relates to the ability to verify the authenticity of the "origination" of an interdomain routing object. This verification refers to the address prefix and the originating AS, and the questions that this verification function is intended to answer include:

- Is this a valid address prefix and AS number? Have these resources been allocated through the IP number resource allocation process?
- Has the holder of the title of "right-of-use" for the address prefix authorized the AS holder to originate a routing advertisement for this prefix?

Here an address holder is authorizing a particular ISP to generate a route announcement for its particular address prefix. In this case the prefix holder would generate an EE Resource Certificate with the IP number resource extension spanning the set of addresses that match the address prefixes that are the intended subject of the routing authority, and place validity dates in the EE certificate that correspond to the intended validity dates of the routing authority.

The signed authority document would contain the AS number that is being authorized in this manner, a description of the range of prefixes that the prefix holder has authorized, and the EE certificate. The document would be signed by the EE certificate private key using a CMS signing structure. The resultant object is published in the RPKI distributed publication repository as a *Routing Origination Authorization* (ROA). A relying party can validate the ROA by checking to ensure that the digital signature in the ROA is correct, indicating that the authority document has not been tampered with in any way since it was signed, that the resources in the associated EE certificate encompass the prefixes specified in the document, and the EE certificate itself is valid in the context of the RPKI by verifying that there is an issuer-subject chain of valid certificates that link one of the relying party's nominated trust anchors to the EE certificate.

The ROA itself is valid as long as the signing EE certificate is valid. To withdraw the authority prior to the expiration of the EE certificate, the ROA publisher can simply revoke the EE certificate, leading to the concept of "one-off-use" EE certificates in the RPKI, where a key pair and a corresponding EE certificate are generated in order to sign a single attestation or authority. If the authority's lifetime is extended, the authority is reissued with a new EE certificate and a new digital signature, and, as noted, the authority can be prematurely terminated through revocation of the EE certificate, so at no stage is there a need to reuse the original signing private key. After the private key is used to sign this object, the key is destroyed, alleviating to some extent the key management load.

In any security system knowledge of what is authorized is helpful, but knowledge of what has not been authorized is perhaps even more helpful. For ROAs there is an analogous situation to DNSSEC, where DNSSEC is most effective from a client's perspective after the entire DNS space is DNSSEC signed. Where there are gaps in the DNSSEC signing chains the client is left in an uncertain state regarding the verification outcomes of the unlinked DNS sub-hierarchies. The same could apply to ROAs, in that in an environment where not every originated route object has a published ROA, the absence of a ROA does not necessarily indicate an unauthorized route origination. If one of the objectives of this study is to define a framework that can unambiguously identify the unauthorized use of IP number resources in routing (route "hijacks") even in a world where ROAs are used in a piecemeal fashion, then one possible refinement to the ROA model is the introduction of a comparable negative authority, the *Bogon Origination Attestation* (BOA).

In this case the prefix holder generates a signed attestation, or BOA, in a similar manner to the ROA, but does not provide any originating AS. Instead the BOA refers to "all originating ASs," and has the semantic interpretation that any use in the routing space of this address prefix described in the BOA, or any more specific address prefix, should be regarded as unauthorized and the route should be discarded.

Although this process makes the detection of route hijacks more direct in a world of piecemeal use of ROAs, there is now the added complication of having both "positive" and "negative" authorities. The proposed resolution of this dilemma is to use a relative priority rule that ROAs take precedence over BOAs, so that if a valid ROA and a valid BOA both exist that describe the origination component of a route, then the route can be regarded as authorized.

It should be noted, however, that at this stage these concepts are "work in progress," and are part of the SIDR Working Group's agenda of study, and the working group has not as yet reached any consensus regarding the decision to advance these proposals onward along the Internet Standards Process.

Also on the near-term horizon for SIDR is examining approaches to secure the AS path in BGP updates. The RPSEC Working Group has explored two approaches in this space. One involves an incremental multiple signature technique that allows a receiver of a BGP update to verify that the AS path described in the update is matched by a sequence of interlocking AS digital signatures using the RPKI. At the same time that an AS adds its own AS to the AS path prior to further *External Border Gateway Protocol* (eBGP) propagation of the route update, the AS would digitally sign over an analogous sequence of AS signatures. This approach allows a receiver to perform a match of the AS sequence in the AS path with the AS number sequence identified in the AS signature block. A match here would indicate that the BGP update has indeed been sequentially passed along the sequence identified by the AS path. This approach was originally proposed in the *Secure BGP* (sBGP) design[21] and has attracted some comment related to the computation overhead associated with the application and validation of these AS path signature sequences. An alternative approach has been one that is described by RPSEC as being less rigorous, and refers to a "feasibility" check, which checks to ensure that each pair of ASs represented in the AS path has an associated verifiable assertion of inter-AS adjacency that is digitally signed by both ASs.

It should also be noted that this activity of addressing aspects of improving the robustness of interdomain routing has some previous context. In many parts of the Internet, some degree of routing integrity is managed through the use of *Internet Routing Registries* (IRRs) and the publication of routing policies through the use of *Routing Policy Specification Language* (RPSL) objects.

Although opinions vary as to the robustness of the security offered by the IRR approach, at the very least it can mitigate some weakness in the routing system through the use of a "second check" that can be used to filter the information that is being provided in a BGP feed.

The weaknesses in the IRR system tend to relate to the consistency, completeness, and authenticity of the IRR data, and in many cases the trust in the integrity of the data relies on the admission practices of the IRR itself, and individual data objects cannot be verified by clients of the IRR. One possible way to address this situation has been through the use of *Routing Policy System Security* (RPSS) measures, but the adoption of these measures has not been widespread, and the question still remains for the client that even if an IRR object was authenticated upon admission, it does not mean that when the object is subsequently used by an IRR client the information reflects the current situation, and the information could well be invalid or not reflect the current policies of the author of the IRR object.

One possible approach being considered by the SIDR Working Group is to implement the RPSS authentication models using object signing in the context of the RPKI. For example, the RPSS assumption that routes should be announced only with the consent of the holder of the origin AS number of the announcement and with the consent of the holder of the address space implies in RPSS that both parties should authorize the entry of a *route object* into the IRR. Translating this stipulation into an analogous model using the RPKI would require that a route object be signed with the digital signatures of both the AS holder and the address space holder, and a IRR client can verify this route object at the time of use by verifying both digital signatures. Either the address space holder or the AS holder can revoke authorization by revoking the EE certificate used to sign the route object, and the verification is independent of the particular IRR that has published the route object. It is also a possibility that the IRR itself can be folded into the RPKI distributed publication repository framework, because there is no particular requirement in such an environment for a disparate collection of IRRs with their own partial collections of routing policy information, although at this stage this discussion is heading into the realm of more advanced speculation about the potential for application of Resource Certificates and digital signatures to RPSL and the IRR framework.

Putting Resource Certificates into Context

Resource Certificates and the associated RPKI represent a major part of any effort to construct a secure interdomain routing framework. An RPKI, even partially populated with signed information, allows BGP speakers to make preferential selections to use routing information where the IP address block and the AS numbers being used are recognized as valid to use, and the parties using these IP addresses and AS numbers are properly authorized to so do. The RPKI can also be used to identify instances of unauthorized use of IP addresses and attempts to hijack routes.

However, the RPKI represents only one part of a larger framework of securing interdomain routing, and the next step is that of applying the RPKI to the local BGP processing framework. There is also the need to move beyond validation of route origination and look at the associated topic of validation of the AS path, and potentially to consider the most challenging task, of attempting to validate whether the initial forwarding decision associated with a route object actually represents the correct first hop along a usable forwarding path for packets to reach the network destination.

The concerns here include not only a consideration of what can be secured and validated, but matters of scalability and efficiency in terms of deployment cost. The various approaches to routing security studied so far offer a wide variety of outcomes in terms of the amount of routing information that is validated, the level of trust that can be placed in a validation outcome, and the overheads of generating and validating digital signatures on routing information. The next step appears to include the task of establishing an appropriate balance between the overheads of operating the security framework and the extent to which efforts to disrupt the routing system can be successfully deflected by such measures.

The RPKI has been designed as a robust, simple framework. As far as possible existing technologies and processes have been exploited, reflecting to some extent a level of conservatism of the routing community and the difficulty in securing widespread acceptance of novel technologies.

References and Further Reading

The following documents provide further detail about the IETF work on resource certification. The Internet Drafts listed here are still a "work in progress," and although they are reflective of the areas of activity of the SIDR Working Group, they do not necessarily represent finished work.

Internet Drafts

Requirements:

[1] B. Christian, T. Tauber, eds., "BGP Security Requirements," work in progress, Internet Draft, **`draft-ietf-rpsec-10.txt`**, November 2008. *The report of the consensus outcomes of the RPSEC Working Group in enumerating the requirements for securing interdomain routing. The outstanding topic in this report remains in the area of AS path validation and the level of requirement associated with the two approaches described in the report.*

Architecture:

[2] M. Lepinski, S. Kent, "An Infrastructure to Support Secure Internet Routing," work in progress, Internet Draft, **`draft-ietf-sidr-arch-04.txt`**, November 2008. *An overview of the RPKI approach, describing the RPKI, the distributed repository structure, and common operations.*

Resource Certificates:

[3] G. Huston, G. Michaelson, R. Loomans, "A Profile for X.509 PKIX Resource Certificates," work in progress, Internet Draft, **`draft-ietf-sidr-res-certs-15.txt`**, November 2008. *The specification of the Resource Certificate.*

RPKI Repository Structure:

[4] G. Huston, G. Michaelson, R. Loomans, "A Profile for Resource Certificate Repository Structure," work in progress, Internet Draft, **`draft-ietf-sidr-repos-struct-01.txt`**, October 2008. *A description of the proposed distributed publication repository structure for the RPKI, including contents, access protocols, and object name conventions.*

[5] R. Austein et al., "Manifests for the Resource Public Key Infrastructure," work in progress, Internet Draft, **`draft-ietf-sidr-rpki-manifests-04.txt`**, October 2008. *A specification for repository manifests. Manifests are signed constructs that describe all the objects currently loaded into a repository publication point, and are used by relying parties as a means of ensuring that a local RPKI repository cache is correctly synchronized against the authoritative original publication point.*

[6] G. Huston, R. Loomans, B. Ellacot, R. Austein, "A Protocol for Provisioning Resource Certificates," work in progress, Internet Draft, **`draft-ietf-sidr-rescerts-provisioning-03.txt`**, August 2008. *A proposed protocol for use between a subject and a certificate issuer to ensure that certificate requests, the IP number resource allocation state, and the issued certificate status are correctly synchronized. This synchronization extends the conventional certificate request model into a transaction protocol that also includes the ability to perform certificate revocation requests and status queries from the subject.*

RPKI Signed Objects:

[7] M. Lepinski, S. Kent, D. Kong, "A Profile for Route Origin Authorizations (ROAs)," work in progress, Internet Draft, **`draft-ietf-sidr-roa-format-04.txt`**, November 2008. *The specification of the syntax for signed ROAs.*

[8] G. Huston, T. Manderson, G. Michaelson, "A Profile for Bogon Origin Attestations (BOAs),"work in progress, Internet Draft, **`draft-ietf-sidr-bogons-02.txt`**, October 2008. *The specification of the syntax for signed BOAs.*

[9] G. Huston, G. Michaelson, "Validation of Route Origination in BGP Using the Resource Certificate PKI," work in progress, Internet Draft, **`draft-ietf-sidr-roa-validation-01.txt`**, October 2008. *The specification of the semantics of ROAs and BOAs and the manner in which these objects may be interpreted in terms of the integration of these origination security credentials onto a BGP route-selection process.*

Certificate Policy and Practice Statements:

[10] K. Seo, R. Watro, D. Kong, S. Kent, "Certificate Policy (CP) for the Resource PKI (RPKI)," work in progress, Internet Draft, **draft-ietf-sidr-cp-04.txt**, November 2008. *A description of the certificate policy that applies to all certificates issued within the RPKI framework.*

[11] D. Kong, K. Seo, S. Kent, "Template for an Internet Registry's Certification Practice Statement (CPS) for the Resource PKI (RPKI)," work in progress, Internet Draft, **draft-ietf-sidr-cps-irs-04.txt**, November 2008. *A template for the Practice Statement used by Internet Registries (IRs) to describe their operational practices in the issuance and management of Resource Certificates.*

[12] D. Kong, K. Seo, S. Kent, "Template for an Internet Service Provider's Certification Practice Statement (CPS) for the Resource PKI (RPKI)," work in progress, Internet Draft, **draft-ietf-sidr-cps-isp-03.txt**, November 2008. *A template for the Practice Statement used by ISPs to describe their operational practices in the issuance and management of Resource Certificates.*

Individual Submissions:

[13] G, Huston, G. Michaelson, "A Profile for AS Adjacency Attestation Objects," work in progress, Internet Draft, **draft-huston-sidr-aao-profile-00.txt**, September 2008. *The specification of the syntax for a pairwise inter-AS routing adjacency attestation.*

[14] R. Kisteleki, J. Boumans, "Securing RPSL Objects with RPKI Signatures," work in progress, Internet Draft, **draft-kisteleki-sidr-rpsl-sig-00.txt**, October 2008. *The specification of the addition of RPKI digital signatures to RPSL Objects in the context of an Internet Route Registry.*

[15] T. Manderson, G. Michaelson, "RPKI Repository Retrieval Mechanism," work in progress, Internet Draft, **draft-manderson-sidr-fetch-00**, October 2008. *A proposed mechanism to use the manifest as the basis for performing a synchronization operation between a local RPKI cache and a source point.*

RFCs:

[16] D. Cooper et al., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280, May 2008.

[17] C. Lynn, S. Kent and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers," RFC 3779, June 2004.

[18] R. Housley, "Cryptographic Message Syntax (CMS)," RFC 3852, July 2004.

[19] C. Alaettinoglu, et al., "Routing Policy Specification Language (RPSL)," RFC 2622, June 1999.

[20] C. Villamizar et al., "Routing Policy System Security," RFC 2725, December 1999.

Other Documents:

[21] Kent, S., "Securing BGP: S-BGP," *The Internet Protocol Journal,* Volume 6, No. 3, September 2003.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. The author of numerous Internet-related books, he is currently the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of the Internet Society from 1992 until 2001. E-mail: **gih@apnic.net**

Host Identity Protocol: Identifier/Locator Split for Host Mobility and Multihoming

by Andrei Gurtov and Miika Komu, Helsinki Institute for Information Technology, and Robert Moskowitz, ICSAlab

A host and its location are identified using *Internet Protocol* (IP) addresses in the current Internet architecture. However, IP addresses can serve only as short-term identifiers because a considerable amount of hosts are *portable* devices and they change their IP addresses when moved from one network to another. Short-term identifiers disrupt long-term transport layer connections, such as Internet phone calls, and make locating the peer host more difficult. Therefore, mobility and multihoming are hard to implement securely in the present Internet. Upon changing an IP address, the host must prove to its peers that it is the same entity they communicated with before, requiring the use of cryptographic identities.

Another challenge the Internet faces is due to the fact that deployed protocols in the Internet are prone to *Denial-of-Service* (DoS) attacks. Substantial memory state can be created before the communicating peer is authenticated. Impersonation attacks are possible because IP addresses are relatively easy to forge. Because of difficulties in configuring *IP Security* (IPsec) for users, most Internet traffic is still transmitted in plaintext, making it easy for attackers to collect passwords or lists of visited websites, for example, in public *Wireless Local-Area Networks* (WLANs). As the IPv6 protocol is seeing gradual deployment, interoperating traditional IPv4 applications with new IPv6 applications remain a challenge.

The so-called *identifier/locator split* is recognized by the *Internet Engineering Task Force* (IETF) community as a next big change in the Internet architecture. Although the problem has been known for a long time[17], it has only recently started to get sufficient attention. Developments in public key cryptography and increased computational resources of hosts enables the use of cryptographic mechanisms to securely handle identities. Several proposals are under consideration in the IETF, including the *Locator Identifier Separation Protocol* (LISP)[16] for the network-based and the *Host Identity Protocol* (HIP) for the host-based approach. LISP focuses on improving scalability of the routing system, whereas HIP provides secure end-to-end mobility and multihoming. Therefore, the two proposals are complementary rather than competing.

HIP Architecture

The HIP architecture[1,2] uses the identity/locator split advantage to address Internet architecture challenges in an integrated approach. HIP was proposed by Bob Moskowitz in 1999 and since then has been under active development in the IETF Working Group and *Internet Research Task Force* (IRTF) Research Group.

HIP enables host mobility and multihoming across different address families (IPv4 and IPv6), offers end-to-end encryption and protection against certain DoS attacks, allows moving away from IP address-based access control to permanent host identities, and restores end-to-end host identification in the presence of several addressing domains separated by *Network Address Translation* (NAT) devices.

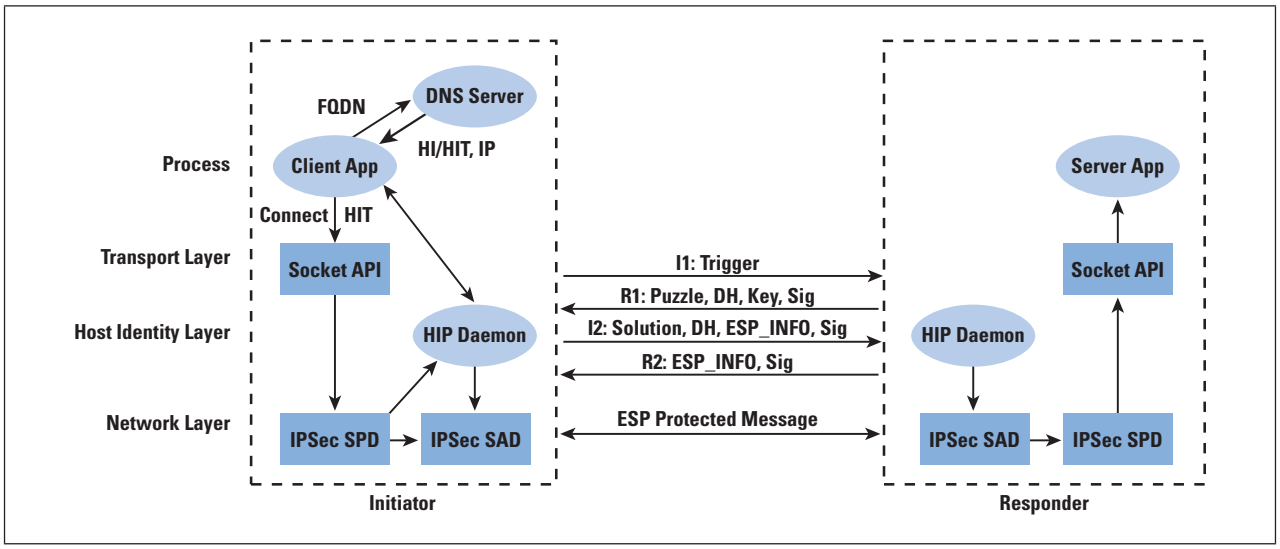
HIP separates the identity of a host from its location. The location of the host is bound to IP addresses and used for routing packets to the host in the same way as in the current Internet architecture. However, transport and application layers use *host identity,* consisting of the public key component of a private-public key pair. Each host is responsible for creating one or more public/private key pairs to provide identities for itself. Because the host identities are based on public key cryptography, they are computationally difficult to forge. Host identities are location-independent identifiers that allow a mobile host to preserve its transport layer connections upon movement. On the other hand, the host identity can be used for looking up the current location of a host because the host identity is a long-term identifier. A client host obtains the host identity of a server typically from the *Domain Name System* (DNS)[7] or a *Distributed Hash Table* (DHT). However, the infrastructure may not support this DHT in certain scenarios, such as in peer-to-peer and temporary environments. In such cases, *opportunistic* HIP can be used for contacting a peer without prior information of the identity of the peer. Opportunistic HIP is based on a "leap-of-faith," meaning that it is prone to man-in-the-middle attacks for the initial connection. It is similar to the *Secure Shell* (SSH) *Protocol,* where the public key of the server is added to the known host list after the first connection.

The problem of certifying the keys in *Public Key Infrastructure* (PKI) or otherwise creating trust relationships between hosts has explicitly been left out of the HIP architecture, because it is expected that each system using HIP may want to address it differently. For mere mobility and multihoming, the systems can work without any explicit trust management, in an opportunistic manner.

All other parties use the host identifier, that is, the *public key,* to identify and authenticate the host. Typically, a host identifier is a 128-bit-long bit string, the *Host Identity Tag* (HIT), as shown in Figure 1. A HIT is constructed by applying a cryptographic hash function over the public key. The introduction of new endpoint identifiers changes the role of IP addresses. When HIP is used, IP addresses become pure topological labels, naming locations in the Internet. One benefit of this identity/locator separation is that hosts in private address realms (behind NATs) can name each other in a unique way with HITs. A second benefit is that the hosts can change their IP address without breaking transport layer connections of applications and rely on HIP to manage host mobility; the relationship between location names and identifiers becomes dynamic.

To start communicating through HIP, two hosts must establish a HIP association. Known as the HIP *Base Exchange* (BEX)[3], this process consists of four messages (I1, R1, I2, and R2) transferred between the initiator and the responder. After BEX is successfully completed, both hosts are confident that private keys corresponding to host identifiers (public keys) are indeed possessed by their peers. Another purpose of the HIP base exchange is to create a pair of IPsec *Encapsulated Security Payload* (ESP) *Security Associations* (SAs), one for each direction. HIP uses IPsec ESP *Bound End-to-End Mode* (BEET)[4,9] to provide data encryption and integrity protection for network applications.

Figure 1: HIP Architecture



Because neither transport layer connections nor security associations created after the HIP base exchange are bound to IP addresses, a mobile client can change its IP address (that is, upon moving, because of a *Dynamic Host Configuration Protocol* [DHCP] lease or IPv6 router advertisement) and continue to transmit ESP-protected packets to its peer. HIP supports such mobility events by implementing an end-to-end three-way UPDATE signaling mechanism[8] between communicating nodes. HIP multihoming uses the same mechanisms as mobility for updating the peer with a current set of host IP addresses.

A rendezvous server[6] provides a mechanism to locate a host, for example, when two communicating hosts move simultaneously. To employ a rendezvous mechanism, a host first must perform a registration procedure[5], which is an extended version of the HIP base exchange.

The HIP control packets as well as ESP-encapsulated data packets have difficulties in going through NAT applications and firewalls. To traverse NAT, HIP uses *User Datagram Protocol* (UDP)-based encapsulation provided by the *Interactive Connectivity Establishment* (ICE) protocol.

It enables two hosts located behind NAT to communicate through a Rendezvous server. Bob Moskowitz suggests an alternative approach, where HIP always uses IPv6 for end-to-end communication and the *Teredo* protocol is employed to traverse NAT instances in IPv4 networks if native IPv6 connectivity is not available.

Most Internet applications can run unmodified over HIP[10], although only HIP-aware (new) applications using the extended socket interface can take better advantage of the new features that HIP provides. As HIP secures application data traffic with IPsec that is located logically "deep" within the networking stack, the challenge is to provide proper and understandable security indicators to the user to convince the user that the connection, for example, to a banking website, is secured. Such indicators can be developed as extensions to applications (for example, a security plug-in to the *Firefox* browser) or within a hostwide HIP management utility that controls all applications.

HIP provides a network layer alternative to using *Secure Sockets Layer/Transport Layer Security* (SSL/TLS) for application security, which has its benefits and drawbacks. HIP is a generic solution that should work for any transport protocol, whereas until recently TLS supported only TCP. HIP enables host mobility and multihoming, which is not supported by TLS. TLS runs on top of TCP, leaving it vulnerable to various TCP attacks; for example, using spoofed *reset* (RST) packets or DoS attacks with SYNs. Applications must be designed explicitly to use TLS, whereas HIP can provide security as an add-on to existing traditional applications. On the other hand, TLS does not have a problem with traversing traditional middleboxes such as NATs and firewalls that need special attention for HIP. Both protocols share the characteristic of endorsing host identity. TLS relies on certificates issued by one of the known Certification Authorities, whereas HIP can use *Domain Name System Security Extensions* (DNSSEC)[18] or a PKI infrastructure.

There are currently three open-source interoperating HIP implementations. *OpenHIP* from Boeing runs on Linux, Windows, and Mac OS, whereas *HIP on Linux* (HIPL) runs on Linux and Symbian, and *HIP for Inter.net* from Ericsson runs on FreeBSD and Linux. Several testbeds are deployed based on HIP, including the Everett Boeing factory[11], the P2PSIP pilot in Finland[14], and Wi-Fi P2P Internet Sharing Architecture in Germany[12]. Ericsson NomadicLab and TeliaSonera have demonstrated using HIP for transparent IPv4 and IPv6 handovers, mobile router, simultaneous multiaccess, and the use of proxy for traditional hosts[13,15].

Acknowledgements

We are grateful to Pekka Nikander, Tom Henderson, and others in the IETF and the *Internet Research Task Force* (IRTF) community who were encouraging and contributing to the development of HIP. We thank Andrey Khurri for the figure on HIP architecture and Henry Sinnreich for encouraging us to write this article.

We also thank members of InfraHIP II project for comments helping to improve this article.

References

[1] Moskowitz, R. and Nikander, P., "Host Identity Protocol Architecture," RFC 4423, May 2006.

[2] Gurtov, A., *Host Identity Protocol (HIP): Towards the Secure Mobile Internet,* ISBN 978-0-470-99790-1, Wiley and Sons, June 2008.

[3] Moskowitz, R., Nikander, P., Jokela, P. and Henderson, T., "Host Identity Protocol," RFC 5201, April 2008.

[4] Jokela, P., Moskowitz, R. and Nikander, P., "Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP)," RFC 5202, April 2008.

[5] Laganier, J., Koponen, T. and Eggert, L., "Host Identity Protocol (HIP) Registration Extension," RFC 5203, April 2008.

[6] Laganier, J. and Eggert, L., "Host Identity Protocol (HIP) Rendezvous Extension," RFC 5204, April 2008.

[7] Nikander, P. and Laganier, J., "Host Identity Protocol (HIP) Domain Name System (DNS) Extension," RFC 5205, April 2008.

[8] Nikander, P., Henderson, T., Vogt, C. and Arkko, J. "End-host Mobility and Multihoming with the Host Identity Protocol," RFC 5206, April 2008.

[9] Nikander, P. and Melen, J., "A Bound End-to-End Tunnel (BEET) Mode for ESP," Internet Draft, Work in Progress, **draft-nikander-esp-beet-mode-09**

[10] Henderson, T., Nikander, P. and Komu, M., "Using the Host Identity Protocol with Legacy Applications," RFC 5338, September 2008.

[11] Boeing, "Secure Mobile Architecture (SMA) for Automation Security," **http://www.isa.org/wsummit/presentations/Boeing-NGI_SMA_Automation_Security_Vancouver_ISA_presentationtemplates_7-23-07.ppt**

[12] Heer, T., Götz, S., Weingärtner, E. and Wehrle, K., "Secure Wi-Fi Sharing on Global Scales," in Proceedings of the 15th International Conference on Telecommunication (ICT), St. Petersburg, Russian Federation, IEEE, 2008. **https://www.ds-group.info/members/heer/publications-tobias-heer/pdfs/HeerEtAl2008.pdf**

[13] Jokela, P., Ylitalo, J., and Salmela, P., "HIP Mobile Router Demo," March 2007.
http://www.ietf.org/proceedings/07mar/slides/HIPRG-3.pdf

[14] Koskela, J., Heikkila, J. and Gurtov, A., "A Secure P2PSIP System with SPAM Prevention," Poster at ACM Mobicom, September 2008.

[15] Korhonen, J., Mäkelä, A., and Rinta-aho, T., "HIP Based Network Access Protocol in Operator Network Deployments," in First Ambient Networks Workshop on Mobility, Multiaccess, and Network Management (M2NM'07), Sydney, Australia, October 2007.

[16] Meyer, D., "The Locator Identifier Separation Protocol (LISP)," *The Internet Protocol Journal,* Volume 11, No. 1, March 2008.

[17] Saltzer J., "On The Naming and Binding of Network Destinations," RFC 1498, September 1992.

[18] Gieben, M., "DNSSEC: The Protocol, Deployment, and a Bit of Development," *The Internet Protocol Journal*, Volume 7, No. 2, June 2004.

[19] Sinnreich, H., "Letter to the Editor," *The Internet Protocol Journal,* Volume 11, No. 3, page 37, September 2008.

ANDREI GURTOV received M.Sc and Ph.D. degrees in Computer Science from the University of Helsinki, Finland. He presently is Principal Scientist, leading the Networking Research group at the Helsinki Institute for Information Technology, focusing on distributed system security and next-generation Internet architecture. He co-chairs the IRTF research group on HIP and teaches as an adjunct professor at Helsinki University of Technology. He is a regular visitor of the ICSI Center for Internet Research (ICIR) at Berkeley. Andrei has co-authored more than 50 publications, including a book, research papers, patents, and RFCs. He can be reached through the webpage: **http://www.hiit.fi/~gurtov**

MIIKA KOMU received his M.Sc. from Helsinki University of Technology and continues his studies as a postgraduate student. He is working as a full-time researcher and software engineer at Helsinki Institute for Information Technology. He is an active IETF participant and co-author of RFC 5338. Miika is an open source advocate and martial arts fan. E-mail: **miika.komu@hiit.fi**

ROBERT MOSKOWITZ is senior technical director for ICSA Labs and is an active member in the IAB, IETF, and IEEE. At ICSA Labs, Moskowitz leads the IPsec product and system certification program. Prior to the ICSA, he led the adoption of the world's largest IPsec network deployment servicing the automotive industry. As a former co-chair of the IPsec Working Group, Moskowitz provided a user set of multivendor, multipolicy, and multiuser requirements that galvanized many of the debates on the use of IPsec. A contributing editor for *Network Computing Magazine,* Moskowitz is currently helping define the new security component for the 802.11 standard. E-mail: **rgm@htt-consult.com**

Fragments

Allocation Policy for the Remaining IPv4 Address Space Ratified by ICANN

On 6 March 2009, the *International Corporation for Assigned Names and Numbers* (ICANN) Board ratified the *Global Policy for the Allocation of the Remaining IPv4 Address Space*. The policy requires ICANN to reserve one /8 for each *Regional Internet Registry* (RIR) from the *Internet Assigned Numbers Authority* (IANA) free pool. This has been done. The remainder of the implementation will be done once the IANA free pool has been fully allocated to RIRs. There are currently 32 unallocated unicast IPv4 /8s. 27 are in the IANA free pool and five are reserved under the Global Policy for the Allocation of the Remaining IPv4 Address Space.

On 4 February 2009, the Chair of the *Address Supporting Organization Address Council* (ASO AC) forwarded the Proposed Global Policy for the Allocation of the Remaining IPv4 Address Space for ratification by the ICANN Board. On 5 March 2009, the ASO AC submitted advice in full support of the proposal to the ICANN Board. This proposed global policy had been submitted to the ASO AC by the Executive Council of the *Number Resource Organization* (NRO) on 3 December 2008, and adopted by the ASO AC on 8 January 2009. Each RIR community individually discussed the policy and approved its adoption via its own policy development process. The policy text is published on the ICANN web site at:

`http://www.icann.org/en/general/allocation-remaining-ipv4-space.htm`

ISOC's Trust and Identity Initiative

The Internet Society's *Trust and Identity Initiative* recognizes that in order to be trusted, the Internet must provide channels for secure, reliable, private, communication between entities, which can be clearly authenticated in a mutually understood manner. The mechanisms that provide this level of assurance must support both the end-to-end nature of Internet architecture and reasonable means for entities to manage and protect their own identity details.

A *trusted* Internet takes into account security, transaction protection, and identity assertion and management. Given the network dependence on unique numbers and the escalating amount of geolocation data being gathered, the privacy implications of the current Internet represent a significant and growing concern. Trust must be a primary design element at every layer of the architecture, and in some cases, existing elements may need to be redesigned or improved to meet emerging requirements.

In late 2007, the ISOC Board of Trustees held an intensive retreat to consider ISOC's role in identifying and pursuing trust and identity issues. The report arising from that meeting, "Trust and the Future of the Internet,"[1] forms the basis of ISOC's current long term strategic initiative.

The Trust and Identity initiative focuses on the following major research programs:

- *Architecture and Trust:* This research program investigates the implementation of open-trust mechanisms throughout the full cycle of Internet research, standardization, development, and deployment.
- *Current Problems and Solutions and Trust:* This research program investigates the mitigation of the social, policy, and economic factors that may hinder development and deployment for trust-enabling technologies.
- *Identity and Trust:* This research program investigates the elevation of identity to a core issue in network research and standards development. ISOC is taking a lead role in reviewing the current Internet architecture and the model of Internet development and deployment. This includes active engagement with participants within the traditional ISOC sphere, as well as with the research, enterprise, and end-user communities. We offer the kind of support for research that enhances and facilitates trust and collaboration with the standards community and that advances the most interesting outcomes of that research.

ISOC is reaching out to the businesses and end users that rely on the Internet to exchange sensitive data. Their needs and concerns inform both our baseline research agendas and ongoing standards and development work. ISOC continues to support the advancement of current technical solutions and best practices through our existing programs.

[1] "Trust and the Future of the Internet,"
http://www.isoc.org/isoc/mission/initiative/docs/trust-report-2008.pdf

[2] "Trust and Identity Initiative" brochure,
http://www.isoc.org/pubs/isoc/docs/trust.pdf

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

June 2009 Volume 12, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

After many years of using DSL as my only Internet access option from home, I recently upgraded to a broadband solution provided by a cable modem. As a result, I faced the task of renumbering (and partially rewiring) my home network. As you might have guessed, the addressing scheme provided by my new ISP offers *Network Address Translation* (NAT), as well as a small number (5) of fixed IPv4 addresses, the latter at an extra cost as you might expect. I probably should have tried to enable IPv6 just as an experiment, but this task will have to wait for another day. In the meantime, I was pleased to find a relatively user-friendly web interface to the cable modem that allows me to configure numerous parameters, including the range of the *Dynamic Host Configuration Protocol* (DHCP) pool so that certain devices (printers and wireless access points in particular) can have fixed IP addresses for ease of use and configuration. The entire exercise, which took a couple of hours on my very small network, reminded me of what network managers face every day, particularly as they consider the inevitable migration to IPv6. Let me take this opportunity to invite you to share your network management and operations experience, plans for IPv6 migration, and so on. You can send us Letters to the Editor or article proposals. The address, as always, is **ipj@cisco.com**

The *Domain Name System* (DNS) has been the target of attacks over its many years of existence. In recent years, new attacks have emerged that exploit some of the attributes of the DNS protocol and its implementation. One of the corrective measures is to improve the security of DNS caches. There are several ways to improve cache security, most of which involve changing the protocol. Another way, without changing the protocol, is to reduce the attack surface of your cache by shrinking the number of users of any given cache. Our first article, by Bill Manning, explores this view in more detail.

This journal has covered numerous current and emerging *wireless* technologies such as Bluetooth, Wi-Fi, WiMAX, and mobile cellular systems. In this issue, Esa Piri and Kostas Pentikousis describe *Media-Independent Handovers* (MIH), which allow mobile devices to use different wireless and wired network infrastructures transparently. The protocols associated with operation across such diverse access networks are being standardized by the IEEE 802.21 working group.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Intermediate DNS Caching as an Attack Vector

by Bill Manning

The *Domain Name System* (DNS) specification calls for the use of *caching*. Caching is expected to improve the overall responsiveness of the system by ensuring that answers to questions are known and stored locally and that the query load placed on the authoritative servers is minimized. Certain presumptions are associated with caches that may no longer hold. This article looks at some of these presumptions and explores some of the problems that emerge when they are violated. Based on our observations, we offer some recommendations on DNS cache best practices and show our results of testing these practices.

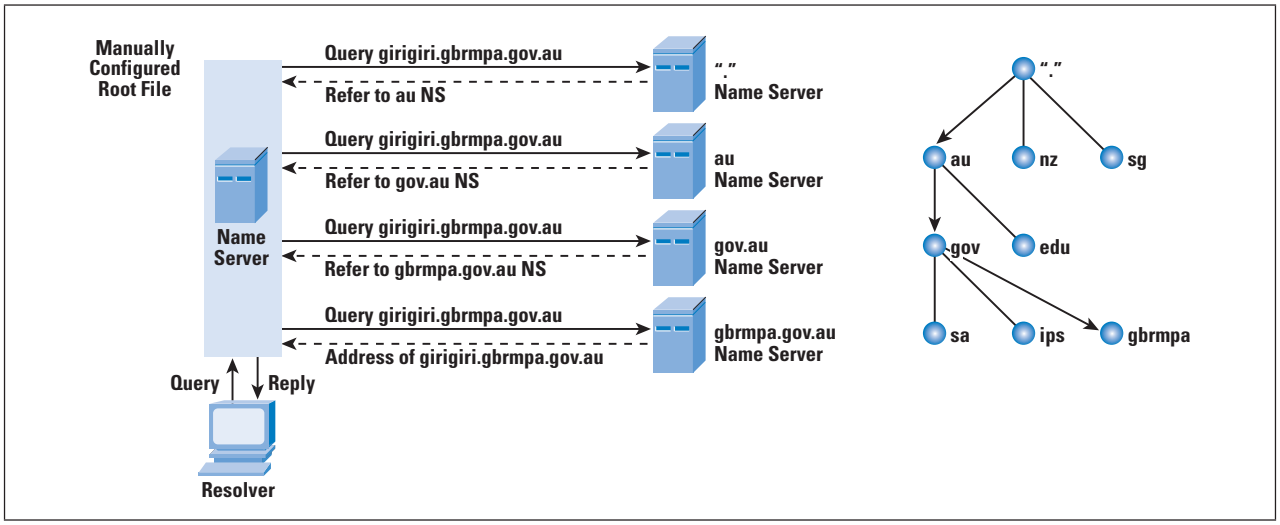
The Problem

A DNS resolver can no longer trust the data it gets—because the data generally comes from nonauthoritative nodes or caches operated by third parties, most of whom have no vested interest in providing accurate data. Removing or bypassing caching from the DNS and going directly to the authoritative servers is considered a fatal flaw because authoritative servers are presumed to have neither the bandwidth nor the processing power to accommodate the perceived demand from a cacheless service. This article looks at the bandwidth and processing capabilities of modern authoritative servers to ascertain the viability of these presumptions. We start by looking briefly at the DNS.

The DNS

The DNS namespace is made visible and useful by nodes publishing authoritative information about the namespace and *resolvers* that send queries about the namespace to these servers. As an optimization, other nodes may act as intermediates or proxies for the authoritative servers for one to many resolvers. These intermediate nodes are called *caching nameservers* or *iterative mode resolvers*. This flow is shown in Figure 1.

Figure 1: DNS Query Flow



Several assumptions about the use and placement of caches have been questioned recently. The simplest is one of placement. A cache works best when the *Round-Trip Time* (RTT) between the resolver and the cache is low. Historically, a cache was placed at traffic aggregation points such as an *Internet Service Provider* (ISP) operating a cache for its clients. With increased mobility of nodes, this presumption is no longer as firm. There are reported cases where resolvers continue to use caches 300 ms away, while an authoritative server is 15 ms away. So if the intent is to reduce network bandwidth, then a cache presuming its client resolvers are all "local" might be misconstrued.

Fixing a resolver to a specific cache does have the benefit of being tied to a known business relationship; for example, using your ISP's caching service. In contrast, mobile nodes often get an IP address from a provider's *Dynamic Host Configuration Protocol* (DHCP) servers, which also hand out more "local" caching servers to be used by the mobile node.

This scenario would be fine—as long as the DNS namespace was in fact a coherent, single space. Unfortunately it is not. So-called *Walled-Garden* networks that have their own versions of DNS namespace have been and remain common. In the Internet, there are more and more alternate root hierarchies that diverge from what most think of as "the" root namespace in either subtle or wildly divergent ways. To date, there is no deployed way for a resolver to determine the origin of the data stored in a cache. A resolver then has no way other than verification of the data to know that the locally assigned cache is in fact using the namespace desired. This situation represents one important reason for going back to a well-known cache, even if it is topologically remote. But this assumption may no longer be valid.

ISPs and even some caching service providers are starting to manipulate caches as a means to monetize their operations.[1] Numerous techniques are in use, from the nominally benign method of using wildcards to more insidious capture and rewrite of NXDOMAIN replies, to outright intentional cache pollution.

In this climate, a resolver should choose its cache carefully. We argue that it is reasonable, in many of today's environments, to place the cache within 1 ms of the resolver; for example, run a cache on the local node. This argument is an extension of the assertion[2] that claims that caches are effective for client populations that are about 10 or fewer.

This technique has the added advantage of reducing the "attack surface" by reducing the effect of cache poisoning or rewriting replies to a small handful of nodes. The perceived disadvantage is the increased load on network bandwidth and query load on authoritative servers as the number of caches increases.

The Experiment

Our experiment has two parts: first we looked at authoritative server processing capabilities and then at the bandwidth effects of a larger number of caches.

Authoritative service is generally run on systems with modern software, supporting threading or precomputed responses. Independent testing shows that these stock software solutions can, on current hardware, support query rates in the hundreds of thousands of queries per second.[3]

A brief survey of authoritative server operators indicates that normal query rates range from 12,000 to 64,000 queries per second.[4,5,6]

On the surface, this result would indicate that there is enough overhead to be able to process more queries, regardless of how they are originated. Regarding bandwidth, a survey of *Top-Level Domain* (TLD) operators has shown that 92 percent of the delegations have two or more authoritative servers for that data on networks with a minimum uplink bandwidth of 100 Mbps. Selected path characterization from clients to target authoritative servers seems to support our presumption that bandwidth is not of concern.

The DNS was designed to function as a roughly symmetrical transfer of information: a request or query is sent and the reply reflects the query and supplies the answer and additional data. Historically, the request and reply were within the same order of magnitude. Into the future, this model may no longer be valid. With *Domain Name System Security Extensions* (DNSSEC), *IP Version 6* (IPv6), and *Naming Authority Pointer* (NAPTR) records being possible candidates in the *Resource Record set* (RRset), the traffic profile more resembles an HTTP request/response, with a significant amount of data being returned from a simple question.[7]

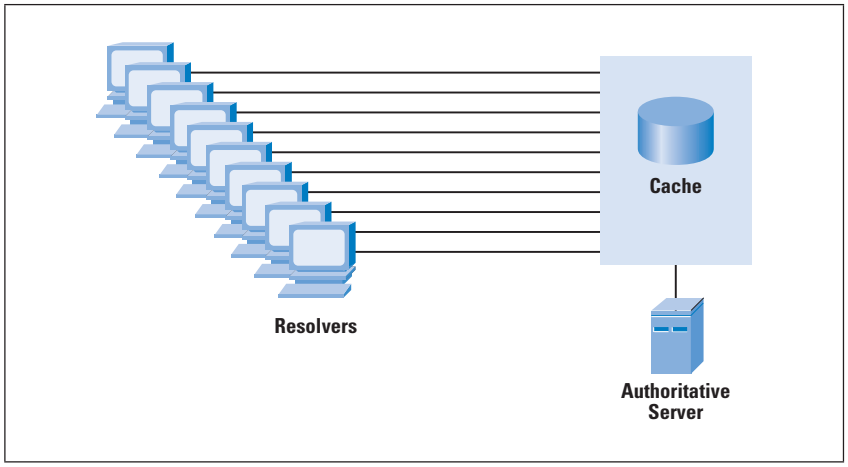
With this information, we can project a worse case in today's environment where a query/reply is about 260 bytes to a worst case in a future environment where a query/reply is about 9 KB, clearly indicating that the amount of bandwidth to authoritative servers needs to grow as new DNS capabilities are deployed, but for the nonce, most have a bandwidth overhead sufficient to absorb a modest change in the number of queries presented.

Modification of the Number of Caching Servers

We began with a cache that serviced 140 stub resolvers on the *University of Southern California's Information Sciences Institute* (USC/ISI) campus in a "normal" dense cache mode (Figure 2).

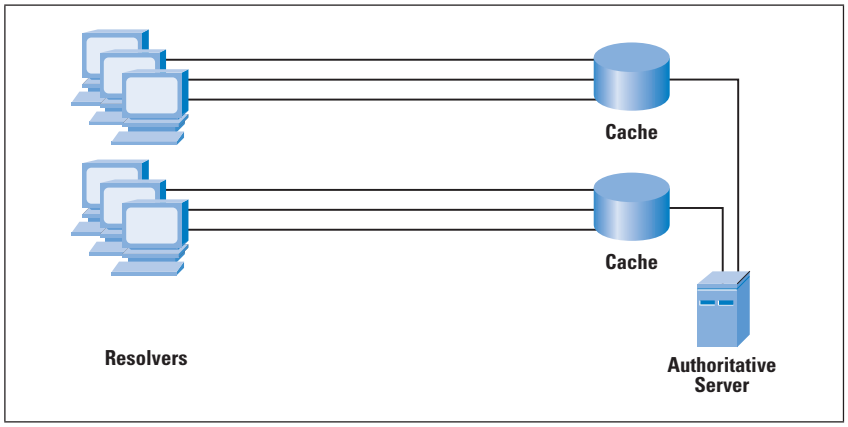
Traffic traces show a distribution of priming queries to 534 authoritative servers in the first 15 minutes of clearing the cache.

Figure 2: Dense Cache



We then added 9 new caches and redistributed the 140 stub resolvers among the 10 caches into a sparse cache mode (Figure 3) and restarted all the caches. In the first 15 minutes, the number of priming queries from each of the caches averaged 61, with a total of 622 unique priming queries for all caches. The number of "duplicate" queries between caches averaged 45. Although the number of queries to the authoritative servers was slightly higher, the results seem to indicate that there is a small but significant difference in each of the caches[8].

Figure 3: Sparse Cache



Conclusions

Reducing the size of the user population for each cache reduces the attack surface for the DNS overall because we have effectively compartmentalized the threat to a small number of nodes. Generally, restarting a cache for a small number of nodes is considered acceptable, whereas restarting a cache for 10,000 or 100,000 nodes would significantly affect operations.

Moving the cache closer to the resolver increases overall response time and may support better mobility of the node. If validation is also placed with the cache, it is possible to increase the confidence of validation because that information may not have to use DNS protocols to send validation data over untrusted, open networks.

The concept of supporting larger numbers of full DNS servers on more nodes raises concerns, but most systems these days have enough processing power and bandwidth to support this application. Administrative and management processes can be fully automated. Overall, this design complements other, protocol-based attempts to increase DNS integrity.

References

[1] "Preliminary Report on DNS Response Modification," 20 June 2008, **http://www.icann.org/en/committees/security/sac032.pdf**

[2] "DNS Performance and the Effectiveness of Caching," Jaeyeon Jung, Emil Sit, Hari Balakrishnan, and Robert Morris, *IEEE/ACM Transactions on Networking,* Volume 10, No. 5, pp. 589–603, October 2002.

[3] **https://www.dns-oarc.net/files/workshop-2006/Dickinson-Performance.pdf**

[4] "An analysis of Wide-Area Name Server Traffic: A Study of the Internet Domain Name System," Peter B. Danzig, Katia Obraczka, and Anant Kumar, ACM SIGCOMM *Computer Communications Review,* Volume 22, No. 4, pp. 281–292, 1992.

[5] "An Analysis of the Queries from Caching Servers to Root Servers, Tsuyoshi Toyono, NTT Laboratories, 2007 OARC Workshop,
https://www.dns-oarc.net/files/dnsops-2007/Toyono-Caching-analysis.pdf

[6] RootServer supplied statistics:
http://h.root-servers.org/
http://k.root-servers.org/index.html#stats
http://m.root-servers.org/

[7] **http://snad.ncsl.nist.gov/dnssec/mem_usage.html**
http://snad.ncsl.nist.gov/dnssec/bandwidth.html

[8] "Sharp Transition Towards Shared Vocabularies in Multi-agent Systems," Andrea Baronchelli, Maddalena Felici, Vittorio Loreto, Emanuele Caglioti, and Luc Steels, *Journal of Statistical Mechanics: Theory and Experiment,* 2006, P06014.
http://www.iop.org/EJ/abstract/1742-5468/2006/06/P06014

BILL MANNING has been in the network field since 1979, currently with the Keio University, Shonan Fujisawa Campus, and USC/ISI. He has been an IETF Working Group chair and RFC author, and he currently serves on numerous ICANN committees. He is part of the team that runs one of the Internet Root nameservers. E-mail: **bmanning@sfc.wide.ad.jp**

IEEE 802.21: Media-Independent Handover Services

by Esa Piri and Kostas Pentikousis, VTT Technical Research Centre of Finland

Popular mobile devices now ship with several integrated wired and wireless network interfaces. *Personal Digital Assistants* (PDAs) and smartphones, for example, are increasingly supporting communications through both cellular technologies and *Wireless LANs* (WLANs); laptops typically come with built-in Ethernet, Wi-Fi, and Bluetooth[1]. As multiaccess devices proliferate, we move closer to a network environment that is often referred to as "*beyond 3G*" (B3G). Key success factors for cellular *third-generation* (3G) communications include better cell capacities, increased data rates, transparent mobility within large geographical areas, and global reachability. For B3G, the next frontier lies beyond transparent mobile connections within the same access technology because users will expect to be globally reachable anytime, anywhere, and remain "*always best-connected*" (ABC)[2]. In order to select the best possible connectivity option (anytime, anywhere), mobile devices and access networks will have to work together in order to enable users to take full advantage of all available options.

The IEEE 802.21 working group (see `www.ieee802.org/21`) recently finalized the first standard for dealing with handovers in heterogeneous networks, also called *Media-Independent Handovers* (MIH)[3]. The standard is expected to allow mobile users (and operators) to take full advantage of overlapping and diverse access networks. It provides a framework for efficiently discovering networks in range and executing intelligent heterogeneous handovers, based on their respective capabilities and current link conditions. This article aims to serve as a primer for those interested in the IEEE 802.21 standard. After introducing the IEEE 802.21 reference model, we present the MIH services and provide illustrative use cases that highlight the benefits of employing the Media-Independent Handover Services standard in heterogeneous networks.

Mobile and Wireless

The widespread success of 3G technologies[4, 5] is evidenced by the rapid increase in the amount of data traffic over cellular networks in recent years. In Sweden, for example, the total amount of mobile data traffic leapt tenfold from just over 203 TB in 2006 to 2191 TB in 2007[6]. This trend is expected to continue unabated with the deployment of *High-Speed Packet Access* (HSPA) and *Long-Term Evolution* (LTE) in the coming years. Of course, the amount of traffic over cellular networks is only a proportion of the traffic that originates from or terminates at WLANs worldwide. Campuswide deployments of WLANs are becoming the norm in developed countries, and we even find citywide WLANs, as in the case of the city of Oulu, Finland (see `www.panoulu.net`).

Finally, many anticipate that mobile WiMAX[7] deployments will significantly affect telecommunications markets. In short, we are moving toward a far more heterogeneous network access environment than the one users and operators face today, with multiple overlapping mobile and wireless networks with diverse characteristics.

Multiaccess Devices in Heterogeneous Networks

As communication environments become more complex because of the diversity of network access technologies that support, for example, different access rates and *Quality of Service* (QoS) levels, users expect more from their wireless operator. Mobile devices, once featuring tiny screens, extremely limited processing and storage capacities, and narrowband connectivity[8], now pack capabilities that just a few years ago were typical of high-end laptops. This scenario has allowed users to increasingly depend on mobile devices for e-mail and *Instant Messaging* (IM), but also for making *Voice over IP* (VoIP) calls, listening to streaming Internet radio, and watching online videos.

With respect to user mobility patterns, campuswide Wi-Fi users typically spend most of their connection time attached to a small set of access points located within a small radius[9, 10]. This situation is not surprising, because Wi-Fi was originally designed and subsequently deployed mainly as an extension to wired infrastructures. In the future, however, we anticipate that multiaccess devices will employ different network interfaces to attach to different access networks, establishing multiple parallel connections over 3G/*Universal Mobile Telecommunications Service* (UMTS) and Wi-Fi, for example. With global reachability and ABC mechanisms in place, mobile devices will be able to selectively connect to different access networks depending on certain criteria. Keep in mind that from a conventional, IP-centered point of view, changing the *Point of Attachment* (PoA) calls for mobility management actions[11, 12, 13], although in practice there may be no physical mobility whatsoever.

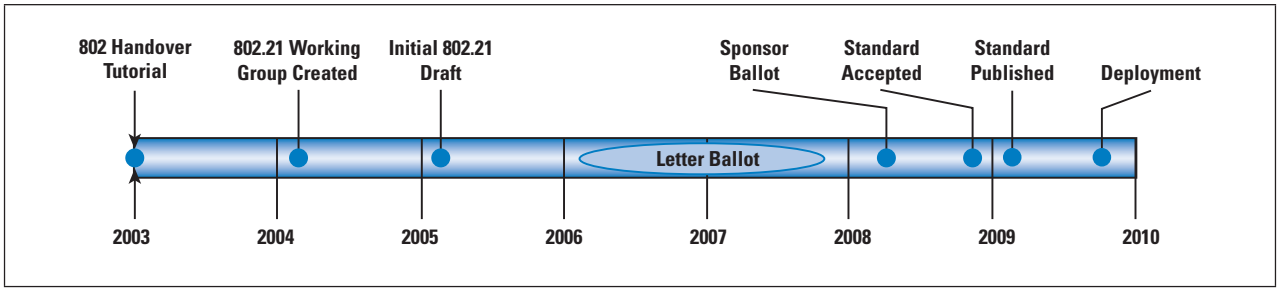
Given the diversity of networked applications running on mobile devices, knowledgeable network resource planning and operation is needed, in turn calling for a framework that allows users and their applications to state their network access preferences. This framework should also allow operators to steer terminal access patterns aiming at maximizing resource usage and increasing user satisfaction. For instance, podcasts can be downloaded only when connected to an uncongested WLAN, but web, map/navigation, and e-mail clients can use the cellular network or WLAN access on demand. Currently, this process can only be done manually: users need to be watchful for available access networks and choose which one to attach to based on very rudimentary information such as signal quality. If mobile nodes could collect timely and consistent information about the state of all available networks in range and were given the means to control their network connectivity, then a whole range of possibilities would become available.

In order to optimize the use of available network resources, mobile nodes need to be able to collect information about numerous heterogeneous networks in a generic and standardized way, irrespective of the underlying network access technology. The collected information, both dynamic and static, can then be used by handover decision-making processes, such as, say, mobility managers. Mobility managers can be enhanced versions of *Mobile IP* (MIP)[11, 12, 13], proprietary solutions, or other proposals stemming from recent research, such as [14]. Researchers in the area have proposed several cross-layer frameworks for enhancing the efficiency of handover decision makers (see [14, 15] and the references therein), but none of them has been formally standardized or is widely accepted so far. What is needed is a standard framework that can attract ample support from major vendors and operators, and can be deployed incrementally.

Introducing IEEE 802.21-2008

Figure 1 illustrates the progress toward the IEEE 802.21-2008 standard. The working group was initiated in 2004, and the latest draft version of the standard was accepted as a new standard by the IEEE-SA Standards Board in November 2008[3]. The standard was published in January 2009. It is anticipated that actual deployment of the standard will take place at the earliest in late 2009–2010.

Figure 1: Timeline of the IEEE 802.21-2008 Standardization Effort



IEEE 802.21-2008, also known as *Media-Independent Handover Services*, features a broad set of properties that meet the requirements of effective heterogeneous handovers. It allows for transparent service continuity during handovers by specifying mechanisms to gather and distribute information from various link types to a handover decision maker. The collected information comprises timely and consistent notifications about changes in link conditions and available access networks.

Note that the scope of IEEE 802.21-2008 is restricted to access technology-independent handovers. Intratechnology handovers, handover policies, security mechanisms, media-specific link layer enhancements to support IEEE 802.21-2008, and *Layer 3* (L3) and upper-layer enhancements are outside the scope of IEEE 802.21-2008. This article summarizes the salient points of [3], which henceforth is referred to as IEEE 802.21.

The IEEE 802.21 Reference Model

IEEE 802.21 facilitates a variety of handover methods, including both *hard handovers* and *soft handovers*. A hard handover, also known as "break-before-make" handover, typically implies an abrupt switch between two access points, base stations, or, generally speaking, PoAs. Soft handovers require the establishment of a connection with the target PoA while still routing traffic through the serving PoA. In soft ("make-before-break") handovers, mobile nodes remain briefly connected with two PoAs. Note, however, that depending on service requirements and application traffic patterns, hard handovers may often go unnoticed. For example, web browsing and audio/video streaming with prebuffering can be accommodated when handing over between different PoAs in the range of one network by employing mechanisms that allow transferring the node connection context from one PoA to another quickly.

The main design elements of IEEE 802.21 can be classified into three categories: a framework for enabling transparent service continuity while handing over between heterogeneous access technologies; a set of handover-enabling functions; and a set of *Service Access Points* (SAPs).

Transparent Service Continuity

IEEE 802.21 specifies a framework that enables transparent service continuity while a mobile node switches between heterogeneous access technologies. The consequences of a particular handover need to be communicated and considered early in the process and, clearly, before the handover execution. In soft handovers, it is crucial that service continuity, during and after the handover, is ensured without any user intervention. To this end, IEEE 802.21 specifies essential mechanisms to gather all necessary information required for an affiliation with a new access point before breaking up the currently used connection. Interactive applications, such as VoIP, are typically the most demanding in terms of handover delays, and high-quality VoIP calls can be served only by soft handovers. On the other hand, video streaming can accommodate hard handovers, as long as the vertical break-before-make handover delay does not exceed the application buffer interval delay. In the case of hard handovers, handover preparation signaling can initiate the connection context transfer from the serving PoA to the target PoA beforehand.

For instance, lack of the required level of QoS support or low available capacity in a candidate access network may lead the network selecting entity to prevent a planned handover. On the other hand, for example, increasing delay, jitter, or packet-loss rates in the currently serving network may degrade the perceived QoS throughout the network, or only for a particular application, triggering the mobility manager to start assessing the potential of candidate target access networks and subsequently initiate an IEEE 802.21-assisted handover.

IEEE 802.21 also allows the reception of dynamic information about the performance of the serving network and other networks in range. In other words, IEEE 802.21 provides methods for continuous monitoring of available access conditions. However, IEEE 802.21 does not specify any methods for collecting this dynamic information at the link layer.

Handover-Enabling Functions

IEEE 802.21 defines a set of handover-enabling functions, which are specified with respect to existing network elements in the protocol stack, and introduces a new logical entity called *Media-Independent Handover Function* (MIHF). The MIHF logically resides between the link layer and the network layer. It provides, among others, abstracted services to entities residing at the network layer and above, called *MIH Users* (MIHUs). MIHUs are anticipated to make handover and link-selection decisions based on their internal policies, context, and the information received from the MIHF. To this end, the primary role of the MIHF is to assist in handovers and handover decision making by providing all necessary information to the network selector or mobility management entities. The latter are responsible for handover decisions regardless of the entity position in the network. The MIHF is not meant to make any decisions with respect to network selection.

Service Access Points

SAPs with associated primitives between the MIHF and MIHUs (MIH_SAP) give MIHUs access to the following services that the MIHF provides:

- The *Media-Independent Event Service* (MIES) provides event reporting about, for example, dynamic changes in link conditions, link status, and link quality. Events can be both local and remote. Remote events are obtained from a peer MIHF entity.
- The *Media-Independent Command Service* (MICS) enables MIHUs to manage and control the parameters related to link behavior and handovers. MICS provides a set of commands for accomplishing that, as we will see later in this article. Commands can be both local and remote. The information obtained with MICS is dynamic.
- The *Media-Independent Information Service* (MIIS) allows MIHUs to receive static information about the characteristics and services of the serving network and other available networks in range. This information can be used to assist in making a decision about which handover target to choose and to make preliminary preparations for a handover.

Figure 2 illustrates the general reference model of IEEE 802.21. The scope of IEEE 802.21 includes only the operation of MIHF and the primitives associated with the interfaces between MIHF and other entities. A single media-independent interface between MIHF and MIHU (MIH_SAP) is sufficient.

On the other hand, there is a need for defining a separate technology-dependent interface, which is specific to the corresponding media type supported, between the MIHF and the lower layers (MIH_LINK_SAP).

The primitives associated with the MIH_LINK_SAP enable MIHF to receive timely and consistent link information and control link operation during handovers. For example, the currently supported link layers include wired and wireless media types from the IEEE family of standards (for example, 802.3, 802.11, 802.15, and 802.16), as well as those defined by the *Third-Generation Partnership Project* (3GPP) and *Third-Generation Partnership Project 2* (3GPP2). Besides these, IEEE 802.21 specifies a media-independent SAP (MIH_NET_SAP), which provides transport services for Layer 2 (L2) and Layer 3 (L3) MIH message exchange with remote MIHFs. Functions over the LLC_SAP are not specified in IEEE 802.21.

Figure 2: The IEEE 802.21-2008 Reference Model

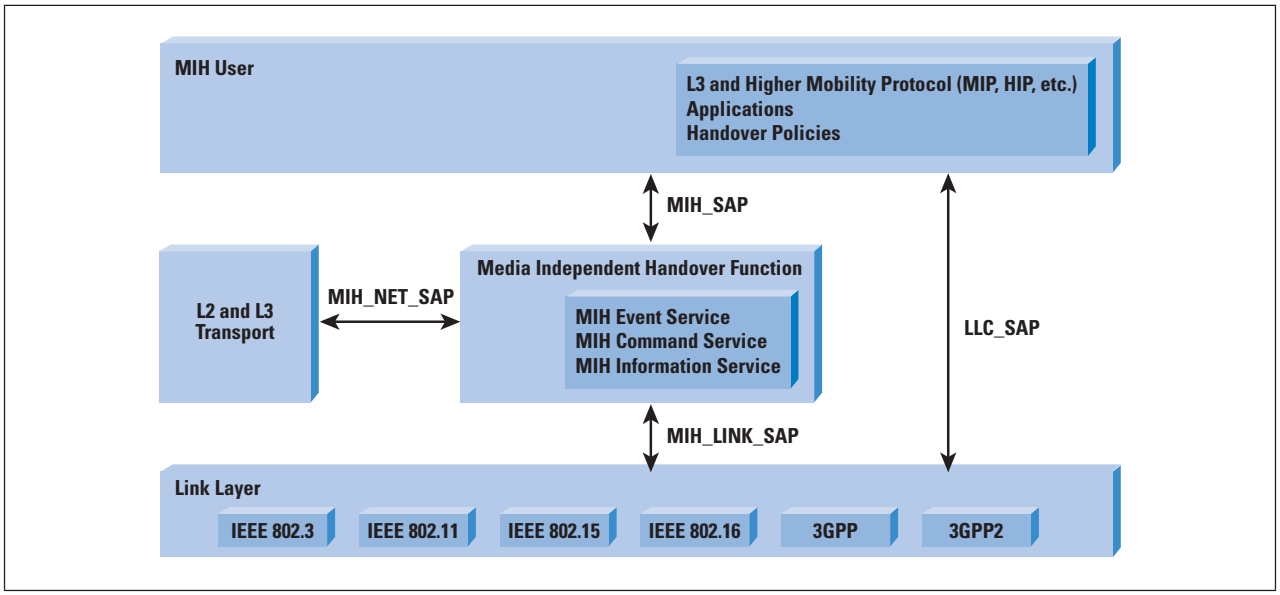
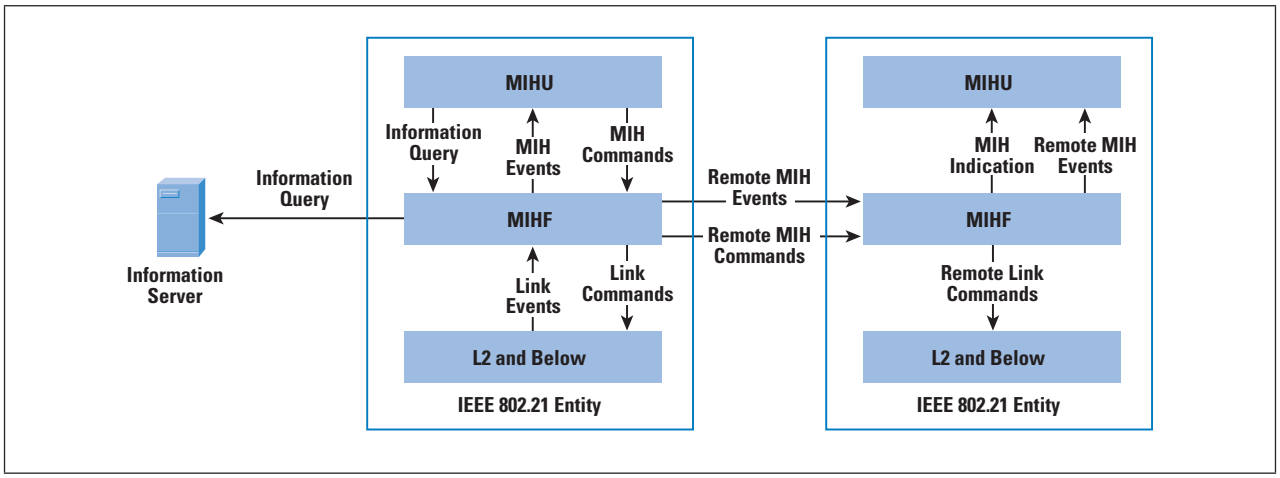


Figure 3 presents the messages directions of each MIHF service class, including both local and remote events and commands. The MIHF can subscribe to particular sets of events from a peer MIHF. Remote commands are initiated by local MIHUs and are conveyed to the peer MIHF through the local MIHF. Finally, MIIS information can be obtained through queries to the local database and to remote Information Servers.

Figure 3: MIHF Services



IEEE 802.21 Illustrated

Figure 4 illustrates an example topology where different wireless networks overlap. Imagine that the multiaccess mobile device user watches a high-bitrate IPTV channel as she moves in this area. Three wireless access technologies are considered in this example: Wi-Fi (IEEE 802.11), WiMAX (IEEE 802.16), and 3G/UMTS (3GPP). In this example, we assume that all networks and the mobile device are IEEE 802.21-compatible and that the Wi-Fi area is covered by several 802.11 PoAs, as would be the case in a campus- or citywide deployment.

Figure 4: Example Topology with Heterogeneous Overlapping Wireless Access Networks

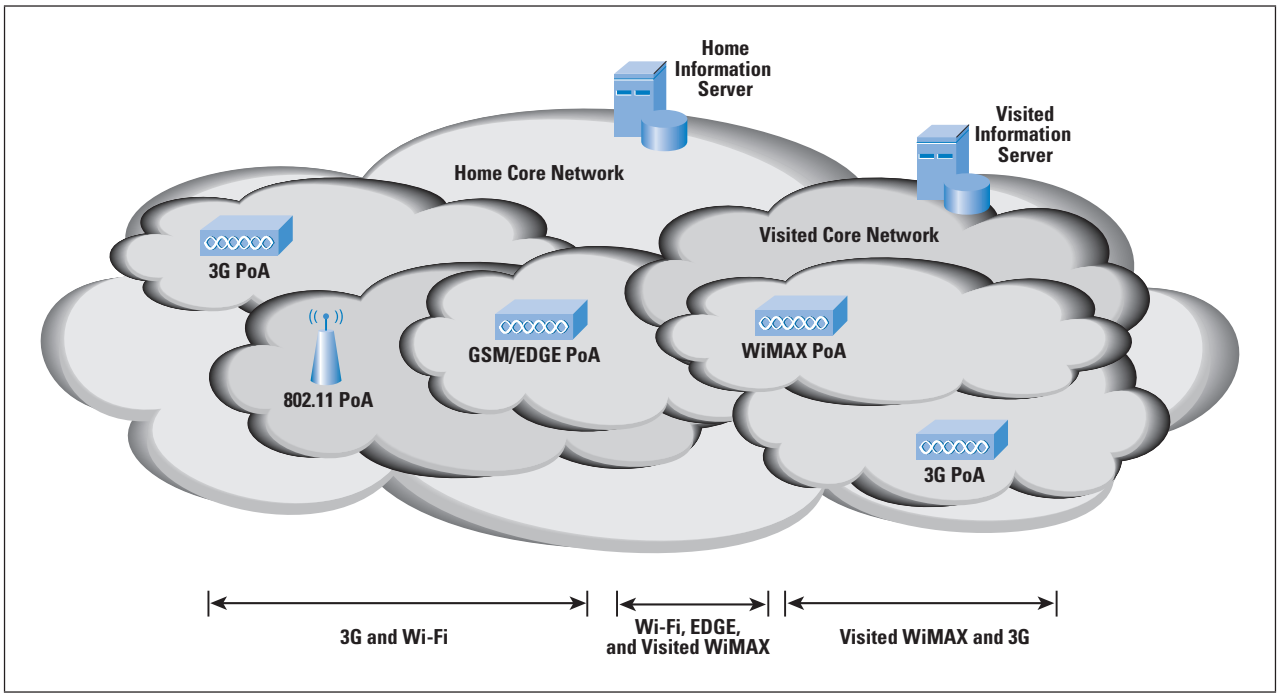
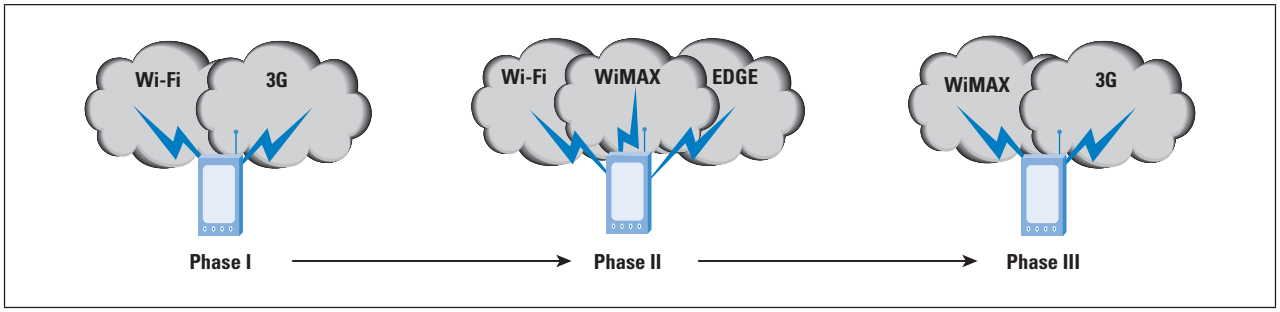


Figure 5 illustrates the network access environment as perceived by a mobile device in the area. The figure depicts three snapshots, indicating the overlapping networks in range at different locations. In order to deliver the IPTV stream transparently, for each of the available access networks we need to consider their effective available bandwidth, the associated cost per traffic unit, the terminal speed, the cell coverage area, the level of QoS support it can provide, and so on. Using information made available through the MIHF, we can determine which should be the next target access network.

Figure 5: Example Network Environment in Different Locations



In Phase I, the mobile node has two network access options. It can use a free and open Wi-Fi network or connect to the cellular operator's 3G/UMTS network. Note that opting to use the latter may, for instance, depend on the charging scheme of the operator. If subscribers pay based on traffic volume, one would assume that the free Wi-Fi network is a better option. On the other hand, as flat-rate plans become more popular, 3G may be a better option with its extended coverage and QoS guarantees. The IEEE 802.21 MIIS can provide this type of information, allowing for automation in dynamic access selection.

In Phase II, as the user moves, the device goes through a cellular technology handover from 3G/UMTS to *Enhanced Data rates for GSM Evolution* (EDGE)[8]. At the same place, the public Wi-Fi network is still available and a new WiMAX network has just been detected. Assume that EDGE is not sufficient for delivering the IPTV stream. If in Phase I the network selection process opted for using the cellular network, then in Phase II the client application will experience significant degradation in service if it continues to use the EDGE access network. A vertical handover to the Wi-Fi or the WiMAX network should be considered. In contrast, if the mobile node first chose to stream the IPTV channel over the Wi-Fi access network, then it may need to reassess the situation based on events and link parameter reports using MIES and MICS, as we explain in the following sections. For example, an information query can reveal whether the WiMAX network is operated by a partner *Internet Service Provider* (ISP), and what the roaming cost would be.

Finally, in Phase III, the coverage area of the public Wi-Fi network ends. Through IEEE 802.21 services we find out that the only available networks are the roaming partner WiMAX and the home cellular network that is now offering 3G service.

The environment with several overlapping networks described previously and illustrated in Figures 4 and 5 is already a reality today in many places, and it is widely anticipated to be prevalent in the future. Next, we examine the three services defined by IEEE 802.21, namely MIES, MICS, and MIIS.

Media-Independent Event Service

Events indicate or predict changes in the state and transmission behavior of physical, data link, and logical link layers. In general, events are triggers for initiating candidate network discovery and handover procedures. The events defined in IEEE 802.21 are categorized as either *Link Events* or *MIH Events,* depending on their origin. Link events emanate from the link layers, whereas MIH events emanate from the MIHF and can be both remote and local. Local events propagate from lower layers to upper layers through the MIHF. Remote events occur at the protocol stack of another network entity and are transmitted from a peer MIHF to the local MIHF, as illustrated in Figure 3.

The *Media-Independent Event Service* (MIES) currently supports five types of events: MAC and PHY State Change events, Link Parameter events, Predictive events, Link Handover events, and Link Transmission events. A short introduction to the event types and corresponding events follows.

MAC and PHY State Change events correspond to state changes in MAC and *physical* (PHY) layers. The most characteristic events in this category are *Link_Up* and *Link_Down* events, which are generated when a Layer 2 connection with an access point is established or is torn down, respectively. Another event, called *Link_Detected,* indicates that a PoA has been detected but no affiliation is established yet.

Link Parameter events relate to changes in Layer 2 parameters. A *Link_Parameters_Report* can be sent when a MIHU has set thresholds for certain parameters. For example, a MIHU can set thresholds for the *Received Signal Strength Indicator* (RSSI) on IEEE 802.11 links, so that when a threshold is crossed proper action can be taken. A *Link_Parameters_Report* is also used for issuing periodical notifications about link conditions. Based on Link Parameter events, a MIHU can initiate the handover candidate discovery process, or trigger applications to adapt to changing link conditions.

Predictive events inform about the probability of dramatic (negative) changes in link characteristics in the near future. For example, if strong decay in signal strength is observed, this decay may indicate imminent loss of link connectivity. Predictive events may include temporal information about when the actual event is expected to occur and what its presumed likelihood is. A *Link_Going_Down* event, for instance, may trigger a MIHU to consider possibilities for handing over to other available networks in range.

Link Handover events indicate the occurrence of Layer 2 handovers. The *Link_Handover_Imminent* event serves as a notification for an imminent handover, whereas a *Link_Handover_Complete* event reports the successful change of PoA. These events emanate from the link layer and are based solely on local Layer 2 information.

Link Transmission events show the transmission status of individual higher-layer *Protocol Data Units* (PDUs) at the link layer. Upper layers can, for example, adapt to data loss during a handover by improving buffer management based on Link Transmission events. These events may allow future upper-layer implementations to identify lost packets and recover without waiting for the expiration of retransmission timers.

Currently, for example, in the case of an ongoing session over TCP, the occurrence of a handover may have dramatic effects in performance. With IEEE 802.21, MIHUs can be informed about individual packets that have already been delivered to the sending buffer of the MAC layer but were not successfully transmitted before the handover occurred. In other words, the MAC layer outgoing buffer may contain TCP segments that cannot be delivered through the wireless network to the peer at the other end of the TCP connection. These segments were not successfully delivered from the local *Automatic Repeat-reQuest* (ARQ) module over the first hop, but are still buffered and cannot be transmitted because there is no link connectivity. In this case, TCP could use the information from Link Transmission events that identifies which packets need to be resent through the new access network, as illustrated in Figure 6 for packet numbers 1 and 2. Note, however, that IEEE 802.21 does not define any identifier for reliable packet identification, only the size of the packet ID (2 bytes), and it is up to the implementer to determine how different messages will be locally identified.

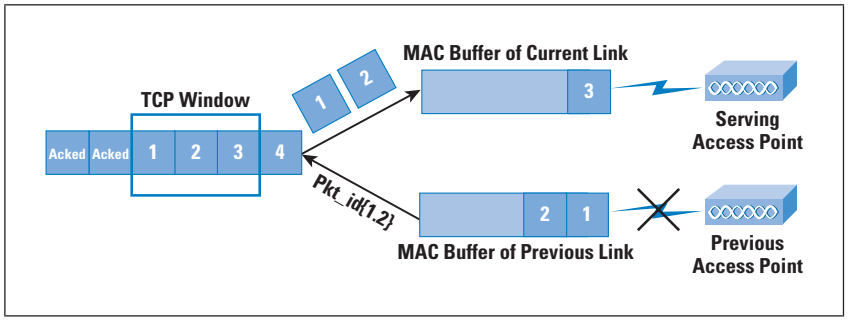


Figure 6: Link Transmission Event Indicating Undelivered Packets

Media-Independent Command Service

The *Media-Independent Command Service* (MICS) enables higher layers to control the stream of events originating from lower layers. Commands can originate from MIHUs (MIH commands) or from the MIHF (Link commands) and the destination can be the MIHF or any lower layer, respectively, as shown in Figure 3. The responses to Link commands are sent to MIHUs as indications. MIHUs can use command services to determine the status of different links in a uniform way, and control each interface accordingly, aiming for optimal connectivity. MICS defines the following set of commands that enable MIHUs to configure, control, and get information from the lower layers:

- *MIH commands* can be directed to lower layers residing at both local and remote MIHF entities. They originate from the upper layers and are directed to the MIHF. Similarly with MIH events, MIH commands can be both remote and local. MIH commands are typically used for network selection and handover management because they allow upper layers to initialize, prepare for, and execute handovers. MIH commands are also used to configure custom thresholds for link parameters. As mentioned previously, when set thresholds are crossed, MIHUs get the corresponding notifications through Link Parameter events.
- *Link commands* originate from the MIHF and are sent to lower layers in order to control their operation. Link commands can be issued only locally. Nevertheless, Link commands can be executed on behalf of local MIHUs, which could act on information received from a remote peer. Link commands are often initiated by MIHUs. For example, an MIHU can issue the *MIH_Get_Link_Parameters* MIH command, which when received by the local MIHF will lead to the generation of a remote *Link_Get_Parameters* Link command, as shown in Figure 3. This way, the MIHF can acquire the current parameter values of active link(s) for MIHU, and then deliver this information to the requesting MIHU. Note that MICS provides dynamic information about different link parameters, in contrast with MIIS, described next, which can report only static information.

Media-Independent Information Service

The *Media-Independent Information Service* (MIIS) facilitates handovers through a unified set of mechanisms that the MIHF can use to discover and obtain static (or rarely changing) information about networks in the vicinity of a multiaccess node. In other words, MIIS allows mobile nodes to check for available networks in range while using their currently active access network. MIIS information exchange occurs at the link layer (Layer 2) or network layer (Layer 3), so that all necessary information related to link layer or higher-layer services is collected before a mobile node authenticates with a new PoA.

MIIS defines a set of *Information Elements* (IEs) that are indispensable for network selection, classified into three groups: General Information and Access Network-Specific Information; PoA-Specific Information; and Other Information, which includes vendor- and network-specific details. The types of information handled by MIIS are solely related to handover decisions and conformance to the affiliation with the new PoA. Information relevant for assessing candidate networks by the handover machinery includes connection establishment details, such as PoA address and location; which security mechanisms are supported in a given access network; and what QoS guarantees can be provided.

General Information Elements and *Access Network-Specific Information Elements* give a general overview of neighboring networks. Information Elements may include, for instance, a list of available networks and their associated operators, roaming agreements and costs, and security and QoS support. For instance, user policies, defined at higher layers, may dictate that if a given access network operator charges users based on their traffic volume, then the network selector entity should not consider the corresponding access when a high-bitrate service, such as IPTV, is active.

PoA-Specific Information Elements refer to each PoA available in the access network and report PoA location and addressing information, supported data rates, PHY and MAC layer types, and channel parameters that can optimize link layer connectivity. Some additional information related to higher-layer services and individual capabilities of particular PoAs may be included as well. For instance, an advanced mobility manager on the mobile node can use the information about the geographical position of a PoA and compare it with the current or expected node location based on its mobility patterns. With careful planning and by taking advantage of this information, mobile nodes may be able to reduce the number of handovers and optimize the use of network resources.

MIIS provides mechanisms for issuing and responding to queries for Information Elements. Such information may reside in a separate server or in a local information database at the mobile node (see Figure 3). An MIHF could have access to an information server in its IEEE 802.21-enabled *Point-of-Service* (PoS) range from which it can obtain information regarding the home PoS and possibly other PoSs, such as those of roaming partners. If the home information server is not able to provide any information regarding the visited network, an MIIS query can be directed to the peer MIHF, residing in the visited PoS, which can access the visited PoS information server. Information queries can often be answered locally, based on information gathered from previous queries and by preprovisioning, for example, from the information server.

Information Elements and their relationships are captured in an Information Service schema which, in turn, defines the information structure. IEEE 802.21 specifies that information that is to be presented across different technologies should be in a standardized, common, and open format, such as XML or *Type Length Value* (TLV).

Service Management

In order to use and provide MIHF services, MIHF entities need to be configured appropriately. IEEE 802.21 defines three service management functions: MIH capability discovery, MIH registration, and MIH event subscription.

MIHF may discover other MIHF entities and their capabilities using the MIH capability discovery procedure. Depending on the information obtained from this procedure, the local MIHF can determine which peer MIHFs it should register with. The MIH capability discovery function uses the MIH protocol (introduced in the following section) at Layer 2 or Layer 3, and media-specific Layer 2 broadcast messages are allowed. For example, an MIHF can listen to media-specific broadcast messages, such as IEEE 802.11 beacons, or media-independent Layer 2 *MIH_Capability_Discover* broadcast messages, because an MIHF entity residing in the network may announce its existence and capabilities periodically. MIHF can also send *MIH_Capability_Discover* request messages using multicast or unicast to detect peer MIHFs in a solicited way. For instance, MIHF can send a request by unicast for obtaining the capabilities of a specific IEEE 802.21 network entity. In this case, only the IEEE 802.21 network entity addressed should respond to these request messages.

MIH registration is a symmetric procedure by which two peer MIHFs authenticate and can then communicate with each other in a more trusted manner. After MIH registration is completed, the two peer MIHF entities can symmetrically request services from their registered peer. Note that MIH registration is not necessary for obtaining some level of support from a peer MIHF. However, by registering and authenticating, peer MIHFs typically will get access to much more extensive information. That is, although the MIHF residing on the mobile node may be able to access information services from the network-side MIHFs without registration and authentication, the available information may be only a subset of that provided after authenticating.

Finally, MIH event subscription enables MIHUs to subscribe to a particular set of events provided by MIES from the local or peer MIHF. Event subscription from a peer MIHF requires registration and knowledge about its capabilities. The subscription contains only the list of events the MIHU is interested in. Note that event sources may not be necessarily capable of providing all events that the subscriber is interested in subscribing to. Each subscription request is matched by a confirmation message from the event source indicating the events approved for subscription.

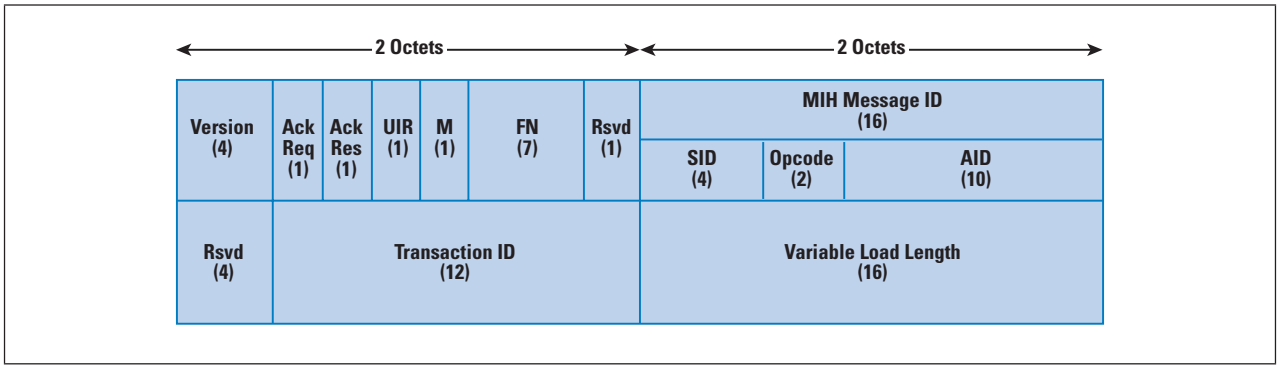
Media-Independent Handover Protocol

The *Media-Independent Handover Protocol* (MIHP) specifies the rules and services for unified communication between peer MIHFs. The protocol defines the message format, header, and encoding format and is meant to be used solely for communicating with peer MIHF entities. For internal communication no particular encoding is dictated.

MIH protocol messages can be carried over Layer 2 management frames, Layer 2 data frames, or over Layer 3/IP transport. Note that cellular technologies do not provide Layer 2 transport without changes in their protocol stack.

The MIH protocol messages, or frames, comprise a header part and a TLV-encoded payload part. The MIHF frame header consists of eight octets. Figure 7 illustrates the MIH protocol header indicating the corresponding bit length for each field in parentheses.

Figure 7: MIH Protocol Header



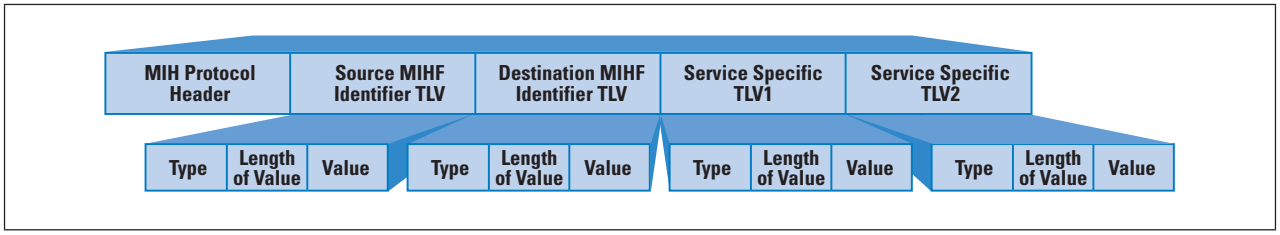
The *Version* field in the MIH frame header specifies the version of the MIH protocol used. The two *Ack* fields are for acknowledgement purposes and are discussed later in the article. The *Unauthenticated Information Request* (UIR) flag indicates that the response message may be sent with a limited length because of the nature of unauthenticated message exchange. Recall that when an MIHF issues requests without registering first with its peer, it may receive less information than if it had registered earlier. If this flag is set, then the information included in the response message may not reflect the complete information available to registered MIHFs. The *More Fragments* (M) and *Fragment Number* (FN) fields are used in message fragmentation.

The *MIH Message ID* field comprises three subfields. The *Service Identifier* (SID) field indicates the MIHF service class (MIES, MICS, MIIS, or Service Management) that this message belongs to. The *Operation code* (Opcode) specifies whether the message is a request, response, or indication. The *Action Identifier* (AID) is related with and scoped by the SID. For instance, if the SID indicates MIES, AID points to the actual event type. The *Variable Load Length* field contains the total length of the variable, TLV-encoded payload carried by this message frame.

The MIH protocol messages use the *Transaction ID* and *MIHF ID* fields as identifiers, but only the former is included in the header. The Transaction ID field is an identifier that helps to match each request, response, or indication message with its acknowledgement.

The payload part contains service-specific messages encoded in TLV format. The first two TLVs in the payload part (not shown in Figure 7) should be the *Source Identifier* and *Destination Identifier,* which are both the same data type as the MIHF ID. Every MIHF must have a unique MIHF ID, which may be assigned to it at configuration time. The MIHF ID shall be invariant and could be, for example, a *Fully Qualified Domain Name* (FQDN) or *Network Access Identifier* (NAI). The MIHF ID is used during the MIH registration phase and is appended to the payload part of every message requiring endpoint identification. In broadcast messages, the Destination Identifier TLV is defined as zero length. Figure 8 shows the message structure consisting of the MIH Protocol header, source and destination identifiers, and service-specific TLVs. In TLV encoding, the Type field (1 octet) denotes the parameter type, the Length field (variable octets) indicates the length of the Value field, and the Value field (variable octets) carries the actual value of the parameter.

Figure 8: MIH Protocol Frame Structure



Acknowledging MIH messages is not mandatory. Still, the MIH protocol does support the use of acknowledgements to ensure reliable message exchange. The sender MIHF can set the *ACK-Req* field to instruct the receiver to return an acknowledgement with *ACK-Rsp* bit set. The *MIH Message ID* and *Transaction ID* must be the same in the request message and its acknowledgement. An acknowledgement message may carry no payload. Note, however, that despite employing these two ID fields, the MIH protocol does not specify any further mechanisms for reliable authentication or shielding message exchanges from third parties.

MIH Communication Model

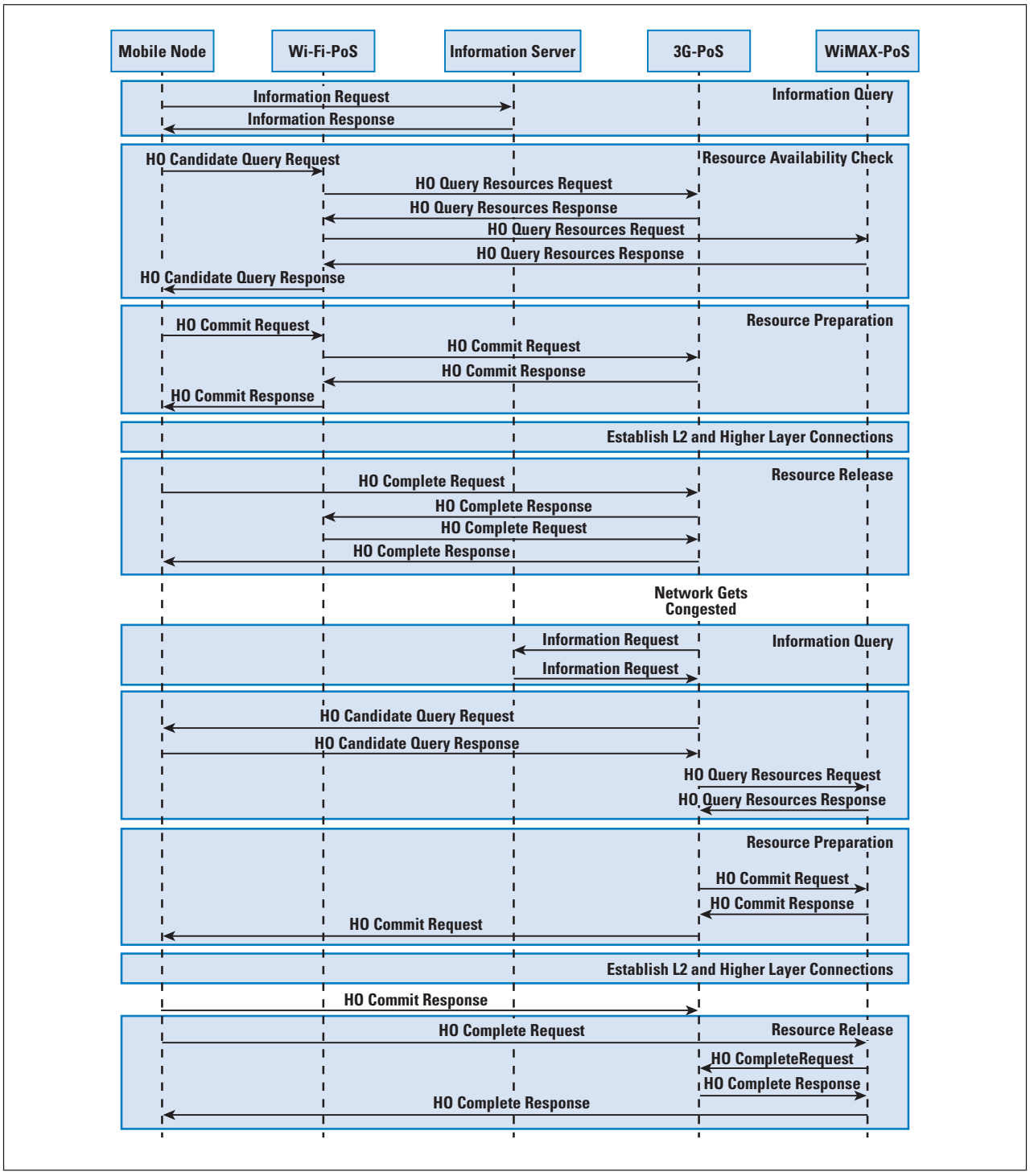
The MIHF communication model specifies different MIHF roles and their communication relationships, such as supported transport mechanisms and service classes. The assigned MIHF roles depend on their location in the network. For example, an MIHF on a mobile node can communicate directly with network-side entities called *MIH PoSs* using Layer 2 or Layer 3 communication. MIH PoSs may include the serving PoA or candidate PoAs. Network-side MIHFs can communicate with each other at Layer 3 or above using the MIH protocol, introduced in the previous section.

Let us revisit the example use case of IEEE 802.21 illustrated in Figures 4 and 5. Figure 9 presents the IEEE 802.21 message exchanges in mobile- and network-initiated handover procedures in the case where the mobile node hands over from a Wi-Fi to the 3G cellular network (between Phase II and Phase III in Figure 5) and then hands over to a WiMAX network (Phase III in Figure 5). First, during the discovery of handover candidate PoAs, the mobile node MIHF employs MIIS to gather static information about the surrounding networks. The request is issued over the currently used Wi-Fi access. This information is obtained from the information server that may reside in a different network than the one currently in use.

After receiving the response to its Information Request, the mobile node initiates the handover process by querying about the availability of resources in the networks it is interested in. These requests are sent through the serving PoS (*Wi-Fi-PoS* in Figure 9), which disseminates the requests to the MIH PoSs of the candidate networks (*3G-PoS* and *WiMAX-PoS* in Figure 9). The response indicating the capabilities of the two candidate networks is returned to the mobile node MIHF from the serving PoS. After receiving this information, an MIHU on the mobile node decides which network to hand over to, based on policies and the output of its network selection algorithms. Then a *Handover Commit Request* message is sent, and after the candidate network has made its final commitment for the handover (and the appropriate resources are reserved successfully), the mobile node establishes a Layer 2 connection with the PoA in the area of the candidate PoS, that is, the 3G-PoS in our example case. Following this successful intertechnology handover, the resources used in the previous link can optionally be released. In the case where no resources are explicitly reserved, this step is skipped.

As we progress in the timeline of our example case, the network-side MIHU initiates a handover to the WiMAX network. This handover could be, for example, the result of observing congestion in the cellular network that indicates that a new PoS should be found for the mobile node. The serving PoS (3G-PoS) collects information about networks in the range of the mobile node from the Information Server. Upon determining that a suitable WiMAX candidate network that can serve the mobile node exists, the 3G-PoS triggers a network-initiated handover. First, the serving PoS requests permission from the mobile node to proceed with the handover. If the mobile node does not object, the serving PoS proceeds with the rest of the handover procedure, which is similar to the mobile-initiated handover described previously except that it is handled by a network entity.

Figure 9: IEEE 802.21-Assisted Handover Message Sequence Diagram



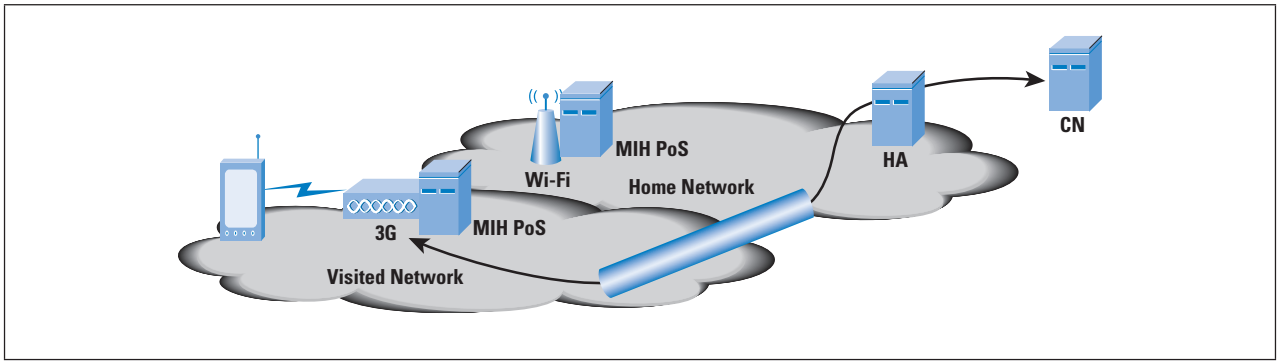
Handover Execution

As illustrated in the example, the handover decision and target assessment constitute a multiphase process where the assistance of IEEE 802.21 is essential. However, the actual handover execution is outside the scope of the standard. This section briefly describes how handovers can be carried out by MIP with the cooperation of IEEE 802.21. After choosing the target network by capitalizing on the IEEE 802.21 services, the mobile node establishes a new connection with the handover target network while still routing traffic through the currently serving network. The mobile node obtains a *Care-of Address* (CoA) for this new link from the IP address space of the target network. The CoA is an IP address assigned to the new link of the mobile node and is used while connected to the visiting network[11]. With MIPv4, the CoA is provided by a *Foreign Agent* (FA) in the visited network, which also acts as a router for the mobile node[12]. With MIPv6, the Foreign Agent is not needed[13] and the CoA is obtained directly, say, for example, from a *Dynamic Host Configuration Protocol* (DHCP) sever. The mobile node can obtain the IP address of the DHCP server in the target network through the IEEE 802.21MIIS.

In MIP, each mobile node has a *Home Agent* (HA), which routes the traffic of the mobile node. After successfully affiliating with a PoA in the target network, the mobile node notifies the Home Agent of the CoA by performing a binding update. In a bidirectional tunnel mode, the Home Agent establishes an IP-IP tunnel between the Home Agent and the Foreign Agent (MIPv4) or the Home Agent and the mobile node CoA (MIPv6). This mode does not require any binding updates on the *Correspondent Node* (CN). In other modes, either the uplink traffic of the mobile node is sent directly to the Correspondent Node using the CoA as source address, or all bidirectional communication between the Correspondent Node and the mobile node uses the CoA only. In the first case, traffic from the Correspondent Node to the mobile node travels through the Home Agent, but in the latter case there is no need for the Home Agent detour. However, these modes need address binding at the Correspondent Node and are in practice less frequently used than the bidirectional tunnel mode.

Figure 10 illustrates a situation where a link with the Wi-Fi PoA is broken down by the mobile node and the IPv6 traffic between the Correspondent Node and the mobile node, now employing IEEE 802.21-enabled 3G network, travels through the tunnel between Home Agent and the mobile node.

Figure 10: Mobile IPv6 Tunnel



Layer 3 handover executions based on RFC 3344[12] and RFC 3775[13] may often exceed the typical handover delay budgets, thus introducing gaps in connectivity that are perceptible at the application layer. Recent standardization efforts have focused on decreasing handover delays by enhancing MIP so that it can provide for transparent mobility management for both IPv4[16] and IPv6[17, 18]. The proposed enhancements either reduce the amount of signaling or allow the mobile node to configure the new Layer 3 connection before reassociating with the new network. In this context, IEEE 802.21 can provide the essential information for preestablishing the connection based on media-independent Layer 2 link detection events as well as static address information from the target network.

Summary and Outlook

We presented an overview of the IEEE 802.21 Media-Independent Handover Services standard. We anticipate that its adoption in the near future will allow for better network resource usage and permit multiaccess devices to select the network access best suited for their communication needs. After motivating the needs for a standard to cope with heterogeneous network handovers, we introduced the IEEE 802.21 Reference Model and the MIH Services. We briefly presented the MIH Protocol, although a more thorough description calls for a separate overview article. Finally, we illustrated network operation when IEEE 802.21 is adopted using example use cases featuring both network- and terminal-initiated intertechnology (or vertical) handovers.

We expect that in the future, when IEEE 802.21-2008 is widely deployed, there will be significant efforts to further amend and extend it in order to provide for even better services. In fact, because security mechanisms are outside the scope of the base IEEE 802.21 standard, the work on defining a security-related extension to IEEE 802.21 (IEEE P802.21a) has already begun. Moreover, another amendment (IEEE P802.21b) that deals with handovers with downlink-only technologies, such as *Digital Video Broadcasting* (DVB), has also been introduced (see `www.ieee802.org/21` for more information about the amendments). Nevertheless, it remains uncertain whether vendors will stand by this promising standard and incorporate it in future products and solutions.

References

[1] T. Sridhar, "Wi-Fi, Bluetooth and WiMAX," *The Internet Protocol Journal,* Volume 11, No. 4, December 2008.

[2] E. Gustafsson and A. Jonsson, "Always Best Connected," *IEEE Wireless Communications,* Volume 10, No. 1, February 2003.

[3] IEEE Std 802.21-2008, *IEEE Standard for Local and Metropolitan Area Networks—Part 21: Media Independent Handover Services,* IEEE, January 2009.

[4] H. Kaaranen, S. Naghian, L. Laitinen, A. Ahtiainen, and V. Niemi, *UMTS Networks: Architecture, Mobility and Services,* 2nd Edition, John Wiley & Sons, 2005.

[5] V. Vanghi, A. Damnjanovic, and B. Vojcic, *The cdma2000 System for Mobile Communications: 3G Wireless Evolution,* Prentice Hall, 2004.

[6] Y. Mälarstig, O. Holmström, and P. Davidsson, *Svensk telemarknad 2007,* PTS-ER-2008:15, ISSN 1650-9862, June 2008.

[7] J. Pinola and K. Pentikousis, "Mobile WiMAX," *The Internet Protocol Journal,* Volume 11, No. 2, June 2008.

[8] K. Pentikousis, "Wireless Data Networks," *The Internet Protocol Journal,* Volume 8, No. 1, March 2005.

[9] M. Balazinska and P. Castro, "Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network," *Proc. First International Conference on Mobile Systems, Applications, and Services (MobiSys),* San Francisco, California, USA, May 2003, pp. 303–316.

[10] T. Henderson, D. Kotz, and I. Abyzov, "The Changing Usage of a Mature Campus-wide Wireless Network," *Computer Networks,* Volume 52, No. 14, October 2008, pp. 2690–2712.

[11] W. Stallings, "Mobile IP," *The Internet Protocol Journal,* Volume 4, No. 2, June 2001.

[12] C. Perkins (Ed.), "IP Mobility Support for IPv4," RFC 3344, August 2002.

[13] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775, June 2004.

[14] K. Pentikousis, R. Agüero, J. Gebert, J. A. Galache, O. Blume, and P. Pääkkönen, "The Ambient Networks Heterogeneous Access Selection Architecture," *Proc. First Ambient Networks Workshop on Mobility, Multiaccess, and Network Management* (M2NM), Sydney, Australia, October 2007, pp. 49–54.

[15] J. Mäkelä and K. Pentikousis, "Trigger Management Mechanisms," *Proc. Second International Symposium on Wireless Pervasive Computing* (ISWPC), San Juan, Puerto Rico, February 2007, pp. 378–383.

[16] K. El Malki (Ed.), "Low Latency Handoffs in Mobile IPv4," RFC 4881, June 2007.

[17] H. Soliman, C Castelluccia, K. El Malki, and L. Bellier, "Hierarchical Mobile IPv6 Mobility Management," RFC 4140, August 2005.

[18] R. Koodli (Ed.), "Mobile IPv6 Fast Handovers," RFC 4068, July 2005.

ESA PIRI received his M.Sc. from the University of Oulu, Oulu, Finland, in 2008. During his studies, he specialized in information networks systems and wrote his Master's thesis on mobility management issues in heterogeneous networks. Currently he is working as a Research Scientist at VTT Technical Research Centre of Finland in Oulu, Finland. He can be contacted by e-mail at: **esa.piri@vtt.fi**

KOSTAS PENTIKOUSIS studied computer science at Aristotle University of Thessaloniki, Greece (B.Sc. 1996, summa cum laude) and State University of New York at Stony Brook, USA (M.Sc. 2000, Ph.D. 2004). He has published internationally in several areas, including mobile computing; transport protocols; applications; network traffic measurements and analysis; and simulation and modeling. Dr. Pentikousis is a Senior Research Scientist at VTT Technical Research Centre of Finland. Visit **http://ipv6.willab.fi/kostas** for more information and contact details.

Book Review

Geeks Bearing Gifts

Geeks Bearing Gifts v1.1: How the computer world got this way, by Ted Nelson, ISBN: 978-0-578-00438-9, Published by Mindful Press, 2009, distributed through Lulu.Com, `http://www.lulu.com`

In a short but interesting book, computer pioneer Ted Nelson takes a very broad look at the origins and evolution of many of the basic ideas that underpin today's computer industry. The emphasis is on concepts and technologies rather than the success of individuals, the companies they founded, and the shape of the computer industry. This approach differentiates the book from other accounts, such as Robert X. Cringley's *Accidental Empires* and Martin Campbell-Kelly's *From Airline Reservations to Sonic the Hedgehog*.

Although the book is suitable for a fairly broad readership, an appreciation of the current makeup of the industry is helpful in understanding the significance of some of Nelson's ideas.

Organization

Geeks Bearing Gifts is divided into 60 short chapters, arranged in chronological order from the time the ideas originated, rather than when they appeared in fully developed form (indeed many are still developing). In the initial chapters Nelson covers topics such as language, alphabets, and encryption before moving on to examine the origins of computing. He then examines the contribution of pioneers from both inside and outside the United States, giving more credibility to contributors from outside of the United States than is normal.

As would be expected, Nelson deals in some detail with the topic of information presentation, in particular the origins of hypertext and associated developments such as *Xanadu* and the World Wide Web. He discusses the differences between these technologies, spending some time reflecting on his attempts to develop Xanadu at Brown University; he suggests that many of the deficiencies of the Web come from misdirection of that phase of the project.

Nelson next examines a wide selection of topics ranging from networks (both local and the Internet), object-orientated programming, and early desktop machines, before reaching the pivot point of his book: the UNIX operating system. He chose UNIX as the fulcrum of his analysis because he believes "so much led into it and so much has resulted from it."

Nelson next considers PUI (the PARC user interface), PCs, the role of the Microsoft and Apple operating systems and their evolution, the influence of the spreadsheet, the Internet, browsers, the Internet crash, and the current major companies in computing. He explores the promise, hype, and reality of the Web 2.0 model and its likely influence. (PARC stands for the Xerox *Palo Alto Research Center*.)

The last two chapters are summaries and thought guides. The first of these suggests that it is people and ideas rather than technology that advance the computer industry and that the myth of technological necessity has stifled imagination. The final chapter illustrates what the book is about—the disagreements and decisions that have made the technical world what it is today.

Synopsis

Nelson captures most of the important developments in the computer industry, although he acknowledges that in 199 pages it is possible to tell the reader only a little of where the software ideas come from and what they are. He sets out to show how varied and conflicting the initiatives that have propelled the evolution of computer technology have been, exposing the "ideas, disagreements, manoeuvres, forgotten possibilities, and politics."

The book reads like a collection of themed essays, rather than a coherent sequence of stories. Nonetheless it is both informative and thought-provoking.

The Author

Ted Nelson is considered to be a radical thinker; he is one of the pioneers of the computer industry initiating the Xanadu project, which was started in the early 1960s with the objective of developing a computer network with a simple user interface. He is credited with inventing the term "hypertext."

He holds a first degree in philosophy, a Masters in sociology, and a Doctorate in Media and Governance. Among his honors are a visiting fellowship at the Oxford Internet Institute and a Fellowship of Wadham College, Oxford; in addition, France has knighted him as "Officier des Arts et Lettres." Visit:

http://en.wikipedia.org/wiki/Ted_Nelson
and
http://www.ibiblio.org/pioneers/nelson.html

...for more information.

—Edward Smith, BT, UK
edward.a.smith@btinternet.com

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

RIPE Announces IPv6 Website

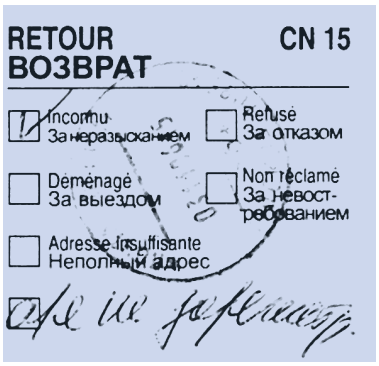
The RIPE NCC recently announced the launch of the *IPv6 Act Now!* website. Available at **www.IPv6ActNow.org**, the website explains IPv6 in terms that everyone can understand and provides a variety of useful information aimed at promoting the global adoption of IPv6. The site is designed for anyone with an interest in IPv6, including network engineers, company directors, law enforcement agencies, government representatives and civil society. The content is regularly updated and includes:

- Education, advice and opinions from the experts
- Latest IPv6-related news stories
- Videos and articles from Internet community leaders
- Current IPv4 exhaustion and IPv6 uptake statistics
- The RIPE community's statement on IPv6 deployment
- Information on community-developed IPv6 distribution policies
- Useful links to other sources of information about IPv6
- A forum for everyone to share experiences, ask questions and find answers

The site also includes contributions from other *Regional Internet Registries* (RIRs) and industry partners. If you have and comments or suggestions about the website, please contact:
ipv6actnow@ripe.net

Four-byte AS numbers from APNIC

From July 1, 2009, the *Asia Pacific Network Information Centre* (APNIC) will assign four-byte *Autonomous System* (AS) numbers by default when receiving requests. Two-byte AS numbers will only be assigned if the applicant can demonstrate that a four-byte only AS number is unsuitable. This change marks the next phase of the transition to four-byte AS numbers. The final phase begins in January 2010, when APNIC will cease to make any distinction between two-byte and four-byte AS numbers, and will operate AS number assignments from an undifferentiated four-byte AS number pool. For more information please see: **http://icons.apnic.net/asn**



Please Tell Us When You Move

We receive large quantities of undeliverable copies of *The Internet Protocol Journal*. For international mailings, the returned mail piece usually includes a standard CN 15 label, an example of which is shown here. We have an extensive collection of CN 15 labels from all over the world, but we would much rather ensure that your journal is delivered to the correct address. So, if you're moving your home or office, please use the online subscription system to update your details, or just send an e-mail message to **ipj@cisco.com** with the new information. You can also suspend paper delivery and read IPJ online if you wish.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

September 2009 Volume 12, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

This journal has covered numerous emerging technologies since we started publishing in June 1998. It would be an interesting excercise to look at which of these technologies have been successfully deployed, which ones have been rejected, and which ones are still emerging or slowly being deployed. In this issue we examine another emerging technology, or perhaps "a new concept" would be a better term, because a collection of new and old technologies are coming together to form what is collectively known as *Cloud Computing*. In a two-part article on cloud computing, T. Sridhar gives an overview of the concepts underlying this area of development. Part 1 of the article is subtitled "Models and Technologies." It will be followed by Part 2: "Infrastructure and Implementation Topics," which will be published in our next issue.

In the last year, I have had one of my credit cards "compromised" (unauthorized charges posted to the account) and subsequently replaced twice. This situation is always annoying and worrisome. Most likely, these breaches resulted from the card information being captured through an online purchase transaction. I am sure I will never know the full story, and luckily the credit card companies are pretty good about detecting fraudulent charges and quickly resolving the matter. When you start thinking about the number of network and server elements involved in a typical e-commerce transaction, it isn't entirely surprising that someone with criminal intentions could exploit a weakness in the overall system. Our second article, by Michael Behringer, explores the topic of "end-to-end security" in more detail.

Those of you who have been subscribers to this journal for several years have probably noticed that your subscription has been "auto-renewed" once a year without requiring any renewal action on your part. Starting with the December 2009 issue, we will no longer extend your subscription when it expires unless you renew it by visiting the IPJ "Subscriber Services" webpage. You will need to use your e-mail address and Subscription ID in order to gain access to your record, where you can renew, update your delivery address, or change delivery method. IPJ is available on paper, as well as online in both HTML and PDF formats. You can also contact us at **ipj@cisco.com** regarding your renewal. The expiration date and Subscription ID are printed on the back of the journal for subscribers in the United States, and on the envelope for our international subscribers. We believe that this new renewal policy will result in fewer undeliverable or unwanted copies being mailed out—a plus for the environment.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Cloud Computing—A Primer Part 1: Models and Technologies

by T. Sridhar

Cloud computing is an emerging area that affects IT infrastructure, network services, and applications. Part 1 of this article introduces various aspects of cloud computing, including the rationale, underlying models, and infrastructures. Part 2 will provide more details about some of the specific technologies and scenarios.

The term "cloud computing" has different connotations for IT professionals, depending upon their point of view and often their own products and offerings. As with all emerging areas, real-world deployments and customer success stories will generate a better understanding of the term. This discussion starts with the *National Institute of Standards and Technology* (NIST) definition:

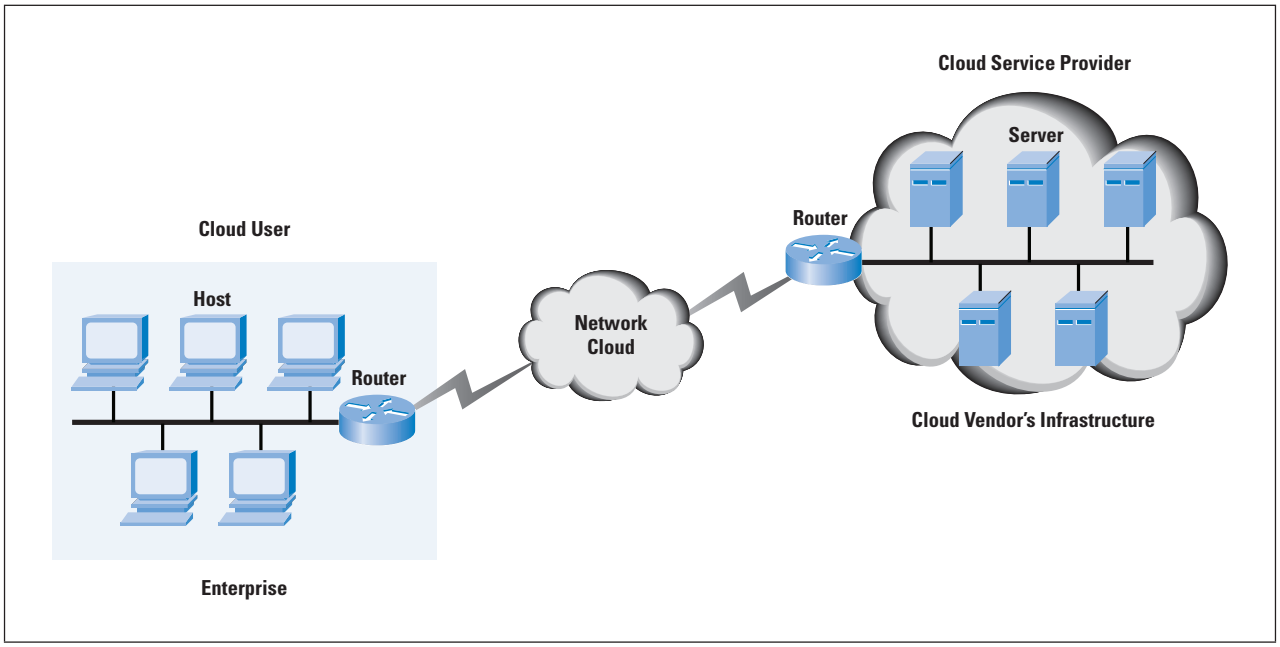
> "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

The following is a list of characteristics of a cloud-computing environment. Not all characteristics may be present in a specific cloud solution.

- *Elasticity and scalability:* Cloud computing gives you the ability to expand and reduce resources according to your specific service requirement. For example, you may need a large number of server resources for the duration of a specific task. You can then release these server resources after you complete your task.
- *Pay-per-use:* You pay for cloud services only when you use them, either for the short term (for example, for CPU time) or for a longer duration (for example, for cloud-based storage or vault services).
- *On demand:* Because you invoke cloud services only when you need them, they are not permanent parts of your IT infrastructure—a significant advantage for cloud use as opposed to internal IT services. With cloud services there is no need to have dedicated resources waiting to be used, as is the case with internal services.
- *Resiliency:* The resiliency of a cloud service offering can completely isolate the failure of server and storage resources from cloud users. Work is migrated to a different physical resource in the cloud with or without user awareness and intervention.
- *Multitenancy:* Public cloud services providers often can host the cloud services for multiple users within the same infrastructure. Server and storage isolation may be physical or virtual—depending upon the specific user requirements.

- *Workload movement:* This characteristic is related to resiliency and cost considerations. Here, cloud-computing providers can migrate workloads across servers—both inside the data center and across data centers (even in a different geographic area). This migration might be necessitated by cost (less expensive to run a workload in a data center in another country based on time of day or power requirements) or efficiency considerations (for example, network bandwidth). A third reason could be regulatory considerations for certain types of workloads.

Figure 1: Cloud Computing Context



Cloud computing involves shifting the bulk of the costs from *capital expenditures* (CapEx), or buying and installing servers, storage, networking, and related infrastructure) to an *operating expense* (OpEx) model, where you pay for usage of these types of resources. Figure 1 provides a context diagram for the cloud.

How Is Cloud Computing Different from Hosted Services?

From an infrastructure perspective, cloud computing is very similar to *hosted services*—a model established several years ago. In hosted services, servers, storage, and networking infrastructure are shared across multiple tenants and over a remote connection with the ability to scale (although scaling is done manually by calling or e-mailing the hosting provider). Cloud computing is different in that it offers a pay-per-use model and rapid (and automatic) scaling up or down of resources along with workload migration. Interestingly, some analysts group all hosted services under cloud computing for their market numbers.

Virtualization and Its Effect on Cloud Computing

It can be argued to good effect that cloud computing has accelerated because of the popularity and adoption of virtualization, specifically server virtualization. So what is virtualization? Here, virtualization software is used to run multiple *Virtual Machines* (VMs) on a single physical server to provide the same functions as multiple physical machines. Known as a *hypervisor,* the virtualization software performs the abstraction of the hardware to the individual VMs.

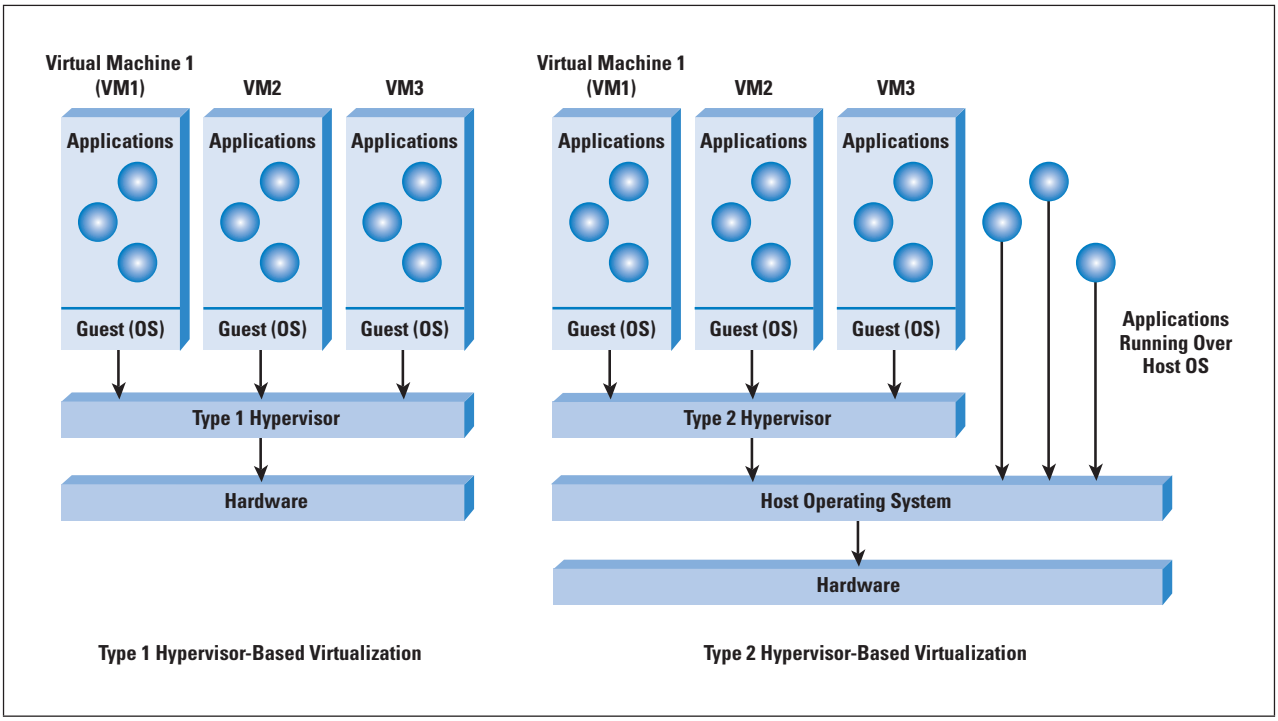
Virtualization is not new—it was first invented and popularized by IBM in the 1960s for running multiple software contexts on its mainframe computers. It regained popularity in the past decade in data centers because of server usage concerns. Data centers and web farms consisted of multiple physical servers. Measurement studies on these server farms noted that individual server usage was often as low as 15 percent for various reasons, including traffic loads and the nature of the applications (available, not always used fully), among others. The consequence of this server sprawl with low usage was large financial outlays for both CapEx and OpEx—extra machines and related power and cooling infrastructure and real estate.

Enter virtualization. A hypervisor is implemented on a server either directly running over the hardware (a *Type 1 hypervisor*) or running over an *operating system* (OS) (a *Type 2 hypervisor*). The hypervisor supports the running of multiple VMs and schedules the VMs along with providing them a unified and consistent access to the CPU, memory, and I/O resources on the physical machine. A VM typically runs an operating system and applications. The applications are not aware that they are running in a virtualized environment, so they do not need to be changed to run in such an environment. Figure 2 depicts these scenarios. The OS inside the VM may be virtualization-aware and require modifications to run over a hypervisor—a scheme known as paravirtualization (as opposed to *full virtualization*).

VM Migration: An Advantage of Virtualization

Some vendors have implemented VM migration in their virtualization solution—a big advantage for application uptime in a data center. What is VM migration? Consider the case of a server with a hypervisor and several VMs, each running an OS and applications. If you need to bring down the server for maintenance (say, adding more memory to the server), you have to shut down the software components and restart them after the maintenance window—significantly affecting application availability. VM migration allows you to move an entire VM (with its contained operating system and applications) from one machine to another and continue operation of the VM on the second machine. This advantage is unique to virtualized environments because you can take down physical servers for maintenance with minimal effect on running applications.

Figure 2: Hypervisors in Virtualization



You can perform this migration after suspending the VM on the source machine, moving its attendant information to the target machine and starting it on the target machine. To lower the downtime, you can perform this migration while the VM is running (hence the name "live migration") and resuming its operation on the target machine after all the state is migrated.

The following are some of the benefits of virtualization in a cloud-computing environment:

- *Elasticity and scalability:* Firing up and shutting down VMs involves less effort as opposed to bringing servers up or down.
- *Workload migration:* Through facilities such as live VM migration, you can carry out workload migration with much less effort as compared to workload migration across physical servers at different locations.
- *Resiliency:* You can isolate physical-server failure from user services through migration of VMs.

It must be clarified that virtualization is not a prerequisite for cloud computing. In fact, there are examples of large cloud service providers using only commodity hardware servers (with no virtualization) to realize their infrastructure. However, virtualization provides a valuable toolkit and enables significant flexibility in cloud-computing deployments.

Major Models in Cloud Computing

This section discusses some popular models of cloud computing that are offered today as services. Although there is broad agreement on these models, there are variations based on specific vendor offerings—not surprising during these early days of cloud computing.

Software as a Service

Consider the case of an enterprise with its set of software licenses for the various applications it uses. These applications could be in human resources, finance, or customer relationship management, to name a few. Instead of obtaining desktop and server licenses for software products it uses, an enterprise can obtain the same functions through a hosted service from a provider through a network connection. The interface to the software is usually through a web browser. This common cloud-computing model is known as *Software as a Service* (SaaS) or a hosted software model; the provider is known as the *SaaS Provider.*

SaaS saves the complexity of software installation, maintenance, upgrades, and patches (for example, for security fixes) for the IT team within the enterprise, because the software is now managed centrally at the SaaS provider's facilities. Also, the SaaS provider can provide this service to multiple customers and enterprises, resulting in a multitenant model. The pricing of such a SaaS service is typically on a per-user basis for a fixed bandwidth and storage. Monitoring application-delivery performance is the responsibility of the SaaS provider. `Salesforce.com` is an example of a SaaS provider. The company was founded to provide hosted software services, unlike some of the software vendors that have hosted versions of their conventional offerings.

Platform as a Service

Unlike the fixed functions offered by SaaS, *Platform as a Service* (PaaS) provides a software platform on which users can build their own applications and host them on the PaaS provider's infrastructure. The software platform is used as a development framework to build, debug, and deploy applications. It often provides middleware-style services such as database and component services for use by applications. PaaS is a true cloud model in that applications do not need to worry about the scalability of the underlying platform (hardware and software). When enterprises write their application to run over the PaaS provider's software platform, the elasticity and scalability is guaranteed transparently by the PaaS platform.

The platforms offered by PaaS vendors like Google (with its *AppEngine*) or `Force.com` (the PaaS offering from `Salesforce.com`) require the applications to follow their own *Application Programming Interface* (API) and be written in a specific language. This situation is likely to change but is a cause for concerns about lock-in. Also, it is not easy to migrate existing applications to a PaaS environment. Consequently, PaaS sees the most success with new applications being developed specifically for the cloud. Monitoring application-delivery performance is the responsibility of the PaaS provider. Pricing for PaaS can be on a per-application developer license and on a hosted-seats basis. Note that PaaS has a greater degree of user control than SaaS.

Infrastructure as a Service

Amazon is arguably the first major proponent of *Infrastructure as a Service* (IaaS) through its *Elastic Computing Cloud* (EC2) service. An IaaS provider offers you "raw" computing, storage, and network infrastructure so that you can load your own software, including operating systems and applications, on to this infrastructure. This scenario is equivalent to a hosting provider provisioning physical servers and storage and letting you install your own OS, web services, and database applications over the provisioned machines. Amazon lets you rent servers with a certain CPU speed, memory, and disk capacity along with the OS and applications that you need to have installed on them (Amazon provides some "canned" software for the OS and applications known as *Amazon Machine Images* [AMIs], so that is one starting point). However, you can also install your own OSs (or no OS) and applications over this server infrastructure.

IaaS offers you the greatest degree of control of the three models. You need to know the resource requirements for your specific application to exploit IaaS well. Scaling and elasticity are your—not the provider's—responsibility. In fact, it is a mini do-it-yourself data center that you have to configure to get the job done. Interestingly, Amazon uses virtualization as a critical underpinning of its EC2 service, so you actually get a VM when you ask for a specific machine configuration, though VMs are not a prerequisite for IaaS. Pricing for the IaaS can be on a usage or subscription basis. CPU time, storage space, and network bandwidth (related to data movement) are some of the resources that can be billed on a usage basis.

In summary, these are three of the more common models for cloud computing. They have variations and add-ons, including *Data Storage as a Service* (providing disk access on the cloud), communications as a service (for example, a universal phone number through the cloud), and so on.

Public, Private, and Internal Clouds

We have focused on cloud service providers whose data centers are external to the users of the service (businesses or individuals). These clouds are known as *public clouds*—both the infrastructure and control of these clouds is with the service provider. A variation on this scenario is the *private cloud*. Here, the cloud provider is responsible only for the infrastructure and not for the control. This setup is equivalent to a section of a shared data center being partitioned for use by a specific customer. Note that the private cloud can offer SaaS, PaaS, or IaaS services, though IaaS might appear to be a more natural fit.

An *internal cloud* is a relatively new term applied to cloud services provided by the IT department of an enterprise from the company's own data centers. This setup might seem counterintuitive at first—why would a company run cloud services for its internal users when public clouds are available? Doesn't this setup negate the advantages of elasticity and scalability by moving this service to inside the enterprise?

It turns out that the internal cloud model is very useful for enterprises. The biggest concerns for enterprises to move to an external cloud provider are security and control. CIOs are naturally cautious about moving their entire application infrastructure and data to an external cloud provider, especially when they have several person-years of investment in their applications and infrastructure as well as elaborate security safeguards around their data. However, the advantages of the cloud—resiliency, scalability, and workload migration—are useful to have in the company's own data centers. IT can use per-usage billing to monitor individual business unit or department usage of the IT resources and charge them back. Controlling server sprawl through virtualization and moving workloads to geographies and locations in the world with lower power and infrastructure costs are of value in a cloud-computing environment. Internal clouds can provide all these benefits.

This classification of clouds as public, private, and internal is not universally accepted. Some researchers see the distinction between private and internal clouds to be a matter of semantics. In fact, the NIST draft definition considers a private cloud to be the same as an internal cloud. However, the concepts are still valid and being realized in service provider and enterprise IT environments today.

When Does Cloud Computing Make Sense?

Outsourcing your entire IT infrastructure to a cloud provider makes sense if your deployment is a "green field" one, especially in the case of a startup. Here, you can focus on your core business without having to set up and provision your IT infrastructure, especially if it primarily involves basic elements such as e-mail, word processing, collaboration tools, and so on. As your company grows, the cloud-provided IT environment can scale along with it.

Another scenario for cloud usage is when an IT department needs to "burst" to access additional IT resources to fulfill a short-term requirement. Examples include testing of an internally developed application to determine scalability, prototyping of "nonstandard" software to evaluate suitability, execution of a one-time task with an exponential demand on IT resources, and so on. The term *cloud bursting* is sometimes used to describe this scenario. The cloud resources may be loosely or tightly coupled with the internal IT resources for the duration of the cloud bursting. In an extremely loosely coupled scenario, only the results of the cloud bursting are provided to the internal IT department. In the tightly coupled scenario, the cloud resources and internal IT resources are working on the same problem and require frequent communication and data sharing.

In some situations cloud computing does not make sense for an enterprise. Regulation and legal considerations may dictate that the enterprise house, secure, and control data in a specific location or geographical area. Access to the data might need to be restricted to a limited set of applications, all of which need to be internal. Another situation where cloud computing is not always the best choice is when application response time is critical. Internal IT departments can plan their server infrastructure and the network infrastructure to accommodate the response-time requirements. Although some cloud providers provide high-bandwidth links and can specify *Service-Level Agreements* (SLAs) (especially in the case of SaaS) for their offerings, companies might be better off keeping such demanding applications in house.

An interesting variation of these scenarios is when companies outsource their web front ends to a cloud provider and keep their application and database servers internal to the enterprise. This setup is useful when the company is ramping up its offerings on the web but is not completely certain about the demand. It can start with a small number of web servers and scale up or down according to the demand. Also, acceleration devices such as *Application Delivery Controllers* (ADCs) can be placed in front of the web servers to ensure performance. These devices provide server load balancing, *Secure Sockets Layer* (SSL) front ends, caching, and compression. The deployment of these devices and the associated front-end infrastructure can be completely transparent to the company; it only needs to focus on the availability and response time of its application behind the web servers.

Cloud Computing Infrastructure

The most significant infrastructure discussion is related to the data center, the interconnection of data centers, and their connectivity to the users (enterprises and consumers) of the cloud service.

A simple view of the cloud data center is that it is similar to a corporate data center but at a different scale because it has to support multiple tenants and provide scalability and elasticity. In addition, the applications hosted in the cloud as well as virtualization (when it is used) also play a part.

A case in point is the *MapReduce* computing paradigm that Google implements to provide some of its services (other companies have their own implementations of MapReduce). Put simply, the MapReduce scheme takes a set of input key-value pairs, processes it, and produces a set of output key-value pairs. To realize the implementation, Google has an infrastructure of commodity servers running Linux interconnected by Ethernet switches. Storage is local through inexpensive *Integrated Drive Electronics* (IDE) disks attached to each server.

Jobs, which consist of a set of tasks, are scheduled and mapped to the available machine set. The scheme is implemented through a *Master* machine and *Worker* machines. The latter are scheduled by the Master to implement Map and Reduce tasks, which themselves operate on chunks of the input data set stored locally. The topology and task distribution among the servers is optimized for the application (MapReduce in this case). Although Google has not made public the details of how the back-end infrastructure is implemented for Google Apps and Gmail, we can assume that the physical and logical organization is optimized for the tasks that need to be carried out, in a manner similar to what is done for MapReduce.

SaaS vendors can partition their cloud data center according to load, tenant, and type of application that they will offer as a service. In some cases they might have to redirect the traffic to a different data center, based on the load in the default data center. IaaS provides the greatest degree of control for the user, as discussed earlier. Even here, the topology and load assignment can be based on the number and type of servers that are allocated.

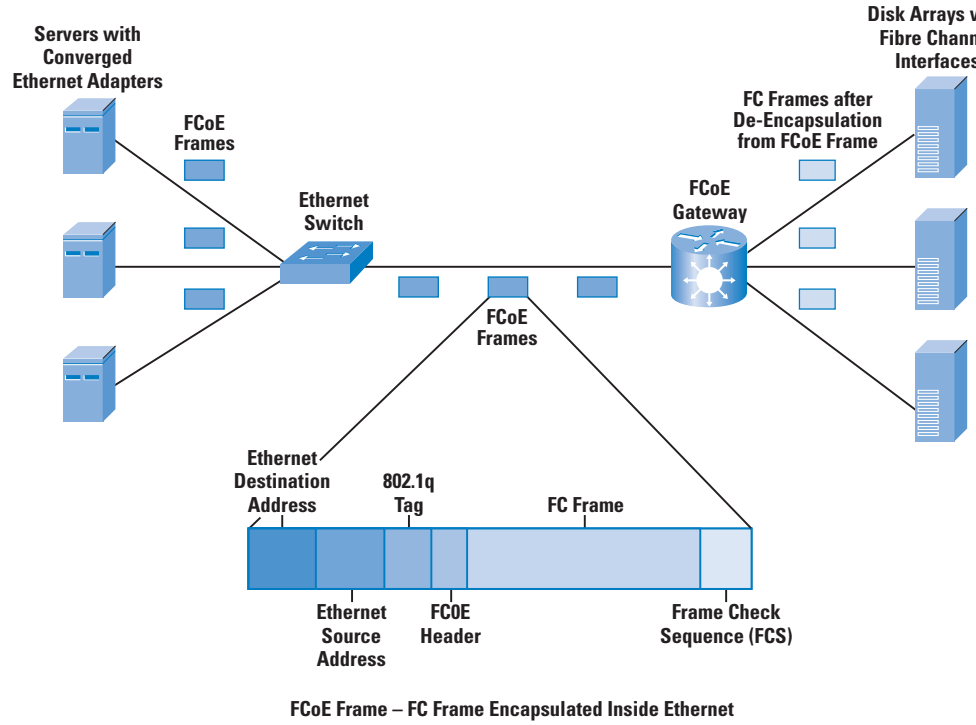
Storage Infrastructure

Storage plays a major part in the data center and for cloud services, especially in environments with virtualization. Storage can be locally attached or accessible through a network—the most popular storage network technologies being *Fibre Channel* and Ethernet. For such network access of storage, servers are equipped with Fibre Channel or Ethernet adapters through which they connect to a Fibre Channel or Ethernet switch. The switch provides the connectivity to storage arrays. Fibre Channel is more popular, though *Network Attached Storage* (NAS) devices with Ethernet interfaces also have a strong presence in the data center. Another Ethernet-based storage option is the *Internet Small Computer System Interface* (iSCSI), which is quite popular among smaller data centers and enterprises because of the cost benefits. This technology involves running the SCSI protocol on a TCP/IP-over-Ethernet connection.

Fibre Channel connections to the storage network necessitate two types of network technologies in the data center: Ethernet for server-to-server and server-to-client connectivity and Fibre Channel for server-to-storage connectivity. A recent initiative in data-center technology is a converged network, which involves the transport of *Fibre Channel over Ethernet* (FCoE). FCoE removes the need for each server to have an Fibre Channel adapter to connect to storage. Instead, Fibre Channel traffic is encapsulated inside an Ethernet frame and sent across to a FCoE gateway that provides Ethernet-to-FCoE termination to connect to Fibre Channel storage arrays (refer to Figure 3). Some storage products provide FCoE functions, so the Ethernet frame can be carried all the way to the storage array. An adapter on the server that provides both "classical" Ethernet and FCoE functions is known as a *Converged Network Adapter* (CNA). Cloud-computing environments can reduce the data-center network complexity and cost through this converged network environment.

Another area in which storage is important is in virtualization and live migration. When a VM migrates to a different physical machine, it is important that the data used by the VM is accessible to both the source and the target machines. Alternatively, if the VM is migrated to a remote data center, the stored data needs to be migrated to the remote data center too. Also, in a virtualized environment, the Fibre Channel, Ethernet, or converged adapter driver should support multiple VMs and interleave its storage traffic to the storage devices. This interleaving is done in consonance with the hypervisor and a designated VM (paravirtualized environments often use this tool), as appropriate.

Figure 3: FCoE in a Cloud Data-Center Environment



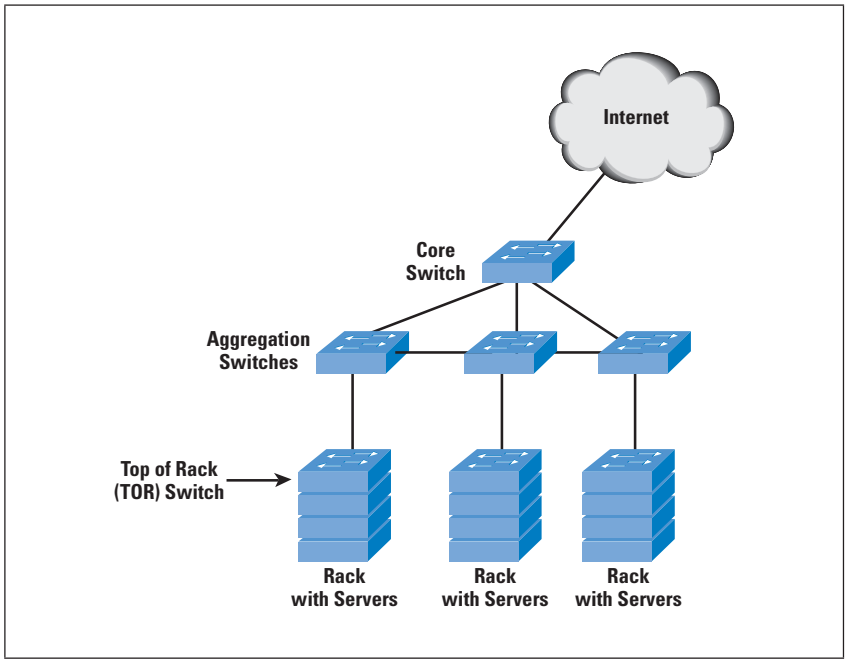
Cloud Computing: Effect on the Network

The previous discussion indicated that the network is a big part of cloud computing. A cloud user connects to the network to access the cloud resources, as indicated earlier in Figure 1. The cloud is accessible through a public network (the Internet) or through a private network (dedicated lines or *Multiprotocol Label Switching* [MPLS] infrastructure, for example). Response-time guarantees depend upon this connectivity. Some cloud vendors offer dedicated links to their data centers and provide appropriate SLAs for uptime or response time and charge for such SLAs. Others might implement a best-effort scheme but provide tools for monitoring and characterizing application performance and response time, so that users can plan their bandwidth needs.

The most significant effect on the network is in the data center, as indicated previously. Let us start with the network architecture or topology. The most common network architecture for enterprises is the three-layer architecture with access, aggregation or distribution, and core switches. The data center requires a slightly different variation to this layering, as proposed by some vendors. The data center consists mainly of servers in racks interconnected through a *Top-of-Rack* (TOR) Ethernet switch which, in turn, connects to an aggregation switch, sometimes known as an *End-of-Rack* (EOR) switch (Figure 4).

The aggregation switch connects to other aggregation switches and through these switches to other servers in the data center. A core switch connects to the various aggregation switches and provides connectivity to the outside world, typically through Layer 3 (IP). It can be argued that most of intra-data center traffic traverses only the TOR and the aggregation switches. Hence the links between these switches and the bandwidth of those links need to account for the traffic patterns. Some vendors have proposed a fat-tree or a leaf-spine topology to address this anomaly, though this is not the only way to design the data-center network. Incidentally, the fat-tree topology is not new—it has been used in *Infiniband* networks in the data center.

Figure 4: Example Data-Center Switch Network Architecture



The presence of virtualized servers adds an extra dimension. Network connections to physical servers will need to involve "fatter pipes" because traffic for multiple VMs will be multiplexed onto the same physical Ethernet connection. This result is to be expected because you have effectively collapsed multiple physical servers into a single physical server with VMs. It is quite common to have servers with 10-Gbps Ethernet cards in this scenario.

New Protocols for Data-Center Networking

Numerous initiatives and standards bodies are addressing the standards related to cloud computing. From the networking side, the IEEE is working on new protocols and the enhancement of existing protocols for data centers. These enhancements are particularly useful in data centers with converged networks—the area is often known as *Convergence Enhanced Ethernet* (CEE).

A previous section indicated the importance of FCoE for converged storage network environments. The IEEE is working to enable FCoE guarantees (because Fibre Channel is a reliable protocol as compared to best-effort Ethernet) through an Ethernet link in what is known as "Lossless Ethernet." FCoE is enabled through a *Priority Flow Control* (PFC) mechanism in the 802.1Qbb activities in the IEEE. In addition, draft IEEE 802.1Qau provides end-to-end congestion notification through a signaling mechanism propagating up to the ingress port, that is, the port connected to the server *Network Interface Card* (NIC). This feature is useful in a data-center topology.

A third draft IEEE 802.1aq defines shortest-path bridging. This work is similar to the work being done in the IETF TRILL *(Transparent Interconnect of Lots of Links)* working group. The key motivation behind this work is the relatively flat nature of the data-center topology and the requirement to forward packets across the shortest path between the endpoints (servers) to reduce latency, rather than a root bridge or priority mechanism normally used in the *Spanning Tree Protocol* (STP). The shortest-path bridging initiative in IEEE 802.1aq is an incremental advance to the *Multiple Spanning Tree Protocol* (MSTP), which uses the *Intermediate System-to-Intermediate System* (IS-IS) link-state protocol to share learned topologies between switches and to determine the shortest path between endpoints.

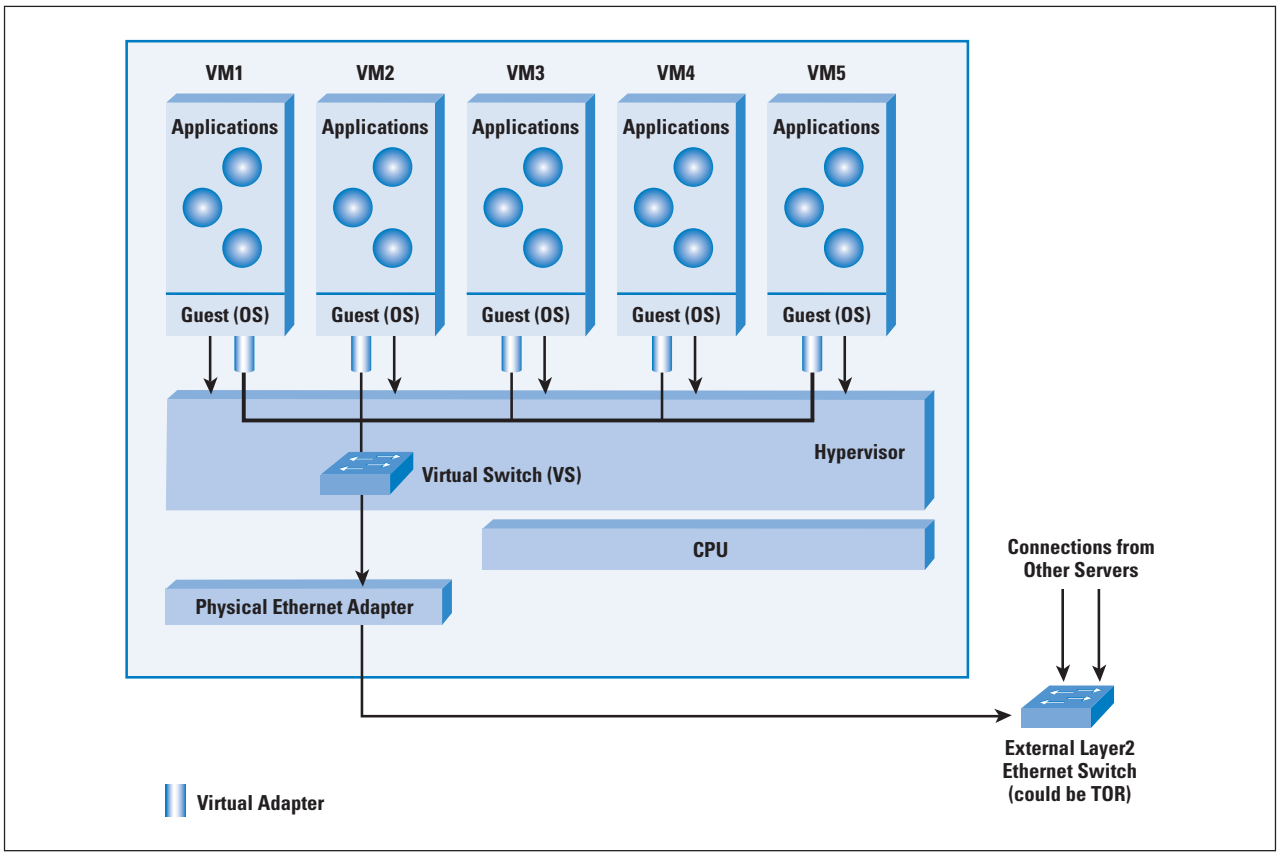
The fourth draft 802.1Qaz is also known as *Enhanced Transmission Selection* (ETS). It allows lower-priority traffic to burst and use the unused bandwidth from the higher-priority traffic queues, thus providing greater flexibility.

Virtualized Network Equipment Functions

Though cloud computing does not depend upon virtualization, several cloud infrastructures are built with virtualized servers. In an environment with physical servers, switches are used to connect servers to other servers. Firewalls and application-delivery controllers are other types of equipment that you can use in a data center on the connection to external clients. With a virtualized environment, you can move some or all of these functions to reside inside a server.

Consider the case of the software-based *Virtual Switch* as shown in Figure 5. You can use the Virtual Switch to switch between VMs inside the same physical server and aggregate the traffic for connection to the external switch. The Virtual Switch is often implemented as a plug-in to the hypervisor. The VMs have virtual Ethernet adapters that connect to the Virtual Switch, which in turn connects to the physical Ethernet adapter on the server and to the external Ethernet switch. To the network manager, the virtual switch can appear as a part of the network. Unlike physical switches, the Virtual Switch does not necessarily have to run network protocols for its operation, nor does it need to treat all its ports the same because it knows that some of them are connected to virtual Ethernet ports (for example, it can avoid destination address learning on the ports connected to the VMs). It can function through appropriate configuration from an external management entity.

Figure 5: Virtual Ethernet Switch in a Virtualized Server Environment



It is possible to implement a virtualized firewall as a VM instead of as a plug-in to the hypervisor. These VMs are self-contained, with an operating system along with the firewall software. The complete package is known as a *firewall virtual appliance*. These VMs can be loaded and configured so that network packets destined for any of the VMs pass through the firewall VM, where they are validated before being passed to the other VMs. Another use of the firewall VM is as a front end to the physical servers in the data center. The disadvantage of a virtual appliance is the performance hit due to its implementation as a software function in a virtualized environment.

Management

Management has several facets in a cloud-computing environment: billing, application-response monitoring, configuring network resources (virtual and physical), and workload migration. In a private cloud or tightly coupled environment, management of the applications may have to be shared between the internal cloud and the private cloud.

You can manage cloud-computing environments in several ways, depending upon the specific area. You can manage the network equipment (physical and virtual) through the *Simple Network Management Protocol* (SNMP) and a network management console. In a virtualized environment, the virtualization vendor often offers a framework to manage and monitor VMs, so this is another part of the equation. Several vendors offer products to act as management front ends for public clouds; for example, Amazon, whose products act as brokers and management consoles for your application deployed over the Amazon cloud offering.

It is clear that this area of management for cloud computing is still evolving and needs to be tied together for a unified management view.

Cloud Computing: Common Myths

Thus far, we have considered the important technologies, terminology, and developments in cloud computing. This section outlines some common myths about cloud computing.

- *Myth: Cloud computing should satisfy all the requirements specified: scalability, on demand, pay per use, resilience, multitenancy, and workload migration.*

  In fact, cloud-computing deployments seldom satisfy all the requirements. Depending upon the type of service offered (SaaS, IaaS, or PaaS), the service can satisfy specific subsets of these requirements. There is, however, value in trying to satisfy most of these requirements when you are building a cloud service.

- *Myth: Cloud computing is useful only if you are outsourcing your IT functions to an external service provider.*

  Not true. You can use cloud computing in your own IT department for on-demand, scalable, and pay-per-use deployments. Several vendors offer software tools that you can use to build clouds within your enterprise's own data center.

- *Myth: Cloud computing requires virtualization.*

  Although virtualization brings some benefits to cloud computing, including aspects such as efficient use of servers and workload migration, it is not a requirement for cloud computing. However, virtualization is likely to see increased usage in cloud deployments.

- *Myth: Cloud computing requires you to expose your data to the outside world.*

  With internal clouds you will never need to expose your data to the outside world. If data security and privacy are concerns, you can develop a cloud model where web front ends are in the cloud and back-end data always resides in your company's premises.

- *Myth: Converged networks are essential to cloud computing.*

  Although converged networks (with FCoE, for example) have benefits and will see increased adoption in data centers in the future, cloud computing is possible without converged networks. In fact, some cloud vendors use only Fibre Channel for all their storage needs today. Use of converged networks in the future will result in cost efficiencies, but it is not a requirement today.

Cloud Computing: Gaps and Concerns

Cloud-computing technology is still evolving. Various companies, standards bodies, and alliances are addressing several remaining gaps and concerns. Some of these concerns follow:

- *Security:* Security is a significant concern for enterprise IT managers when they consider using a cloud service provider. Physical security through isolation is a critical requirement for private clouds, but not all cloud users need this level of investment. For those users, the cloud provider must guarantee data isolation and application security (and availability) through isolation across multiple tenants. In addition, authentication and authorization of cloud users and encryption of the "network pipe" from the cloud user to the service provider application are other factors to be considered.
- *Network concerns:* When cloud bursting is involved, should the servers in the cloud be on the same Layer 2 network as the servers in the enterprise? Or, should a Layer 3 topology be involved because the cloud servers are on a network outside the enterprise? In addition, how would this work across multiple cloud data centers?
- *Cloud-to-cloud and Federation concerns:* Consider a case where an enterprise uses two separate cloud service providers. Compute and storage resource sharing along with common authentication (or migration of authentication information) are some of the problems with having the clouds "interoperate." For virtualized cloud services, VM migration is another factor to be considered in federation.
- *Legal and regulatory concerns:* These factors become important especially in those cases involving storing data in the cloud. It could be that the laws governing the data are not the laws of the jurisdiction where the company is located.

Conclusion

This article introduced the still-evolving area of cloud computing, including the technologies and some deployment concerns. Definitions and standardization in this area are a work in progress, but there is clear value in cloud computing as a solution for several IT requirements. In Part 2 we will provide a more detailed look at some of the technologies and scenarios for cloud computing.

For Further Reading

[1] Draft NIST Working Definition of Cloud Computing, **http://csrc.nist.gov/groups/SNS/cloud-computing/index.html**

[2] "Identifying Applications for Public and Private Clouds," Tom Nolle, Searchcloudcomputing, **http://searchcloudcomputing.techtarget.com/tip/0,289483,sid201_gci1358701,00.html?track=NL-1329&ad=710605&asrc=EM_NLT_7835341&uid=8788654**

[3] "The Wisdom of Clouds," James Urquhart's blog on Cloud Computing, **http://news.cnet.com/the-wisdom-of-clouds/**

[4] "Virtualization – State of the Art," SCOPE Alliance, **http://www.scope-alliance.org/sites/default/files/documents/SCOPE-Virtualization-StateofTheArt-Version-1.0.pdf**

[5] "Live Migration of Virtual Machines," Clark, et al., **http://www.cl.cam.ac.uk/research/srg/netos/papers/2005-migration-nsdi-pre.pdf**

[6] "MapReduce: Simplified Data Processing on Large Clusters," Dean & Ghemawat, **http://labs.google.com/papers/mapreduce.html**

[7] "Cloud Computing Drives New Networking Requirements," *The Lippis Report,* 120, **http://lippisreport.com/2009/02/lippis-report-120-cloud-computing-drives-new-networking-requirements/**

[8] "A New Approach to Network Design When You Are in the Cloud," *The Lippis Report,* 121, **http://lippisreport.com/2009/03/a-new-approach-to-network-design-in-the-cloud/**

[9] "Unified Fabric Options Are Finally Here," *The Lippis Report,* 126, **http://lippisreport.com/2009/05/lippis-report-126-unified-fabric-options-are-finally-here/**

[10] "Virtualization with Hyper-V," Microsoft, **http://www.microsoft.com/windowsserver2008/en/us/hyperv-overview.aspx**

[11] "Citrix XenServer," Citrix, **http://www.citrix.com/English/ps2/products/feature.asp?contentID=1686939**

[12] "VMware Virtual Networking Concepts," VMware, **http://www.vmware.com/files/pdf/virtual_networking_concepts.pdf**

[13] "Cisco Nexus 1000v Virtual Ethernet Switch," Cisco Systems, **http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9902/data_sheet_c78-492971.html**

[14] "Application Delivery Challenge," Layland Consulting, **http://www.edge-delivery.org/dl/whitepapers/Application_Delivery_Challenge.pdf**

[15] "Cloud Networking: Design Patterns for 'Cloud-Centric' Application Environments," **http://www.aristanetworks.com/en/CloudCentricDesignPatterns.pdf**

[16] IEEE 802.1Qaz – Enhanced Transmission Selection, **http://www.ieee802.org/1/pages/802.1az.html**

[17] IEEE 802.1Qau – Congestion Notification, **http://www.ieee802.org/1/pages/802.1au.html**

[18] IEEE 802.1Qbb – Priority Flow Control, **http://www.ieee802.org/1/pages/802.1bb.html**

[19] IEEE 802.1aq - Shortest Path Bridging, **http://www.ieee802.org/1/pages/802.1aq.html**

[20] IETF Transparent Interconnection of Lots of Links (trill) Working Group, **http://www.ietf.org/dyn/wg/charter/trill-charter.html**

T. SRIDHAR received his BE in Electronics and Communications Engineering from the College of Engineering, Guindy, Anna University, Madras, India, and his Master of Science in Electrical and Computer Engineering from the University of Texas at Austin. He can be reached at **TSridhar@leitnet.com**

Why End-to-End Security Is Necessary But Not Sufficient

by Michael H. Behringer, Cisco Systems

End-to-end security relies on protocols and mechanisms that are implemented exclusively on the endpoints of a connection. The most typical example is an HTTPS connection (based, for example, on *Transport Layer Security* (TLS)[1]) to a web server; *IP Security* (IPsec)[2] can also be used for end-to-end security, as was initially proposed as a default connection mechanism for IPv6.

There is a perception that end-to-end security is sufficient as a security solution, and that network-based security is obsolete in the presence of end-to-end security. This article outlines why in practice end-to-end security alone is not sufficient, and why network-based security is also required.

Defining "End"

The traditional definition of an endpoint is a client or server. In this definition end-to-end security starts on the client and ends on the server. Given the multitude of applications running in parallel on an operating system, and given increasing virtualization, this definition is usually no longer precise enough. The operating system can establish a security association on either the session or application level. It can also be terminated on a front end, on behalf of numerous servers, as is the case in many TLS[1] deployments.

Because the main goal of this article is to understand why the network has a role to play in security, the precise definition of an endpoint is not relevant here. Abstractly seen, an endpoint is an entity that communicates over a network with another entity. This definition, albeit vague, is sufficient for the discussion at hand.

End-to-End Security Is Fundamental

Security on the endpoints (client-server, or client-client for peer-to-peer) is an absolute requirement for secure communications. Such a solution contains the following components:

- *Identity:* This component encompasses known and verifiable entity identities on both ends; note that an identity can be temporary for a connection. For example, a user often is identified by username and password, whereas a server may be identified through a server certificate.
- *Protocols* (for example, TLS [1] and IPsec [2]): Protocols are used to dynamically negotiate session keys, and to provide the required security functions (for example, encryption and integrity verification) for a connection. Protocols use algorithms to implement these functions.

- *Algorithms* (for example, *Advanced Encryption Standard* [AES][3], *Triple Digital Encryption Standard* [3DES][4], and *Secure Hash Algorithm* [SHA-1][5]): These algorithms use the previously mentioned session keys to protect data in transit, for example through encryption or integrity checks.
- *Secure implementation:* The endpoint (client or server) that runs one of these protocols mentioned previously must be free of bugs that could compromise security. Web browser security is relevant here. Also malware can compromise security, for example by logging key strokes on a PC.
- *Secure operation:* Users and operators have to understand the security mechanisms, and how to deal with exceptions. For example, web browsers warn about invalid server certificates, but users can override the warning and still make the connection. This concern is a nontechnical one, but is of critical concern today.

For full end-to-end security, all of these components must be secure. In networks with end-to-end security, both ends can typically (depending on the protocols and algorithms used) rely on the fact that their communication is not visible to anyone else, and that no one else can modify the data in transit. End-to-end security is used successfully today, for example, in online banking applications. Correct and complete end-to-end security is required; without it, many applications such as online banking would not be possible.

However, a single security problem in any of the components can compromise the overall security for a connection. Today, most critical are implementation problems on endpoints, as well as human errors, specifically in handling exception cases.

Practical Shortcomings of End-to-End Security

Solutions that rely exclusively on end-to-end security have many potential problems, which fall into two broad categories: those that affect the end user and those that affect the network operator (the service provider, or the enterprise network operator, for example).

The End-User View

As reports on online crime and fraud demonstrate very clearly, even in the perceived presence of end-to-end security it is difficult to ensure that none of the components mentioned previously is "broken." Although protocols and algorithms in use tend to be secure and reliable, the main problems lie in the two main areas of endpoint security (secure implementation component) and lack of user education (secure operation component).

Endpoint security concerns include the presence of malware, as well as bugs in software. Even security professionals have difficulty determining whether a PC contains malware. Such malware can control the connection before it is secured, thereby achieving the ability to see the data, as well as potentially change it in real time. Although endpoint security software such as antivirus solutions as well as zero-day prevention solutions provides good security, they are not always installed, and antivirus software is often not up-to-date. Users also can temporarily disable the solutions. Therefore, the presence of malware remains a security concern. Bugs in software are also relevant, for example in the web browser or the operating system.

The lack of user education is the other important concern on the endpoint: Users must know how to identify a secured connection, for example by the little padlock in a web browser (although not even this security mechanism is completely secure). They must also know how to deal with exceptions such as expired or invalid certificates. Most average users do not entirely understand all these details, leading to breaches of security.

The Network Operator View

In the early days of IPv6 it was postulated that the protocol would come with IPsec end-to-end security built in and always "on," thereby eliminating all security problems. This assumption turned out to be wrong, because many problems remain on the network side—for example, general problems with end-to-end security—and they apply to all variants, such as IPsec, TLS, or *Secure Sockets Layer* (SSL).

Today, most enterprise network operators as well as service providers are skeptical about the ubiquitous use of end-to-end security solutions. The fundamental concern is that the endpoints generally cannot be trusted. The network operator, whether enterprise, university, or service provider, has an obligation to enforce certain policies on the endpoint, for example, to ensure that it does not spread worms, send spam mail, or attack servers. If, however, network operators cannot "see" the traffic of an endpoint because it is end-to-end secured, then they cannot comply with their obligations to control the endpoints.

From a network operator's perspective it is therefore not generally desirable to use end-to-end security for all communications, but only for those that really need it.

Why Network-Based Security Is Essential

There are many examples where network-based security is essential, and where end-to-end security solutions not only do not help, but may actually present an additional problem. In all those cases it is essential to have strong network-based security solutions in place. Some examples explain this in more detail.

The Service Provider with DSL Customers

A service provider with DSL customers needs to control its users' traffic in various ways. However, the provider has no control over the endpoints, because those are the customers' property. Because they also cannot force their customers to use appropriate security software, there is always a certain percentage of infected PCs on any given service provider's network. Critical service provider concerns follow:

- *Control of PCs infected with malware:* Such PCs (also referred to as "bots" or "zombies") can infect other PCs and participate in illegal activities, such as spam mail, click fraud[12], *Denial-of-Service* (DoS) attacks, etc. There is a strong, often legal requirement for providers to identify such infected PCs, to isolate them, and to alert their owners and help them to "disinfect" the PC. Network-based security mechanisms are required, essentially because security on the endpoint has failed.
- *Attacks from the users:* Even in the absence of malware, a service provider's user can participate in illegal activities, such as DoS attacks, or intrusions on web servers or routers. Network-based methods are required to detect such attempts, beginning with simple forms such as IP spoofing [6], and to prevent or block them. One example is network-based solutions against DoS attacks[7,8].
- *Control of bandwidth:* Many service providers need to enforce bandwidth limits on some applications or users because they violate service agreements. Also here, applications are necessary to control the PCs, and to limit their usage of the service to remain within contracted boundaries. Service providers today employ a large number of network-based security mechanisms, ranging from visibility solutions to enforcement of certain policies. Endpoint security does not solve these problems, because the PC is not under control of the service provider, and is typically untrusted.
- *Services:* Service providers also try to differentiate themselves from their competition by offering managed services, for example managed security services[9]. Those services are also network-based, and they complement endpoint security solutions that their customers use.

The Service Provider with Customers Under Attack

Service providers may also be required to help their customers when they are under attack. DoS attacks illustrate why endpoint security may not be sufficient, and network-based security is required. Under a DoS attack, a web server, for example, may receive more traffic than it can handle. Such attacks can also overload network resources, such as subscriber lines or routers; therefore, endpoint security is not able to solve such attacks. Massive overprovisioning would be the only way to handle DoS attacks, but this approach is commercially not generally feasible. Network-based solutions based on flow analysis and selective discard of flows are required to help in such situations.

The Enterprise Network

At first glance it seems that enterprises should have full control over the PCs in the enterprise. In such a case, it would be possible to rely completely on end-to-end security. However, this assumption is unrealistic. Numerous current shortcomings make this approach impractical today:

- Enterprise PCs can also get infected with malware, leading to the same problem as for service providers described previously: the need to monitor and control the behavior of a PC in the network. Solutions to control endpoints are themselves network-based; for example, network endpoint assessment[10] and user authentication (802.1x)[11].
- Attacks from users, or against services within the enterprise, also exist in an enterprise environment, as explained previously for service providers. Solutions are network-based.
- The enforcement of *Quality of Service* (QoS) is also a security concern: Users could wrongly classify all their traffic as "high-priority." In the absence of full application control on the PC (which is impractical today), the network needs to control flows from the PC, and potentially enforce a QoS policy. If all flows were encrypted end-to-end, this control would be "blind," probably leading to undesired results. Network security mechanisms are required to control the QoS policy.
- Scale: In an enterprise with several offices that are connected over an untrusted network (for example, the Internet), it may be impractical today to roll out full end-to-end security across the entire enterprise. The currently used approach in most enterprises is to connect the offices with IPsec gateways, and leave traffic within an office in the clear. This scenario increases manageability and scalability of the network. Again, this solution is network-based security solution.
- Although PCs can theoretically be equipped with IPsec (for example) for all communications, many end devices in an enterprise do not support the security mechanisms required. Printers, faxes, and scanners are examples. Full end-to-end security, however, would require all endpoints to support a common mechanism, such as IPsec or TLS. Until all such devices have this support, network-based mechanisms are required to secure communications with them.

Summary

End-to-end security protocols and solutions are an essential cornerstone in network security. We cannot live without them. However, it is unrealistic in today's networks to assume that end-to-end security solutions alone will suffice. The fundamental underlying problem is that typically the network operator, where a PC is attached, has a need and often an obligation to monitor the behavior of the endpoint, and to control malicious activities emerging from that PC. All solutions to control endpoints, however, are by definition network-based. Therefore, network-based security mechanisms are also an essential component of overall network security: Overall security requires both endpoint security and network-based security.

References

[1] T. Dierks, et al., "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, August 2008.

[2] S. Kent, et al., "Security Architecture for the Internet Protocol," RFC 4301, December 2005.

[3] Joan Daemen and Vincent Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard,* Springer-Verlag, 2002. ISBN 3-540-42580-2.

[4] ANSI X9.52:1998, "Triple Data Encryption Algorithm Modes of Operation," July 1998.

[5] FIPS 180-2, "Secure Hash Standard (SHS)," February 2004.

[6] F. Ali, "IP Spoofing," *The Internet Protocol Journal,* Volume 10, No. 4, December 2007.

[7] W. Eddy, "Defenses Against TCP SYN Flooding Attacks," *The Internet Protocol Journal,* Volume 9, No. 4, December 2006.,

[8] C. Patrikakis, et al., "Distributed Denial of Service Attacks," *The Internet Protocol Journal,* Volume 7, No. 4, December 2004.

[9] K. Trivedi and D. Holloway, "Secure Multivendor Networks," *The Internet Protocol Journal,* Volume 10, No. 3, September 2007.

[10] P. Sangster, et al., "Network Endpoint Assessment (NEA): Overview and Requirements," RFC 5209, June 2008.

[11] IEEE 802.1X "Port-Based Network Access Control," `http://www.ieee802.org/1/pages/802.1x.html`

[12] According to Wikipedia: "Click fraud is a type of Internet crime that occurs in pay per click online advertising when a person, automated script or computer program imitates a legitimate user of a web browser clicking on an ad, for the purpose of generating a charge per click without having actual interest in the target of the ad's link. Click fraud is the subject of some controversy and increasing litigation due to the advertising networks being a key beneficiary of the fraud.

Use of a computer to commit this type of Internet fraud is a felony in many jurisdictions, for example, as covered by *Penal Code 502* in California, USA, and the *Computer Misuse Act 1990* in the United Kingdom. There have been arrests relating to click fraud with regard to malicious clicking in order to deplete a competitor's advertising budget."
http://en.wikipedia.org/wiki/Click_fraud

MICHAEL H. BEHRINGER works at Cisco Systems as a distinguished engineer, where he focuses on core security problems, such as MPLS security, multicast security, and Denial-of-Service attack prevention. Michael holds a diploma in computer science from the Technical University of Munich. He is an active member of the IETF, and has published several papers, RFCs, and a book about MPLS VPN security. E-mail: **mbehring@cisco.com**

Letter to the Editor

End of Eternity

Dear Ole,

In their "The End of Eternity" articles, (IPJ Volume 11, No. 4 and Volume 12, No. 1) Niall Murphy and David Wilson provide a detailed and compelling description of the lasting harm that could result from the exhaustion of unallocated IPv4 addresses—harm to Internet users and aspiring new entrants, to technical-coordination and fault-management mechanisms, and to the likely irreplaceable cooperative decision-making and consensus-development mechanisms that distinguish the Internet from every other important transnational sphere of activity in human history. Thankfully, the authors foresee a potential happy ending—or at least yet another chapter in the story—in "an IPv6 Internet, or at least enough of one to keep off address scarcity for a workable subset of the industry."

However, having foreshadowed how they expect the IP addressing cliffhanger to be resolved, the authors go on to detail a variety of interesting but considerably less persuasive assumptions and predictions, all based on the *stipulation* that establishing IPv4 address markets would represent the best means to "shorten the gap" between the end of IPv4 and the return to a "normal" state of Internet growth and development, that is, one that is unconstrained by IP address-related scarcity (or at least no more constrained than it has been over the last decade-plus of CIDR and hierarchical interdomain routing).

I believe that it is worth highlighting here the logic that binds these two engaging and well-written articles together into something that is, unfortunately, substantially less than the sum of its parts. If the authors are to be taken at their word that "an IPv6 Internet" represents the only currently feasible and also *satisfactory* conclusion to "the IPv4 end game," then that conclusion does not by itself entail that IPv4 markets are the only, or most obvious or effective—or even *workable*—candidate mechanisms for coordinating the distribution of IP addressing in the run-up to more widespread IPv6 adoption. And yet, that postulate is offered, without explanation or defense, as the grounding justification for an investigation of various optional features and collateral effects that the foretold IPv4 address market might have.

Many observers have committed untold pages and pixels to the exploration of hypothetical IPv4 address markets, both in IPJ and elsewhere, going back as far as RFC 1744 (1994). The two articles by Murphy and Wilson represent valuable additions to that growing corpus. However, to my knowledge, no other writings in this area have built on the proposition that IPv6 is indispensable; therefore, IPv4 addresses should be privately traded. To put it in the most generous possible terms, this claim is highly contestable. As separate and independent analyses, IPJ readers may derive many useful insights from these two articles, but attributing any special relevance to those insights based on any presumptive connection between IPv4 markets and the future necessity or viability of IPv6 would be a mistake.

—Tom Vest, Consultant
tvest@eyeconomics.com

Fragments

CSNET Receives 2009 Postel Service Award

The *Internet Society* (ISOC) has awarded the *Jonathan B. Postel Service Award* for 2009 to CSNET, the *Computer Science Network*, a research networking effort that during the early 1980s provided the critical bridge from the original research undertaken through the ARPANET to the modern Internet.

The award recognizes the pioneering work of the four principal investigators that conceived and later led the building of CSNET—Peter J. Denning, David Farber, Anthony C. Hearn and Lawrence Landweber—and the U.S. National Science Foundation program officer and visionary responsible for encouraging and funding CSNET—Kent Curtis.

Stephen Wolff, a past recipient of the Postel Award, said, "CSNET was a critical link in the transition from the research-oriented ARPANET to today's global Internet. CSNET also helped lead the way by sharing technologies, fostering connections, and nurturing the worldwide community that provided a foundation for the global expansion of the Internet."

ISOC presented the award, including a US$20,000 honorarium and a crystal engraved globe, during the 75th meeting of the *Internet Engineering Task Force* (IETF) in Stockholm, Sweden. The awardees have requested that the ISOC present the honorarium to non-profit organizations they believe support the spirit of the award.

Lynn St. Amour, President and CEO of the ISOC, said "In many ways, CSNET helped set the stage for the Internet that today reaches more than 1 billion people. CSNET's community-driven, self-sustaining governance structure was an early example of the model that helps ensure that even as today's Internet grows and evolves, it remains an open platform for innovation around the world."

CSNET began in 1981 with a five-year grant from the U.S. *National Science Foundation* (NSF). Five years later, CSNET connected more than 165 academic, government and industrial computer research groups comprised of more than 50,000 researchers, educators and students across the United States and around the world. It had concluded a seminal resource sharing agreement with the ARPANET and was self-governing and self-supporting. Open to all computer researchers, it demonstrated that researchers valued the kind of informal collaboration it made possible. CSNET's success was critical to the decision by NSF in 1986 to adopt the Internet technology for NSFNET, the network backbone to connect its supercomputing centers and their research communities. CSNET provided software, policies, and experienced alumni to the NSFNET teams. NSFNET became the first backbone of the modern Internet.

The CSNET architecture supported the Internet standards, SMTP and TCP/IP, and a variety of connection protocols including telephone dialup, X.25, and ARPANET. This architecture, along with strong technical support, enabled participants of differing means and skill levels to all join the community. CSNET pioneered the model of university, industry, government partnerships that were key to the pre-commercial Internet.

The CSNET proposal was assembled by a lengthy community consensus process that began in 1979. The four principal investigators, who led this effort and served as the project's management committee, were:

Peter Denning was head of the computer science department at Purdue University. His team included professor Douglas Comer, who was responsible for the software that ran TCP/IP over the GTE Telenet X.25 commercial packet network.

David Farber was a professor of electrical engineering at University of Delaware. His team included then graduate student David Crocker, who was responsible for Phonenet, dial-in telephone connections to relay servers for e-mail exchange.

Anthony Hearn was head of the information sciences department at RAND. His team included Michael O'Brien, who was responsible for the relays connecting CSNET and ARPANET.

Lawrence Landweber was a professor of computer science at the University of Wisconsin. His team included professor Marvin Solomon and Michael Litzkow who were responsible for the name server, a precursor of modern Directory Services.

At the NSF, the late *Kent Curtis* helped conceive the entire effort and, with assistance from Bill Kearn, saw it through its formative years. He was recognized for his pivotal role by the Computing Research Association's first distinguished service award in 1988.

The *Jonathan B. Postel Service Award* was established by the Internet Society to honor individuals or organizations that, like Jon Postel, have made outstanding contributions in service to the data communications community. The award is focused on sustained and substantial technical contributions, service to the community, and leadership. With respect to leadership, the nominating committee places particular emphasis on candidates who have supported and enabled others in addition to their own specific actions. Previous recipients of the Postel Award include Jon himself (posthumously and accepted by his mother), Scott Bradner, Daniel Karrenberg, Stephen Wolff, Peter Kirstein, Phill Gross, Jun Murai, Bob Braden and Joyce K. Reynolds (jointly), Nii Quaynor, and La Fundación Escuela Latinoamericana de Redes (EsLaRed). The award consists of an engraved crystal globe and a US$20,000 honorarium. For more information about the award, visit: **`http://www.isoc.org/postel`**

ISOC is a non-profit organization founded in 1992 to provide leadership in Internet related standards, education, and policy. ISOC is dedicated to ensuring the open development, evolution, and use of the Internet for the benefit of people throughout the world. More information is available at: `http://www.isoc.org`

NRO Declaration on RPKI

The *Number Resource Organization* (NRO) recently declared: "Over several years, a set of mechanisms has been under development for digital certification of Internet number resources, through a so-called *Resource Public Key Infrastructure,* or "RPKI." Like other PKIs, the RPKI requires one or more root authorities, to act as so-called *trust anchors* for one or more certification hierarchies.[1]

The RPKI architecture has been designed to allow a number of trust anchor configurations involving: either a single trust anchor located at the root of a single certification hierarchy; a set of independent trust anchors to be located at the roots of several independent hierarchies; or a hybrid of these. The alternative models may have advantages and disadvantages in various dimensions including: operational efficiency; alignment with resource allocation hierarchies; centralisation vs distribution of functions; recognised global or regional authority; and, operational capacity of the respective host organisations.

The *Regional Internet Registries* (RIRs) believe that the optimal eventual RPKI configuration involves a single authoritative trust anchor. That configuration may not be achievable in the short-term and the details and timelines for its implementation will depend among other things on discussions within the RIRs' communities and dialogues with others including the *Internet Architecture Board* (IAB) and the *Internet Engineering Task Force* (IETF).

In the meantime, the RIRs have agreed to undertake pragmatic implementations of RPKI services based on interim trust anchor models, such as, self-signed trust anchors. All such implementations will comply with the overall RPKI architecture. The implementations will also have the ability to evolve into a single trust anchor model and to provide robust and fully operational (and inter-operational) services for those who wish to use them. The objective is for all RIRs to be ready to start issuing certificates by no later than January 1, 2011.

The RIRs will continue working with and receiving feedback from their respective communities and industry partners to ensure effective ongoing evolution of the RPKI system."

For more information about the NRO, see `http://www.nro.net/`

ARIN Hosts 4-byte ASN Wiki

The *American Registry for Internet Numbers* (ARIN) has created a wiki to focus on issues related to 4-byte *Autonomous System Numbers* (ASNs)[2]. This wiki provides a central repository for ongoing discussion and information exchange associated with 4-byte ASN topics and issues. The wiki can be found at: **www.get4byteasn.info**

Ongoing Internet growth is rapidly depleting the existing pool of 2-byte ASNs (65,536 numbers in total). As a result, the IETF has approved the expansion of AS Numbers from 2-bytes to 4-bytes, to include over 4 billion ASNs. Following a globally coordinated policy, ARIN and the other RIRs began assigning 4-byte ASNs by request in January 2007 and by default in January 2009. However, some routers do not support the use of these 4-byte ASNs.

ARIN has set up this wiki to help educate the community about 4-byte ASN operational issues, to help vendors understand how to provide 4-byte ASN support in their products and to help network operators find those products. A wide range of community stakeholders will be able to share and benefit from information contributed to the wiki. ARIN looks forward to participation from everyone, including users, ISPs, and vendors, with interest in this topic.

Upcoming Events

The *North American Network Operators' Group* (NANOG) will meet in Dearborn, Michigan, October 18–21. Following the NANOG meeting, the *American Registry for Internet Numbers* (ARIN) will meet in the same venue October 21–23. For more information see: **http://nanog.org** and **http://arin.net**

The *Internet Engineering Task Force* (IETF) will meet in Hiroshima, Japan, November 8–13, 2009 and in Anaheim, California, March 21–26, 2010. For more information see:
http://www.ietf.org/meeting/

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in Seoul, Korea, October 25–30, 2009 and Nairobi, Kenya, March 7–12, 2010, and in Brussels, Belgium, June 21–25, 2010. For more information, see: **http://icann.org/**

The *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will meet in Kuala Lumpur, Malaysia, February 23–March 5, 2010. For more information see:
http://www.apricot2010.net/

References

[1] Huston, Geoff, "Resource Certification," *The Internet Protocol Journal,* Volume 12, No. 1, March 2009.

[2] Huston, Geoff, "Exploring Autonomous Systems Numbers," *The Internet Protocol Journal,* Volume 9, No. 1, March 2006.

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

December 2009 Volume 12, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

In our last issue we brought you Part 1 of a two-part article on *Cloud Computing*. T. Sridhar introduced various aspects of cloud computing, including the rationale, underlying models, and infrastructures. Part 2, subtitled "Infrastructure and Implementation Topics," is included in the current issue. Cloud computing has received a great deal of press in recent months and continues to be an area of rapid development. I'm confident that we will have more articles about this topic in future editions of IPJ.

With this issue we start a new series of articles under the general heading "Protocol Basics." The idea is to present a series of in-depth tutorials on numerous protocols that are used every day on the Internet and in enterprise networks. The articles will cover protocol details as well as implementation, deployment, and usage scenarios. In some cases the articles will also summarize the "lessons learned" and present "best-practice" guidelines. To start the series, we asked Bill Stallings to give us an overview of the *Secure Shell* (SSH) Protocol. We invite you to send us suggestions for other protocols that you'd like to see covered in this series.

Today's Internet is a result of many years of technological development and innovative uses of the resulting infrastructure. Of equal importance has been many *policy* choices made over the years, ranging from what protocols to use to how to allocate finite resources such as the IPv4 address space. A new book, *Protocol Politics: The Globalization of Internet Governance,* explores some of this history. The book is examined in an extended review by Tom Vest.

Let me remind you that we will no longer be automatically extending your subscription when it expires. Please take a moment to check your expiration date (printed on the back of the journal for subscribers in the United States, and on the envelope for our international subscribers). Visit the "Subscriber Services" section of our webpage at **www.cisco.com/ipj** to update or renew your subscription. You can also contact us by e-mail to **ipj@cisco.com** regarding any aspect of your subscription.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Cloud Computing—A Primer
Part 2: Infrastructure and Implementation Topics

by T. Sridhar

Cloud computing is an emerging area that affects IT infrastructure, network services, and applications. In Part 1[0] of this two-part article, we introduced various aspects of cloud computing, including the rationale, underlying models, and infrastructures. In Part 2 we discuss specific infrastructure aspects of cloud computing in detail, specifically:

- Network Infrastructure
- Cloud-to-Cloud and Federation Considerations
- Security

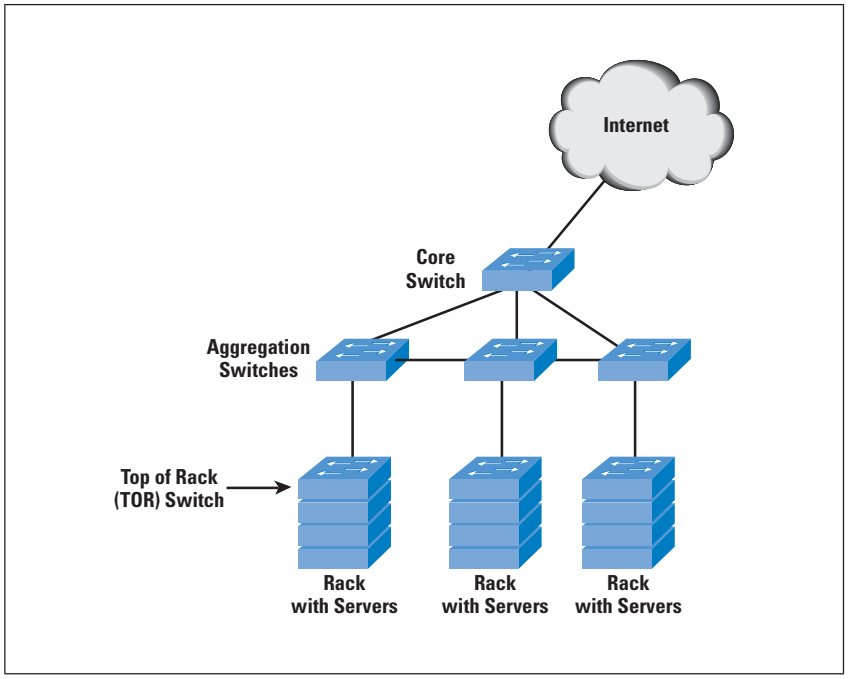
In addition, we will provide some perspective on select topics in cloud computing that have garnered interest. Remember that cloud computing is an emerging area where approaches to some of these topics are still evolving. In addition, although cloud computing is not intrinsically dependent upon virtualization, there is common agreement that virtualization (specifically, server virtualization) will be an integral part of cloud-computing solutions of the future. Consider the discussion in the following sections in this context.

Network Infrastructure

In a limited sense, the cloud can be treated as a large data center run by an external entity providing the capability for elasticity, on-demand resources, and per-usage billing. Data-center architecture often follows the common three-layer network topology of access, aggregation, and core networks with enabling networking elements (switches and routers). Consider the topology shown in Figure 4 of Part 1, reproduced here as Figure 0. The servers can be connected through a 1-Gbps link to a *Top of Rack* (TOR) switch, which in turn is connected through one or more 10-Gbps links to an aggregation *End of Row* (EOR) switch. The EOR switch is used for interserver connectivity across racks. The aggregation switches themselves are connected to core switches for connectivity outside the data center.

From a functional perspective, data-center server organization has often adopted a three-tier architecture (a specific case of an N-tier architecture). The three-tier functional architecture has a web or *Presentation Tier* on the front end, an *Application Tier* to perform the application and business-processing logic, and finally a *Database Tier* (to run the database management system), which is accessed by the Application Tier for its tasks (refer to Figure 1 on page 4).

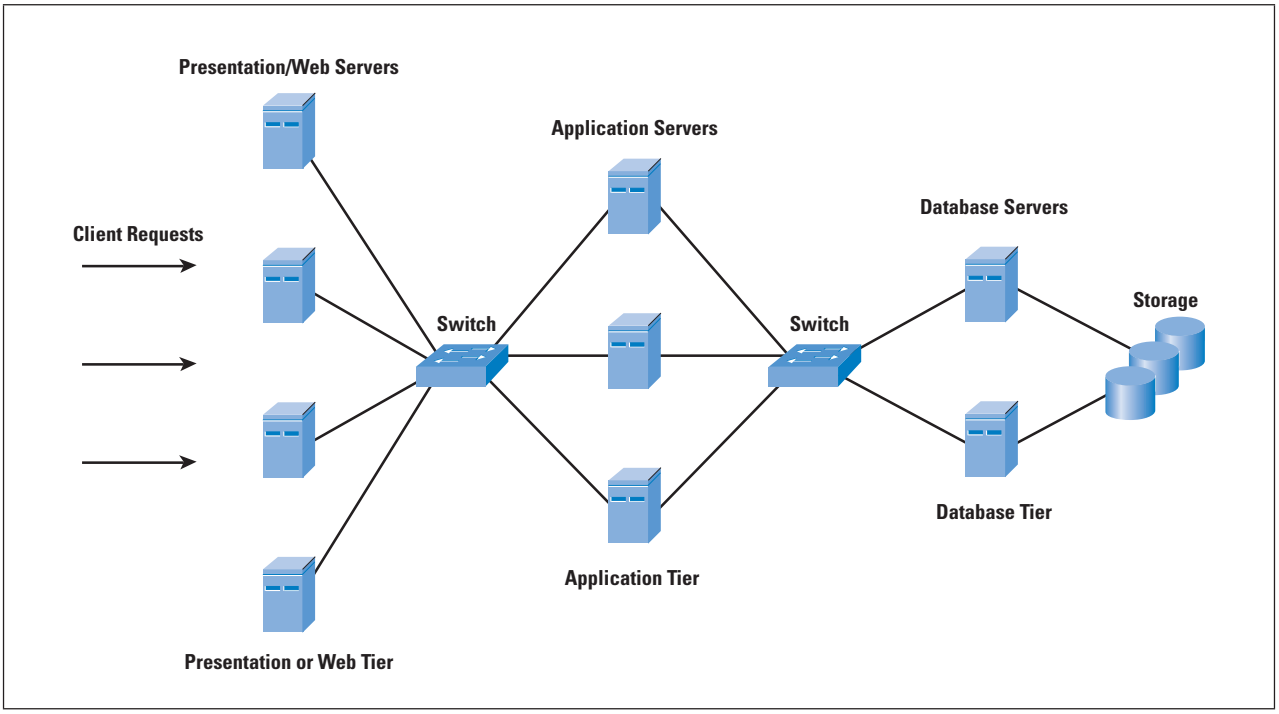
Figure 0: Example Data-Center Switch Network Architecture (from Part 1)



Although it is not necessary for each tier to be represented by its own physical servers (for example, you could have the Application and Database functions mapped into a single physical server), it is a common representation. The reason for this multitiered design is to control the connections and interactions, as well as for scaling and security. It is not uncommon for the Presentation Tier to be in a *Demilitarized Zone* (DMZ) while the other tiers are located deep inside the data center. Although all tiers could connect to storage for performing their functions, the Database Tier is the one with the maximum storage bandwidth requirements.

It follows that the server connectivity and the network topology for the cloud data centers might follow a similar organization. If you are an enterprise, you can perform the same business functions as before, but by using the external cloud. The choice of servers, software loads, and their interconnection will depend upon what you need to accomplish. In the following sections, we discuss how this design is handled in *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS), and *Software as a Service* (SaaS).

Figure 1: Three-Tier Functional Server Architecture



Data-Center Infrastructure Extension – IaaS

If the cloud is thus seen as an extension of the existing data center, IaaS as outlined in Part 1 is a natural fit. Here, you would specify the number of servers in each tier, load the appropriate server image (web, business logic, or database manager), and "connect" them (through a menu or *Application Programming Interface* [API] provided by the IaaS provider) by specifying the links between them. You can also specify the network connectivity at this time (more on this later). For an enterprise IT administrator, this model provides the greatest degree of control and, to an extent, a familiar operating topology. The cloud provider handles the elasticity by ensuring that the number of servers and switches is adequate for you to configure and connect in the specified topology. Per-use billing and on-demand resource addition and removal are also provided by the cloud provider. Note that if you have complete control, you also are responsible for security, application usage, and resource management.

PaaS and SaaS Infrastructure

In the case of PaaS, you transfer more control to your cloud service provider. The platform used to build the service you require can scale transparently without any of your involvement other than at the time of configuration. You do not need to understand the tier connectivity, bandwidth requirements, or how it all functions under the hood.

Cloud service providers can realize this function—often with a three-tier topology similar to that for traditional data centers. However, some of them have innovated to perform parts of the function differently. For example, the database functions may rely upon a model of *scaling out* (splitting the database across multiple servers) instead of *scaling up* (increasing the capability of the machine running the database servers). Their claim is that with clouds involving large amounts of data that you can partition and work on, it is easier to scale out than scale up. According to some cloud service providers, traditional relational databases are not suitable candidates for scale-out. Hence, some cloud vendors have provided their own database models and implementations—a common one being the type known as the *Key-Value database*.

SaaS vendors have the highest degree of control among the three models. The realization of the network topology can be similar to existing data centers and scale up or down according to the number of users that are added. However, because they offer a specific set of applications to the cloud users, their server and network topology is quite straightforward.

For the following discussions, we will use IaaS as the representative cloud service model, with a primary consideration being "cloud bursting"—how an existing IT infrastructure can take advantage of the power of the cloud when it needs additional resources. Note that some of the discussion might also be relevant for internal clouds. In addition, we will assume a virtualized server infrastructure for the IaaS cloud because this infrastructure provides a greater degree of flexibility for cloud service providers (Amazon being a key example).

Virtualization and Its Demands on Switching

In Part 1, we provided the context for a virtual switch within a physical server containing multiple virtual machines. There are some addressing and control factors to consider in this model. Consider a data center with 100 servers, each with 16 virtual machines but with one physical 10-Gbps Ethernet connection to the external switch from each physical machine. If we were to carry forward the model where each physical server is replaced with its virtual equivalent but still needs to be addressable (through a *Media Access Control* [MAC] layer address and an IP address), you would need 16 MAC and IP addresses for the virtual servers that now reside "on top" of the single physical link, for a total of 1600 addresses across all servers. This problem is exacerbated when you increase the number of VMs per server. Switching between MAC addresses belonging to the virtual machines is done by the virtual switch inside the server.

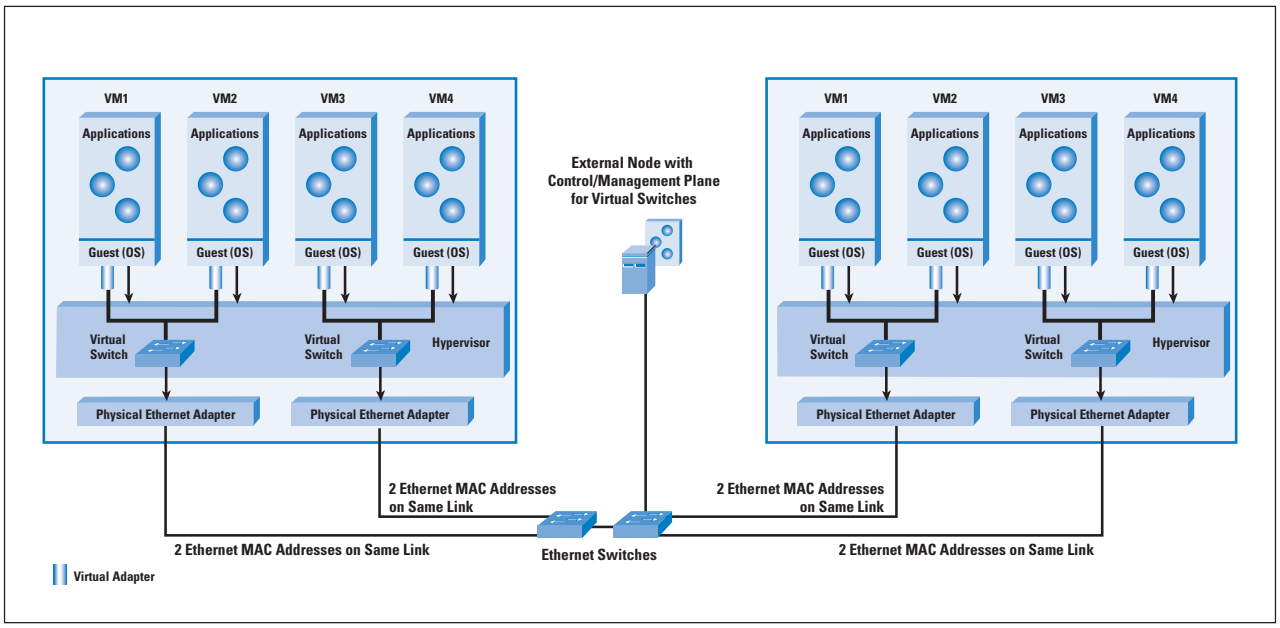
Consider the topology in Figure 2. The virtual switch treats the physical link as an uplink to the external physical switch. This intramachine *Virtual Machine* (VM) switch with an uplink to the external switch is completely in line with access and aggregation switch topologies where the access layer is subsumed inside the server. Note that each physical host can have more than one virtual switch to support greater logical segmentation. In such cases, it is common for each of the virtual switches to have its own physical uplink to the external Ethernet switch.

The virtual switch does not need to learn MAC addresses like a traditional switch—it assumes that all destination-unknown frames should be forwarded over the physical link (or uplink to the physical switch). In addition, it switches traffic between the intramachine VMs according to policy. For example, you could prohibit two VMs on the same machine from communicating with each other by configuring an access control list on the virtual switch. The VMs may all be on the same or on different VLANs. Broadcasts and intra-VLAN traffic are forwarded according to the rules for each VLAN. In effect, the virtual switch is a simple function that is used for aggregation and access control within a physical server containing VMs.

Management of these virtual switches can follow an aggregation model—where multiple virtual switches are managed through an external node (physical machine or VM), as shown in Figure 2. This external node provides the management view on behalf of the switches. Often, the external node can run control-plane protocols for Layer 2/3 functions, in effect appearing like a control or management plane with multiple data-plane instances (the virtual switches). When VMs need to be migrated to other physical servers, this separation of control- or management-plane functions permits easier migration of policy and access lists.

Virtual switches do have some disadvantages. Inter-VM traffic within the same machine is not visible to the network and cannot be subject to appropriate monitoring by network administrators. The IEEE is discussing approaches to providing external network switches the visibility into the intra-VM traffic. The options include “hair pinning,” where inter-VM traffic would still be carried over to an external switch and brought back to the same physical server.

Figure 2: Virtual Switch Aggregation and Management by External Node



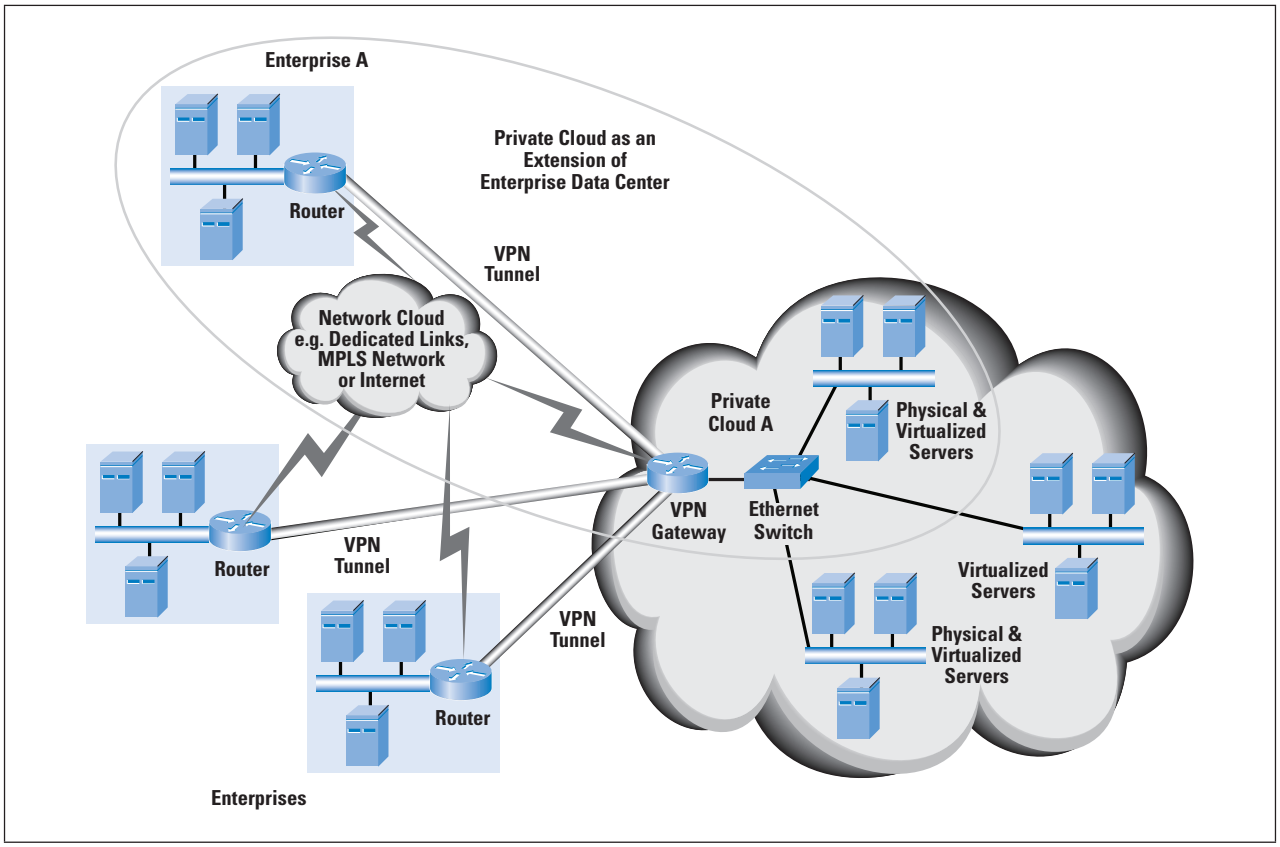
IaaS Private Clouds

Consider an IaaS cloud to which an enterprise connects to augment its server capacity for a limited period of time. Assume that the enterprise uses a `10.x.x.x` private addressing scheme for all its servers because they are internal to the enterprise. It would be ideal if the additional servers provided by the IaaS cloud were part of the same addressing scheme (the `10.x.x.x` scheme). As shown in Figure 3, the IaaS cloud service provider has partitioned a portion of its public cloud to realize a private cloud for enterprise A. The private cloud is reachable as a LAN extension to the servers in enterprise A's data center.

How is this reachability realized? A secure *Virtual Private Network* (VPN) tunnel is first established between the enterprise data center and the public cloud. This tunnel uses public IP addresses to establish the site-to-site VPN connection. The VPN gateway on the cloud service provider side uses multiple contexts—each context corresponding to a specific private cloud. Traffic from enterprise A is decrypted and forwarded over to an Ethernet switch to the private cloud for enterprise A. A server on enterprise A's internal data center sees a server on private cloud A to be on the same network.

In practice, data-center servers might be segmented into their own VLANs or IP networks according to policy and applications. The configuration and forwarding policies on the private cloud end would reflect this segmentation as well.

Figure 3: Example of Private Clouds



The following are some possible evolution scenarios for this scheme:

- *Automation of the VPN connection between the enterprise and cloud service provider:* This automation can be done through a management system responsible for the cloud bursting and server augmentation. The system sets up the VPN tunnels and configures the servers on the cloud service provider end. The management system is set up and operated by the cloud service provider.
- *Integration of the VPN functions with the site-to-site VPN network functions from service providers:* For example, service providers offer MPLS Layer 3 VPNs and Layer 2 VPNs (also known as *Virtual Private LAN Service*, or VPLS) as part of their offerings. Enterprise and cloud service providers could be set up to use these network services.
- *Cloud service providers using multiple data centers:* In such a situation, a VPLS-like service can be used to bridge the individual data centers, providing complete transparency from the enterprise side about the location of the cloud servers.

CloudNet is an example of a framework being developed by AT&T Labs and the University of Massachusetts at Amherst to address the latter two scenarios.

Layer 2 versus Layer 3 Connectivity for Cloud Networks

Enterprises and vendors follow some guidelines regarding where to use Layer 2 (switching) and Layer 3 (routing) in the network. Layer 2 is the simpler mode, where the Ethernet MAC address and *Virtual LAN* (VLAN) information are used for forwarding. The disadvantage of Layer 2 networks is scalability. When we use Layer 2 addressing and connectivity in the manner specified previously for IaaS clouds, we end up with a flat topology, which is not ideal when there are a large number of nodes. The option is to use routing and subnets—to provide segmentation for the appropriate functions at the cost of forwarding performance and network complexity.

VM migration introduces its own set of problems. The most common scenario is when a VM is migrated to a different host on the same Layer 2 topology (with the appropriate VLAN configuration). Consider the case where a VM with open *Transmission Control Protocol* (TCP) connections is migrated. If live migration is used, TCP connections will not see any downtime except for a short "hiccup." However, after the migration, IP and TCP packets destined for the VM will need to be resolved to a different MAC address or the same MAC address but now connected to a different physical switch in the network so that the connections can be continued without disruption. Proposed solutions include an unsolicited *Address Resolution Protocol* (ARP) request from the migrated VM so that the switch tables can be updated, a pseudo-MAC address for the VM that is externally managed (defined in research work being done at the University of California at San Diego), and so on.

With VPLS and similar Layer 2 approaches, VM migration can proceed as before—across the same Layer 2 network. Alternatively, it may be less complex to "freeze" the VM and move it across either a Layer 2 or Layer 3 network with the TCP connections having to be torn down by the counterpart(s) communicating with the VM. This scenario is not a desired one from an application availability consideration, but it can lower complexity.

Cloud Federation

Thus far we have considered the situation of data centers that are owned or run by the same cloud services provider. Connectivity between the data centers to provide the vision of "one cloud" is completely within the control of the cloud service provider.

There may be situations where an organization or enterprise needs to be able to work with multiple cloud providers because of migration from one cloud service to another, merger of companies working with different cloud providers, cloud providers who provide best-of-class services, and so on. Cloud interoperability and the ability to share various types of information between clouds become important in such scenarios. Although cloud service providers might see less urgency for any interoperability, enterprise customers will see a need to push them in that direction.

This broad area of cloud interoperability is sometimes known as *cloud federation*. One definition of cloud federation as proposed by Reuven Cohen of Enomaly follows:

> "Cloud federation manages consistency and access controls when two or more independent geographically distributed clouds share either authentication, files, computing resources, command and control, or access to storage resources."

The following are some of the considerations in cloud federation:

- An enterprise user wishing to access multiple cloud services would be better served if there were just a single sign-on scheme. This scheme may be implemented through an authentication server maintained by an enterprise that provides the appropriate credentials to the cloud service providers. Alternatively, a central trusted authentication server to which all the cloud services interface could be used.
- Computing and storage resources may be orchestrated through the individual enterprise or through an interoperability scheme established between the cloud providers (through a federation agreement, for example). Files may need to be transferred, services invoked, and computing resources added or removed in a useful and transparent manner. A related area is VM migration and how it can be done transparently and reliably. The *Desktop Management Task Force* (DMTF) has released a specification called the *Open Virtualization Format* (OVF) for describing a VM. It can be reasonably assumed that the payload for VM migration will be in the OVF format so that it can be interpreted across multiple vendor offerings. In effect, cloud federation has to provide transparent workload orchestration between the clouds on behalf of the enterprise user.
- Connectivity between clouds includes Layer 2 versus Layer 3 considerations and secure tunnel technologies that need to be agreed upon. Consistency and a common understanding are required irrespective of the model or technologies.
- An often-ignored concern for cloud confederation is charging or billing and reconciliation. Management and billing systems need to work together for cloud federation to be a viable option. This reality is underlined by the fact that clouds rely on per-use billing. Cloud service providers might need to look closely at telecom service provider business models for peering arrangements as a possible starting point.

Cloud federation is a relatively new area in cloud computing. It is likely that standards bodies will first need to agree upon a set of requirements before the service interfaces can be defined and subsequently realized. Provider and vendor innovation will also significantly affect this area—in fact, cloud service operators are likely to establish peering relationships and start addressing this area even before the standards bodies.

Security

As indicated in Part 1, the biggest deterrent for IT managers from venturing into cloud computing is the problem of security and loss of control. Before considering a move to a cloud service provider, enterprises need to consider some of the following security topics:

- The cloud service provider's security processes will need to be as good as or better than the processes that the enterprise uses. An audit of the vendor's processes will need to be done periodically, possibly including patches and security updates for the individual components that are used. For example, in an IaaS scenario with some preconfigured images of operating systems and applications, the cloud service provider should have the latest patches applied on the individual components.
- Infrastructure and data isolation must be assured between multiple tenants of the cloud service provider. This requirement is complicated because it is closely intertwined with the business model used by the cloud provider. For example, an IaaS provider might provide multiple tenants with VMs running on the same physical machine. Depending upon the type of work that is to be executed on the cloud, this setup may or may not be acceptable to a cloud user. In such cases, the cloud service provider should have the ability to provide separate physical servers for specific customers (and bill appropriately).
- In cases where a hypervisor and VMs are used, the hypervisor should be treated as an operating system and have the latest security patches applied to it. Security patches and updates are also essential for paravirtualized operating systems used in the VMs.
- Security functions can run as virtual appliances over hypervisors in a cloud environment. Thus it is possible for cloud users in an IaaS environment to load and configure their own firewall or other security virtual appliance to run within the cloud. The software images used for these virtual appliances need to be managed and patched similar to the way the OS, hypervisor, and other applications are managed and patched.
- Logging and audit trails for applications are important for enterprises to understand both application performance and security gaps. Cloud services providers should enable access to their application monitoring and profiling tools, where applicable.
- Authentication mechanisms ("You are who you say you are") are required at both ends of the connection—at the cloud user and cloud service provider levels. The cloud user and operator must agree upon schemes such as authentication with digital certificates and certificate authorities.

- Configuration and updates to the network infrastructure must be audited and tracked. For example, incorrect VLAN configuration on the switches can result in undesired traffic patterns between physical machines and computing resources. It would be useful to log and audit the configuration records for proper security and uptime.
- Because the cloud service is exposed to the outside world, the cloud infrastructure should support security functions such as intrusion detection and prevention, firewalling to prevent disallowed traffic, and *Denial of Service* (DoS) prevention. The cloud service is vulnerable to *Distributed Denial of Service* (DDoS) attacks—which can effectively choke its access lines, resulting in cloud users being locked out of the cloud service. Network-based DDoS prevention is a possible solution—with one of the techniques involving distribution of the cloud infrastructure to specific geographic areas and the ability to redirect cloud users in case of DDoS lockouts.

Virtualization and Security

Two options are under discussion for security in the context of virtualization. Both are useful in building out security-enabled cloud infrastructures. One option involves plug-ins to the hypervisor so that packets destined to the VMs are captured and processed by the security plug-ins. This setup enables application of security functions to the packet before it gets to the VMs. A second option is to make a specific VM handle the security functions without changing or adding to the hypervisor. The hypervisor plug-in option has the advantage of performance and initial isolation, whereas the separate VM option has the advantage of keeping the hypervisor simple and extrapolating the model that exists in physical server infrastructure. Note that these options are not mutually exclusive.

VM migration is another area where security is an important consideration. The hypervisor is responsible for the two-way communication, with the hypervisor on the destination physical machine to accomplish the migration. It is important that the connection between the source and destination hypervisors is authenticated and encrypted during the course of this migration. In addition, VM migration introduces the possibility of a DoS attack because a rogue hypervisor could overwhelm a destination machine by migrating a large number of VMs to the destination machine. Policies and logic are required at the hypervisor level to ensure that these vulnerabilities are addressed. In addition, network-based throttling might be required so that live migration does not cause congestion, which might happen if a large number of VMs need to be migrated to a destination machine at the same time.

Standards Bodies Involved in Cloud Computing

Numerous standards bodies are involved in cloud computing, addressing aspects of interoperability, virtualization migration formats, and security. Some of the organizations involved have established liaisons with the other *Standards Development Organizations* (SDOs) so that there is no duplication of effort.

The *Desktop Management Task Force* (DMTF) has specified a portable format for packaging the software to run as a VM. Known as the *Open Virtualization Format* (OVF), this package format is seeing increased use. The VM can be written onto a disk or external storage and can be moved from one physical machine to another. The DMTF has also formed a group called the *Open Cloud Standards Incubator,* which focuses on standardizing the interactions between cloud environments, including the development of resource management, packaging formats, and security.

The *Cloud Security Alliance* (CSA) is a new group formed to address security aspects of cloud computing with a focus on security assessment and management. The initial part of the effort is on developing an *Audit, Assertion, Assessment and Assurance* (API) set (A6).

The *Organization for the Advancement of Structured Information Standards* (OASIS) sees cloud computing as an extension of the *Service-Oriented Architecture* (SOA) used today in IT environments. The areas for standardization include security and policy, content format control, registry and directory standards, as well other SOA methods.

The *Storage Networking Industries Association* (SNIA) has a Cloud Storage *Technical Working Group* (TWG) that works on storage-related problems related to implementation in a cloud. The TWG has developed an interface known as the *Cloud Data Management Interface* (CDMI), which clients will use for control and configuration of the cloud.

Some Perspectives on Cloud Computing

In this section we outline and provide some perspective on cloud-computing topics that have seen interest (and some heated discussion). This list is not intended to be comprehensive but to provide a quick snapshot. Though this section has a degree of subjectivity, it is directed only to providing a broader perspective.

- *Cloud computing and SOA:* Some view cloud computing as a specific deployment case of an SOA—and this view is more popular than the one that says that cloud computing is the evolution of SOA. David Linthicum outlines that these views are complementary in that cloud-computing services will most likely be defined through SOA. IaaS provides a new variant because you can now access raw compute and storage resources as a service. Independent of the argument that "We have seen this before," there is value to defining and invoking available services in the cloud.

- *Server virtualization schemes:* Comparisons are sometimes made based on how vendor products approach virtualization—type 1 versus type 2—and full versus paravirtualization. These approaches have pros and cons. The final decision often hinges on total costs, so it might be useful to move forward from this debate. Incidentally, vendors provide several useful tools for VM backup, recovery, fault tolerance, load management, and so on, and these tools work equally well for the various approaches to virtualization. It may be argued that these tools and features such as VM migration and the associated costs are more useful areas for comparison.
- *Other types of virtualization:* This article has deliberately omitted discussion of other types of virtualization, including desktop, application, and presentation virtualization. Some of these schemes (server-hosted desktop virtualization is one example) are affected by the cloud, specifically in the areas of network connectivity, authentication, and quality of experience. In general, any thin-client experience is affected by the cloud or data center because most of the work is done at the servers. From a cloud perspective, these types of virtualization schemes are considered to be applications that need to run reliably and consistently.
- *Data transfer and network bandwidth:* IaaS has provided a flexible model, in which you are charged based on compute power usage, storage consumed, and the duration of usage. However, there is another important factor—data needs to be sent back and forth between the cloud user and cloud service provider. Several IaaS providers charge for the amount of data transferred over the link. These charges can quickly add up if your applications are very chatty and require a lot of back-and-forth data traffic. Another concern here is the amount of time the initial upload or download can consume—for example, when you want to move a large number of your files to the IaaS provider's storage, you can tie up the link for hours. In fact, one provider has a model where cloud users can send storage media through a postal or package service for upload to the cloud provider's storage arrays.
- *WAN acceleration for the cloud:* Continuing on the previous point, chatty protocols and applications can benefit from WAN acceleration devices that can be used on both ends of a WAN link to cache and locally serve enterprise applications. These devices are not specific to the cloud—they have been used for several years for application performance improvement when a WAN link is involved. Recently, virtual network appliances for WAN acceleration are seeing deployment—here the WAN acceleration is performed by an individual VM instead of a dedicated appliance.
- *VM migration:* This article outlined some of the concerns with VM migration with respect to Layer 2 and Layer 3 topologies. Another consideration is the amount of data that needs to be moved when a VM is migrated across a network. It can potentially be in the range of gigabytes, depending upon the VM and the included operating environment.

Live migration implements this transfer in an incremental fashion so that the demand on the network is spread out. However, snapshot migration (where a VM is suspended or frozen and migrated over the network in full) can cause a surge of data on the network, leading to application performance problems for other VMs and physical machines. Throttling the amount of data that can be sent in a specific period of time, bandwidth reservation and policing at the intermediate network devices is highly desirable in such situations.

- *Management:* The current management paradigms for the cloud components are quite discrete and provide a strong level of control. For example, it is possible to log in to the *Command-Line Interface* (CLI) of a specific switch in the data center for configuration and control of the switch parameters. Similarly, it is possible to use the management console provided by the virtualization vendor to configure individual parameters for the hypervisors and VMs (for example, when to initiate VM migration to a different physical machine). Efforts are being made to unify management schemes not just through partnerships between the individual vendors but also with machine-readable interfaces (*Extensible Markup Language* [XML] being a baseline) across the multiple types of equipment and software in the cloud. Enterprise users are unlikely to accept point solutions or tools that require extensive user interaction in the long term.
- *Energy considerations:* One of the benefits of virtualization is the use of a lower number of physical servers to realize a specific function. It follows that overall energy consumption would be reduced because you have fewer servers. Although this fact may indeed be true, it would be good to characterize and monitor the effective energy savings for a specific application ("Your mileage may vary"). For example, the load on each server and the associated I/O and storage traffic may lead to higher power requirements on an individual server basis. Other considerations include the hardware infrastructure of the cloud data center because the power and cooling assumptions per rack are based on average server load.
- *Legal and regulatory considerations:* James Urquhart has compiled a set of criteria for workload migration across multiple locations, one of which is "Follow the law." Consider the case of a cloud services provider or operator that has data centers in two separate countries. The operator might use the data centers for workload migration as well as load balancing. A problem might arise if the laws in one of the countries impose limitations on what can and cannot be done at the data center. Scenarios include access to all data stored at this data center by authorities or the ability to examine all transactions on the wire at the data center. Workload migration policy statements have to be provided to cloud users so that they understand what they are signing up to. Alternatively, they might be provided the ability to set preferences for workload migration. This area is potentially worrisome, so it is important that cloud users are aware of their specific situation.

Conclusion

This article has served as a vendor-neutral primer to the area of cloud computing. In Part 1, we provided an introduction to the still-evolving area of cloud computing, including the technologies and some deployment concerns. In Part 2, we provided a more detailed look at the networking factors in the cloud, security aspects, and cloud federation. We also highlighted some areas that are seeing increased attention with cloud-computing proponents and vendors.

The area of cloud computing is very dynamic and offers scope for innovative technologies and business models. Ongoing work with respect to solutions is substantial, in the vendor research labs and product development organizations as well as in academia. It is clear that cloud computing will see significant advances and innovation in the next few years.

For Further Reading (see Part 1 for additional references)

[0] T. Sridhar, "Cloud Computing: A Primer, Part 1: Models and Technologies," *The Internet Protocol Journal,* Volume 12, No. 3, September 2009.

[1] "Building Data-Centric n-Tier Enterprise Systems," PowerVision white paper, **http://www.powervision.com/html/news/n_tier_arch.pdf**

[2] "Networking in the (Storm) Clouds," Michael Morris, **http://www.networkworld.com/community/node/43872**

[3] "Is the Relational Database Doomed?" Tony Bain, **http://www.readwriteweb.com/enterprise/2009/02/is-the-relational-database-doomed.php**

[4] "Cisco VN-Link: Virtualization-Aware Networking," **http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns892/ns894/white_paper_c11-525307_ps9902_Products_White_Paper.html**

[5] IEEE work (in progress) on Virtual Ethernet Bridging—VEPA and VN-Tag approaches—search for "VEPA" and "VN-Tag" in the directory at: **http://www.ieee802.org/1/files/public/docs2009**

[6] "PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric," Mysore et al., **http://ccr.sigcomm.org/online/?q=node/503**

[7] "VL2: A Scalable and Flexible Data Center Network," Greenberg et al., **http://ccr.sigcomm.org/online/?q=node/502**

[8] "The Case for Enterprise-Ready Virtual Private Clouds," Wood et al., **http://www.usenix.org/event/hotcloud09/tech/full_papers/wood.pdf**

[9] "Solving the Problem of Cloud Interoperability," Reuven Cohen,
http://reuvencohen.sys-con.com/node/798504

[10] "Security Guidance for Critical Areas of Focus in Cloud Computing," Cloud Security Alliance,
http://www.cloudsecurityalliance.org/guidance/csaguide.pdf

[11] Rational Survivability Blog, Chris Hoff's blog on various topics, including cloud security,
http://www.rationalsurvivability.com/blog/

[12] "Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds" by Ristenpart, et al,
http://people.csail.mit.edu/tromer/papers/cloud-sec.pdf

[13] "Empirical Exploitation of Live Virtual Machine Migration," Oberheide et al.,
http://www.eecs.umich.edu/techreports/cse/2007/CSE-TR-539-07.pdf

[14] List of and links to cloud standards organizations,
http://cloud-standards.org/wiki/index.php?title=Main_Page/

[15] "Open Virtualization Format Specification," DMTF,
http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf

[16] "SOA cloud computing relationship leaves some folks in a fog," David Linthicum,
http://www.gcn.com/Articles/2009/03/09/Guest-commentary-SOA-cloud.aspx

[17] "Is your data center ready for virtualization," Eaton white paper,
http://i.zdnet.com/whitepapers/Eaton_Is_your_data_center_ready_for_virtualization.pdf

[18] "The great paradigm shift of cloud computing is not self-service," James Urquhart, http://news.cnet.com/8301-19413_3-10127654-240.html?tag=mncol;txt

T. SRIDHAR received his BE in Electronics and Communications Engineering from the College of Engineering, Guindy, Anna University, Madras, India, and his Master of Science in Electrical and Computer Engineering from the University of Texas at Austin. He can be reached at TSridhar@leitnet.com

Protocol Basics: Secure Shell Protocol

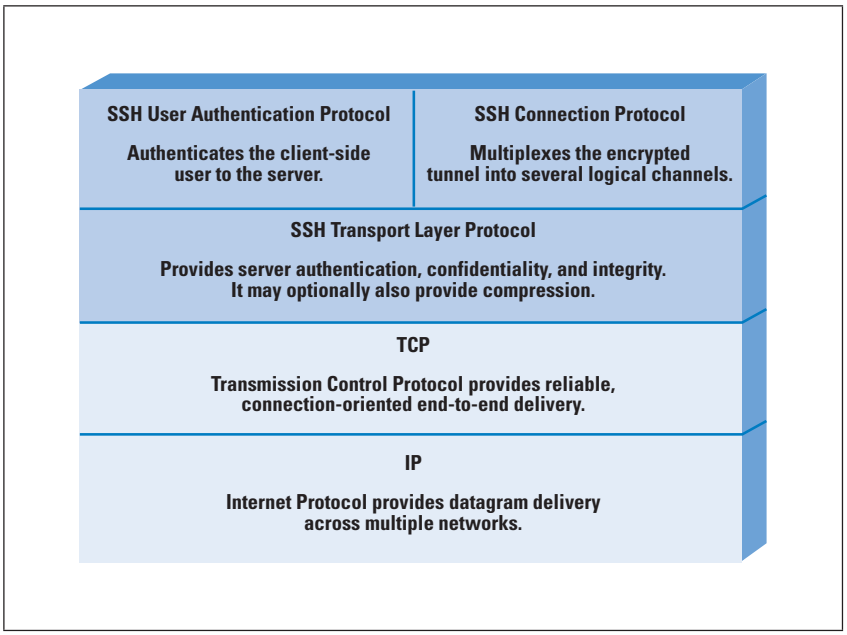
by William Stallings

Secure Shell (SSH) Protocol is a protocol for secure network communications designed to be relatively simple and inexpensive to implement. The initial version, SSH1, focused on providing a secure remote logon facility to replace Telnet and other remote logon schemes that provided no security[4]. SSH also provides a more general client-server capability and can be used to secure such network functions as file transfer and e-mail. A new version, SSH2, provides a standardized definition of SSH and improves on SSH1 in numerous ways. SSH2 is documented as a proposed standard in RFCs 4250 through 4256 [1–3], [5–8].

SSH client and server applications are widely available for most operating systems. It has become the method of choice for remote login and X tunneling and is rapidly becoming one of the most pervasive applications for encryption technology outside of embedded systems. SSH is organized as three protocols that typically run on top of TCP (Figure 1):

- *Transport Layer Protocol:* Provides server authentication, data confidentiality, and data integrity with forward secrecy (that is, if a key is compromised during one session, the knowledge does not affect the security of earlier sessions); the transport layer may optionally provide compression
- *User Authentication Protocol:* Authenticates the user to the server
- *Connection Protocol:* Multiplexes multiple logical communications channels over a single underlying SSH connection

Figure 1: SSH Protocol Stack



Transport Layer Protocol

Server authentication occurs at the transport layer, based on the server possessing a public-private key pair. A server may have multiple host keys using multiple different asymmetric encryption algorithms. Multiple hosts may share the same host key. In any case, the server host key is used during key exchange to authenticate the identity of the host. For this authentication to be possible, the client must have presumptive knowledge of the server public host key. RFC 4251 dictates two alternative trust models that can be used:

1. The client has a local database that associates each host name (as typed by the user) with the corresponding public host key. This method requires no centrally administered infrastructure and no third-party coordination. The downside is that the database of name-to-key associations may become burdensome to maintain.
2. The host name-to-key association is certified by a trusted *Certification Authority* (CA). The client knows only the CA root key and can verify the validity of all host keys certified by accepted CAs. This alternative eases the maintenance problem, because ideally only a single CA key needs to be securely stored on the client. On the other hand, each host key must be appropriately certified by a central authority before authorization is possible.

Figure 2: SSH Transport Layer Protocol Packet Exchanges

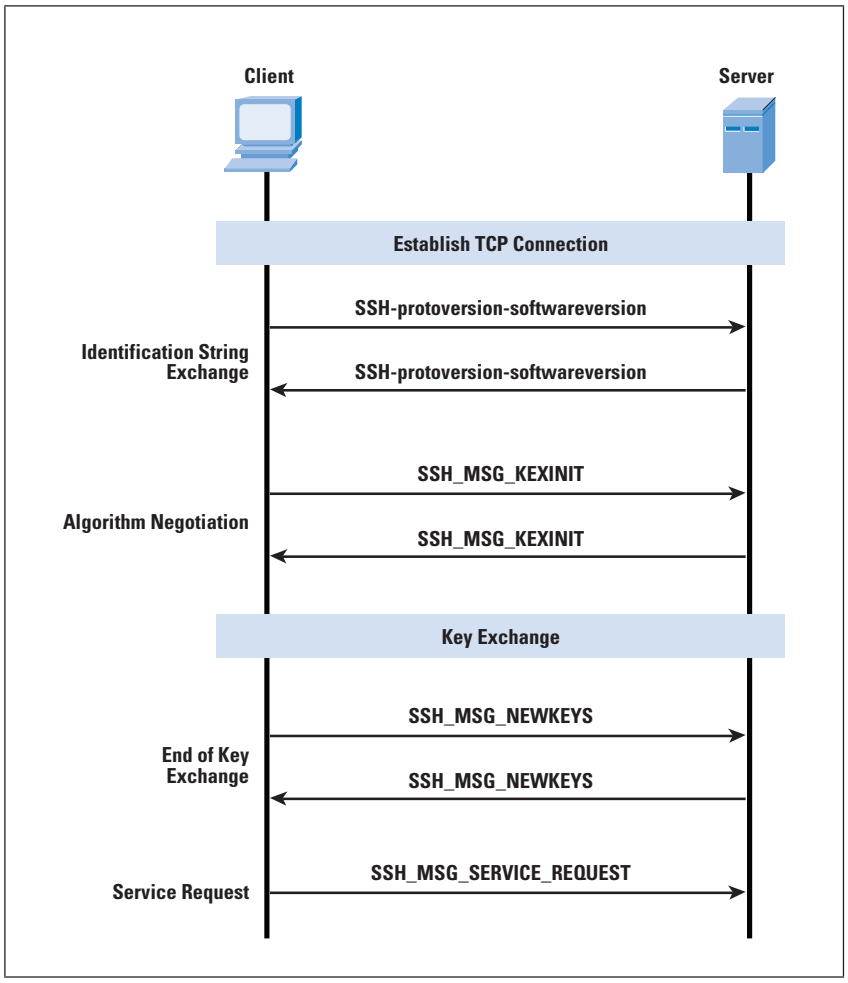
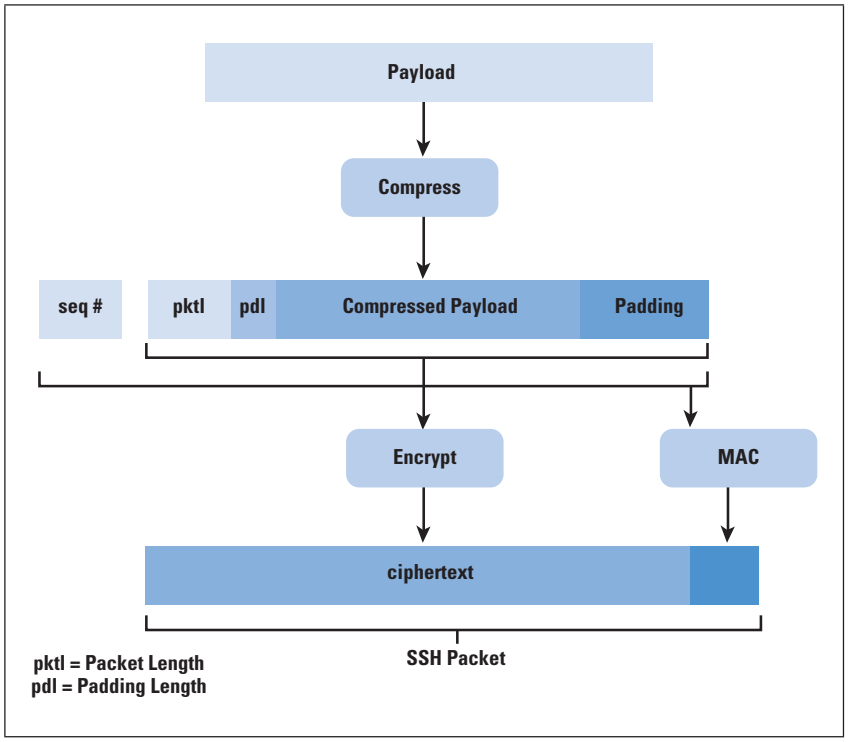


Figure 2 illustrates the sequence of events in the SSH Transport Layer Protocol. First, the client establishes a TCP connection to the server with the TCP protocol and is not part of the Transport Layer Protocol. When the connection is established, the client and server exchange data, referred to as packets, in the data field of a TCP segment. Each packet is in the following format (Figure 3):

- *Packet length:* Packet length is the length of the packet in bytes, not including the packet length and Message Authentication Code (MAC) fields.
- *Padding length:* Padding length is the length of the random padding field.
- *Payload:* Payload constitutes the useful contents of the packet. Prior to algorithm negotiation, this field is uncompressed. If compression is negotiated, then in subsequent packets this field is compressed.
- *Random padding:* After an encryption algorithm is negotiated, this field is added. It contains random bytes of padding so that that total length of the packet (excluding the MAC field) is a multiple of the cipher block size, or 8 bytes for a stream cipher.
- *Message Authentication Code* (MAC): If message authentication has been negotiated, this field contains the MAC value. The MAC value is computed over the entire packet plus a sequence number, excluding the MAC field. The sequence number is an implicit 32-bit packet sequence that is initialized to zero for the first packet and incremented for every packet. The sequence number is not included in the packet sent over the TCP connection.

Figure 3: SSH Transport Layer Protocol Packet Formation



After an encryption algorithm is negotiated, the entire packet (excluding the MAC field) is encrypted after the MAC value is calculated.

The SSH Transport Layer packet exchange consists of a sequence of steps (Figure 2). The first step, the *identification string exchange*, begins with the client sending a packet with an identification string of the form:

SSH-protoversion-softwareversion SP comments CR LF

where SP, CR, and LF are space character, carriage return, and line feed, respectively. An example of a valid string is **`SSH-2.0-billsSSH_3.6.3q3<CR><LF>`**. The server responds with its own identification string. These strings are used in the Diffie–Hellman key exchange.

Next comes *algorithm negotiation*. Each side sends an **`SSH_MSG_KEXINIT`** containing lists of supported algorithms in the order of preference to the sender. Each type of cryptographic algorithm has one list. The algorithms include key exchange, encryption, MAC algorithm, and compression algorithm. Table 1 shows the allowable options for encryption, MAC, and compression. For each category, the algorithm chosen is the first algorithm on the client's list that is also supported by the server.

Table 1: SSH Transport Layer Cryptographic Algorithms

Cipher	
3des-cbc*	Three-key Triple Digital Encryption Standard (3DES) in Cipher-Block-Chaining (CBC) mode
blowfish-cbc	Blowfish in CBC mode
twofish256-cbc	Twofish in CBC mode with a 256-bit key
twofish192-cbc	Twofish with a 192-bit key
twofish128-cbc	Twofish with a 128-bit key
aes256-cbc	Advanced Encryption Standard (AES) in CBC mode with a 256-bit key
aes192-cbc	AES with a 192-bit key
aes128-cbc**	AES with a 128-bit key
Serpent256-cbc	Serpent in CBC mode with a 256-bit key
Serpent192-cbc	Serpent with a 192-bit key
Serpent128-cbc	Serpent with a 128-bit key
arcfour	RC4 with a 128-bit key
cast128-cbc	CAST-128 in CBC mode

MAC Algorithm	
hmac-sha1*	HMAC-SHA1; Digest length = Key length = 20
hmac-sha1-96**	First 96 bits of HMAC-SHA1; Digest length = 12; Key length = 20
hmac-md5	HMAC-SHA1; Digest length = Key length = 16
hmac-md5-96	First 96 bits of HMAC-SHA1; Digest length = 12; Key length = 16

Compression Algorithm	
none*	No compression
zlib	Defined in RFCs 1950 and 1951

** = Required*
*** = Recommended*

The next step is *key exchange*. The specification allows for alternative methods of key exchange, but at present only two versions of Diffie–Hellman key exchange are specified. Both versions are defined in RFC 2409 and require only one packet in each direction. The following steps are involved in the exchange. In this, C is the client; S is the server; p is a large safe prime; g is a generator for a subgroup of GF(p); q is the order of the subgroup; `V_S` is the S identification string; `V_C` is the C identification string; `K_S` is the S public host key; `I_C` is the C `SSH_MSG_KEXINIT` message; and `I_S` is the `S SSH_MSG_KEXINIT` message that was exchanged before this part began. The values of p, g, and q are known to both client and server as a result of the algorithm selection negotiation. The hash function hash() is also decided during algorithm negotiation.

1. C generates a random number x ($1 < x < q$) and computes $e = g^x \bmod p$. C sends e to S.
2. S generates a random number y ($0 < y < q$) and computes $f = g^y \bmod p$. S receives e. It computes $K = e^y \bmod p$, H = hash(V_C || V_S || I_C || I_S || K_S || e || f || K), and signature s on H with its private host key. S sends (K_S || f || s) to C. The signing operation may involve a second hashing operation.
3. C verifies that `K_S` really is the host key for S (for example, using certificates or a local database). C is also allowed to accept the key without verification; however, doing so will render the protocol insecure against active attacks (but may be desirable for practical reasons in the short term in many environments). C then computes $K = f^x \bmod p$, H = hash(V_C || V_S || I_C || I_S || K_S || e || f || K), and verifies the signature s on H.

As a result of these steps, the two sides now share a master key K. In addition, the server has been authenticated to the client, because the server has used its private key to sign its half of the Diffie–Hellman exchange. Finally, the hash value H serves as a session identifier for this connection. When computed, the session identifier is not changed, even if the key exchange is performed again for this connection to obtain fresh keys.

The *end of key exchange* is signaled by the exchange of `SSH_MSG_NEWKEYS` packets. At this point, both sides may start using the keys generated from K, as discussed subsequently.

The final step is *service request*. The client sends an `SSH_MSG_SERVICE_REQUEST` packet to request either the User Authentication or the Connection Protocol. Subsequent to this request, all data is exchanged as the payload of an SSH Transport Layer packet, protected by encryption and MAC.

The keys used for encryption and MAC (and any needed IVs) are generated from the shared secret key *K*, the hash value from the key exchange *H*, and the session identifier, which is equal to *H* unless there has been a subsequent key exchange after the initial key exchange. The values are computed as follows:

- Initial IV client to server: HASH(*K* || *H* || "A" || session_id)
- Initial IV server to client: HASH(*K* || *H* || "B" || session_id)
- Encryption key client to server: HASH(*K* || *H* || "C" || session_id)
- Encryption key server to client: HASH(*K* || *H* || "D" || session_id)
- Integrity key client to server: HASH(*K* || *H* || "E" || session_id)
- Integrity key server to client: HASH(*K* || *H* || "F" || session_id)

where HASH() is the hash function determined during algorithm negotiation.

User Authentication Protocol

The *User Authentication Protocol* provides the means by which the client is authenticated to the server.

Three types of messages are always used in the User Authentication Protocol. Authentication requests from the client have the format:

byte	**`SSH_MSG_USERAUTH_REQUEST (50)`**
string	username
string	service name
string	method name
....	method-specific fields

where *username* is the authorization identity the client is claiming, *service name* is the facility to which the client is requesting access (typically the SSH Connection Protocol), and *method name* is the authentication method being used in this request. The first byte has decimal value 50, which is interpreted as **`SSH_MSG_USERAUTH_REQUEST`**.

If the server either rejects the authentication request or accepts the request but requires one or more additional authentication methods, the server sends a message with the format:

byte	**`SSH_MSG_USERAUTH_FAILURE (51)`**
name-list	authentications that can continue
boolean	partial success

where the *name-list* is a list of methods that may productively continue the dialog. If the server accepts authentication, it sends a single-byte message, **`SSH_MSG_USERAUTH_SUCCESS (52)`**.

The message exchange involves the following steps:

1. The client sends a `SSH_MSG_USERAUTH_REQUEST` with a requested method of none.
2. The server checks to determine if the username is valid. If not, the server returns `SSH_MSG_USERAUTH_FAILURE` with the partial success value of false. If the username is valid, the server proceeds to step 3.
3. The server returns `SSH_MSG_USERAUTH_FAILURE` with a list of one or more authentication methods to be used.
4. The client selects one of the acceptable authentication methods and sends a `SSH_MSG_USERAUTH_REQUEST` with that method name and the required method-specific fields. At this point, there may be a sequence of exchanges to perform the method.
5. If the authentication succeeds and more authentication methods are required, the server proceeds to step 3, using a partial success value of true. If the authentication fails, the server proceeds to step 3, using a partial success value of false.
6. When all required authentication methods succeed, the server sends a `SSH_MSG_USERAUTH_SUCCESS` message, and the Authentication Protocol is over.

The server may require one or more of the following authentication methods:

- *publickey:* The details of this method depend on the public-key algorithm chosen. In essence, the client sends a message to the server that contains the client's public key, with the message signed by the client's private key. When the server receives this message, it checks to see whether the supplied key is acceptable for authentication and, if so, it checks to see whether the signature is correct.
- *password:* The client sends a message containing a plaintext password, which is protected by encryption by the Transport Layer Protocol.
- *hostbased:* Authentication is performed on the client's host rather than the client itself. Thus, a host that supports multiple clients would provide authentication for all its clients. This method works by having the client send a signature created with the private key of the client host. Thus, rather than directly verifying the user's identity, the SSH server verifies the identity of the client host—and then believes the host when it says the user has already authenticated on the client side.

Connection Protocol

The SSH Connection Protocol runs on top of the SSH Transport Layer Protocol and assumes that a secure authentication connection is in use. That secure authentication connection, referred to as a *tunnel,* is used by the Connection Protocol to multiplex a number of logical channels.

RFC 4254, "The Secure Shell (SSH) Connection Protocol," states that the Connection Protocol runs on top of the Transport Layer Protocol and the User Authentication Protocol. RFC 4251, "SSH Protocol Architecture," states that the Connection Protocol runs over the User Authentication Protocol. In fact, the Connection Protocol runs over the Transport Layer Protocol, but assumes that the User Authentication Protocol has been previously invoked.

All types of communication using SSH, such as a terminal session, are supported using separate channels. Either side may open a channel. For each channel, each side associates a unique channel number, which need not be the same on both ends. Channels are flow-controlled using a window mechanism. No data may be sent to a channel until a message is received to indicate that window space is available.

The life of a channel progresses through three stages: opening a channel, data transfer, and closing a channel.

When either side wishes to open a new channel, it allocates a local number for the channel and then sends a message of the form:

byte	**SSH_MSG_CHANNEL_OPEN**
string	channel type
uint32	sender channel
uint32	initial window size
uint32	maximum packet size
....	channel type specific data follows

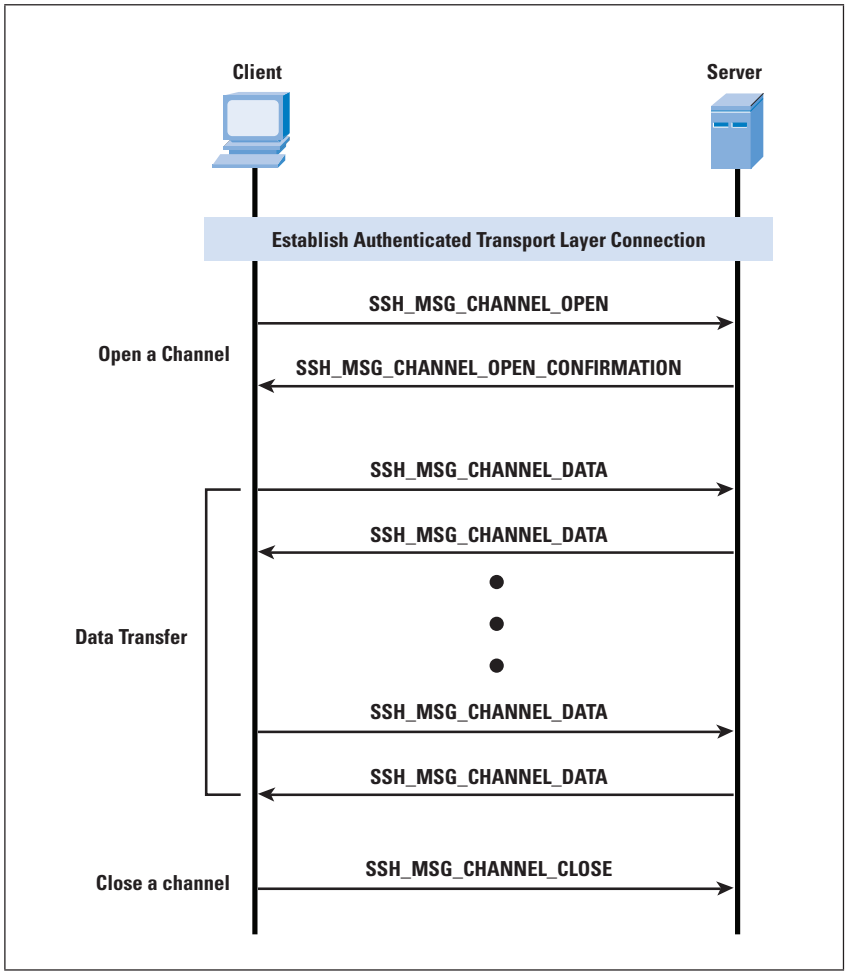
where *uint32* means unsigned 32-bit integer. The *channel type* identifies the application for this channel, as described subsequently. The *sender channel* is the local channel number. The *initial window size* specifies how many bytes of channel data can be sent to the sender of this message without adjusting the window. The *maximum packet size* specifies the maximum size of an individual data packet that can be sent to the sender. For example, one might want to use smaller packets for interactive connections to get better interactive response on slow links.

If the remote side is able to open the channel, it returns a **SSH_MSG_CHANNEL_OPEN_CONFIRMATION** message, which includes the sender channel number, the recipient channel number, and window and packet size values for incoming traffic. Otherwise, the remote side returns a **SSH_MSG_CHANNEL_OPEN_FAILURE** message with a reason code indicating the reason for failure.

After a channel is open, *data transfer* is performed using a **SSH_MSG_CHANNEL_DATA** message, which includes the recipient channel number and a block of data. These messages, in both directions, may continue as long as the channel is open.

When either side wishes to close a channel, is sends a **`SSH_MSG_CHANNEL_CLOSE`** message, which includes the recipient channel number. Figure 4 provides an example of Connection Protocol Exchange.

Figure 4: Example SSH Connectioin Protocol Message Exchange



Four channel types are recognized in the SSH Connection Protocol specification:

- *session:* Session refers to the remote execution of a program. The program may be a shell, an application such as file transfer or e-mail, a system command, or some built-in subsystem. When a session channel is opened, subsequent requests are used to start the remote program.
- *x11:* This channel type refers to the X Window System, a computer software system and network protocol that provides a GUI for networked computers. X allows applications to run on a network server but be displayed on a desktop machine.
- *forwarded-tcpip:* This channel type is remote port forwarding, as explained subsequently.
- *direct-tcpip:* This channel type is local port forwarding, as explained subsequently.

One of the most useful features of SSH is *port forwarding*. Port forwarding provides the ability to convert any insecure TCP connection into a secure SSH connection. It is also referred to as *SSH tunneling*. We need to know what a port is in this context. A *port* is an identifier of a user of TCP. So, any application that runs on top of TCP has a port number. Incoming TCP traffic is delivered to the appropriate application on the basis of the port number. An application may employ multiple port numbers. For example, for the *Simple Mail Transfer Protocol* (SMTP), the server side generally listens on port 25, so that an incoming SMTP request uses TCP and addresses the data to destination port 25. TCP recognizes that this address is the SMTP server address and routes the data to the SMTP server application.

Figure 5: SSH Transport Layer Packet Exchanges

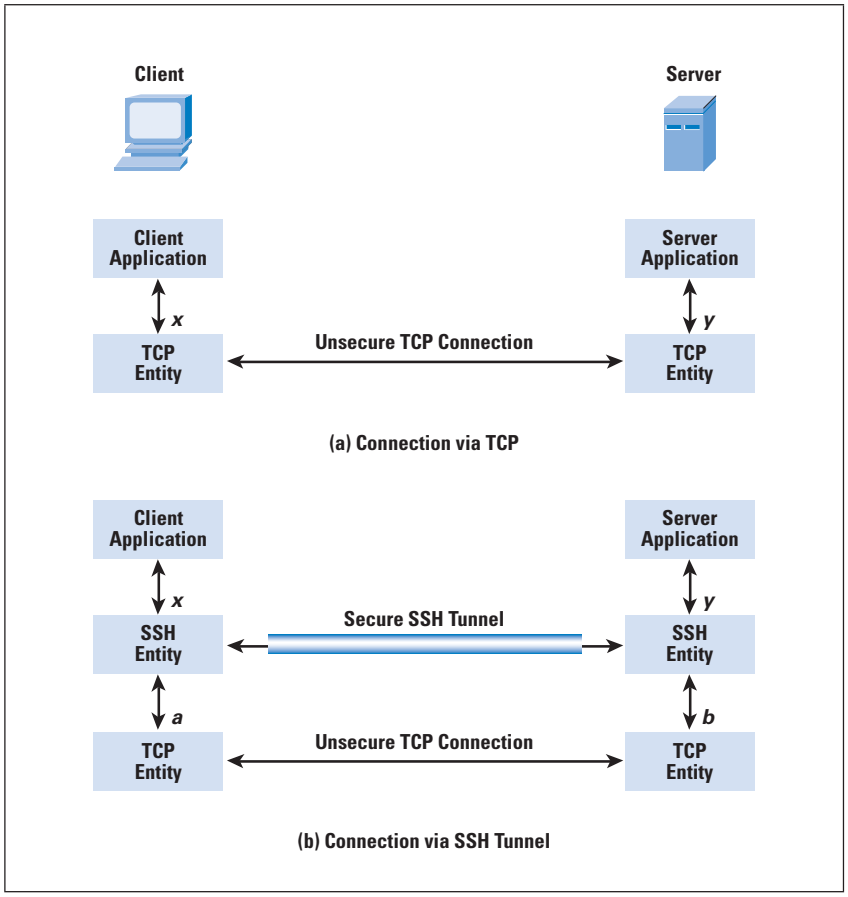


Figure 5 illustrates the basic concept behind port forwarding. We have a client application that is identified by port number *x* and a server application identified by port number *y*. At some point, the client application invokes the local TCP entity and requests a connection to the remote server on port *y*. The local TCP entity negotiates a TCP connection with the remote TCP entity, such that the connection links local port *x* to remote port *y*.

To secure this connection, SSH is configured so that the SSH Transport Layer Protocol establishes a TCP connection between the SSH client and server entities with TCP port numbers *a* and *b*, respectively. A secure SSH tunnel is established over this TCP connection. Traffic from the client at port *x* is redirected to the local SSH entity and travels through the tunnel where the remote SSH entity delivers the data to the server application on port *y*. Traffic in the other direction is similarly redirected.

SSH supports two types of port forwarding: local forwarding and remote forwarding. *Local forwarding* allows the client to set up a "hijacker" process. This process will intercept selected application-level traffic and redirect it from an unsecured TCP connection to a secure SSH tunnel. SSH is configured to listen on selected ports. SSH grabs all traffic using a selected port and sends it through an SSH tunnel. On the other end, the SSH server sends the incoming traffic to the destination port dictated by the client application.

The following example should help clarify local forwarding. Suppose you have an e-mail client on your desktop and use it to get e-mail from your mail server through the *Post Office Protocol* (POP). The assigned port number for POP3 is port 110. We can secure this traffic in the following way:

1. The SSH client sets up a connection to the remote server.
2. Select an unused local port number, say 9999, and configure SSH to accept traffic from this port destined for port 110 on the server.
3. The SSH client informs the SSH server to create a connection to the destination, in this case mailserver port 110.
4. The client takes any bits sent to local port 9999 and sends them to the server inside the encrypted SSH session. The SSH server decrypts the incoming bits and sends the plaintext to port 110.
5. In the other direction, the SSH server takes any bits received on port 110 and sends them inside the SSH session back to the client, which decrypts and sends them to the process connected to port 9999.

With *remote forwarding,* the user's SSH client acts on the server's behalf. The client receives traffic with a given destination port number, places the traffic on the correct port, and sends it to the destination the user chooses.

A typical example of remote forwarding follows: You wish to access a server at work from your home computer. Because the work server is behind a firewall, it will not accept an SSH request from your home computer. However, from work you can set up an SSH tunnel using remote forwarding.

This process involves the following steps:

1. From the work computer, set up an SSH connection to your home computer. The firewall will allow this, because it is a protected outgoing connection.
2. Configure the SSH server to listen on a local port, say 22, and to deliver data across the SSH connection addressed to remote port, say 2222.
3. You can now go to your home computer and configure SSH to accept traffic on port 2222.
4. You now have an SSH tunnel that you can use for remote logon to the work server.

Summary

SSH is one of the most commonly used cryptographic applications. It provides great flexibility and versatility for a wide variety of tasks, including remote administration, file transfer, web development, and penetration testing.

References

[1] Cusack, F. and Forssen, M. "Generic Message Exchange Authentication for the Secure Shell Protocol (SSH)," RFC 4256, January 2006.

[2] Lehtinen, S. and Lonvick, C., "The Secure Shell (SSH) Protocol Assigned Numbers," RFC 4250, January 2006.

[3] Schlyter, J. and Griffin, W. "Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints," RFC 4255, January 2006.

[4] Ylonen, T., "SSH – Secure Login Connections over the Internet," Proceedings, Sixth USENIX UNIX Security Symposium, July 1996.

[5] Ylonen, T. and Lonvick, C., "The Secure Shell (SSH) Protocol Architecture," RFC 4251, January 2006.

[6] Ylonen, T. and Lonvick, C., "The Secure Shell (SSH) Authentication Protocol," RFC 4252, January 2006.

[7] Ylonen, T. and Lonvick, C., "The Secure Shell (SSH) Transport Layer Protocol," RFC 4253, January 2006.

[8] Ylonen, T. and Lonvick, C., "The Secure Shell (SSH) Connection Protocol," RFC 4254, January 2006.

WILLIAM STALLINGS is a consultant, lecturer, and author of more than a dozen books on data communications and computer networking. His latest book is *Cryptography and Network Security* (Prentice Hall, 2010). He maintains a computer science resource site for computer science students and professionals at **WilliamStallings.com/StudentSupport.html** and is on the editorial board of Cryptologia. He has a Ph.D. in computer science from M.I.T. He can be reached at **ws@shore.net**

Book Review

Protocol Politics

Protocol Politics: The Globalization of Internet Governance, by Laura DeNardis, MIT Press, 2009, ISBN 978-0-26204257-4.

In *Protocol Politics*, Dr. Laura DeNardis assembles a variety of stories gleaned from official and unofficial *Internet Engineering Task Force* (IETF) records and firsthand accounts, and supplements them with primer-level descriptions of successive generations of Internet addressing and routing protocols to create a broadly accessible overview of the factors that have shaped the present and evolving state of these most central features of Internet technology.

The author, a former enterprise networking consultant and technology analyst, joined the Yale Law School Information Society Project as a Post-Doctoral Fellow in 2006, and became the Executive Director of the program in late 2008. DeNardis approaches the challenge of organizing these disparate materials by adopting an interpretive framework that highlights the role of power—interpersonal as opposed to electrical—as both the primary input and most important output or consequence of the definition, selection, and implementation of Internet protocols.

The book knits together a wealth of important historical information that has to-date remained largely neglected outside of the technical community. Although DeNardis' choice of framing is perfectly legitimate—and in fact quite common within the academic disciplines that delve into the influence of institutions on industries, economies, and society—in this case it leads her to overreach a bit, and arguably to draw a few prominent conclusions that are not well-supported by the balance of available historical evidence.

Organization of the Book

DeNardis employs this interpretive framework across six densely written chapters, the first four of which directly address the significance of power in a different functional context of relevance to the evolution of Internet addressing and routing. The introductory chapter investigates the significance of scarcity and its effect on protocol resource management and *Internet Governance*. Here she devotes considerable space to detailing the critical importance of IP addresses as the single element among Internet protocols that is both indispensable and nonsubstitutable. DeNardis' insightful overview of the general characteristics of IP addresses is somewhat marred by her mixing together of some basic, intrinsic functional properties of addressing (for example, *identifier* and *locator* functions) with various necessary but extrinsic correlates or consequences of those functional properties (for example, universality, external observability), or with contingent features of current IP address usage conventions (for example, indifference to underlying technologies).

In addition, despite the ostensible focus on scarcity in the chapter, no reference is made to that other, equally essential and quantity-constrained feature of the Internet service landscape—that is, the inherently limited, occasionally overtaxed carrying capacity of Internet routing subsystems, particularly the collectively provisioned inter-domain routing system. Overall, *Protocol Politics* provides almost no exposure to the technical, operational, and economic constraints that define the routing environment, much less to the constraints that those factors impose on number resource distribution arrangements. Chapter One closes with an overview of the priorities that justify and define the sphere of Internet Governance which anticipates many of the concluding observations in the book's final chapter on "Opening Internet Governance." Both chapters acknowledge "technical expertise" only as a source of institutional or political legitimacy, without according any special significance to the *content* of such expertise, or why it matters at all. Readers of *Protocol Politics* may thus come away with insufficient appreciation of the fact that before Code can become Law (or anything else), it first must be running code—*and that not every wish is translatable into running code*.[1]

Piercing the Fog of Protocol War

In the three chapters that follow, DeNardis presents her observations about how power shapes and flows from the definition and selection of Internet protocols. Chapter Two covers the first half of this proposition, focusing on the events that followed the December 1990 IETF meeting where, DeNardis suggests, the twin challenges that would shape the development of Internet addressing intersected with the chief institutional impediment that would ultimately reveal the true political nature of Internet standards development.

The first challenge that she identifies is the foreseeable inadequacy of IPv4 as the exclusive addressing resource pool for a rapidly growing and globalizing Internet. In keeping with the overall theme of the book, the second challenge that DeNardis chooses to highlight is the implicitly political challenge of accommodating greater international participation in the U.S.-centric Internet technical coordination and decision-making bodies. Against this backdrop, DeNardis introduces the other chief protagonist in her story, the *International Organization for Standardization* (ISO), which backed the rival *Open Systems Interconnection* (OSI) family of protocols as an alternative, non-TCP/IP-based foundation for the ongoing, global proliferation of data networking. DeNardis details the convoluted, multidimensional deliberations that followed that 1990 IETF meeting, which eventually culminated in 1994 in the formal recognition of IPv6 as "The Next-Generation Internet Protocol."

Chapter Three goes on to explore the implications of both IPv4 and IPv6 for important civil liberties—especially privacy—and how such considerations did and did not, *but hypothetically might have,* influenced the choice and form of the most important features of TCP/IP.

Chapter Four rounds out the central thesis of the book by illustrating how various national-level considerations—especially government-directed foreign and domestic economic policies—have resulted in an increasingly diverse global pattern of IPv6 adoption.

DeNardis' detailed account of the complexities surrounding the *IP Next-Generation* (IPng) debate and its aftermath incorporates a diverse mix of sources, from pointed remarks made on various mailing lists, to conference presentations and official *Internet Architecture Board* (IAB) meeting minutes, and represents a major feat of historical scholarship. That said, her presentation of "relevant historical facts" from the 1990–1994 period is by no means complete, nor is her interpretation of the facts that she does cover or the conclusions that she draws from them immune to criticism. For example, in puzzling over possible hidden forces behind the selection of IPv6, DeNardis states that:

> "If anything, there was market pressure to adopt an OSI rather than TCP/IP-based protocol. The ISO alternative had the political backing of most Western European governments (sic) influential technology companies, and users invested in OSI protocols, and was even congruent with OSI directives of the United States. The selection of IPv6…" (p. 61)

Although these facts may be beyond dispute, they do not represent the full picture. To give one illustration, in 1989, almost 2 years before the date that DeNardis marks as the start of the IETF's lone struggle against the combined forces of Europe, influential carriers and hardware manufacturers, and the U.S. government, an indigenous movement of European network operators emerged and began self-organizing to facilitate the exchange of TCP/IP-based traffic, contact information, and operational tips, and to discuss best practices in areas of networking where individual network-level decisions could have far-reaching effects on internetwork performance.

That organization would go on to become *Réseaux IP Européens Network Coordination Centre* (RIPE NCC), the first independent, transnational registry for Internet Protocol number resources, and the institution that would provide the organizational template for the *Regional Internet Registries* (RIRs) that subsequently sprang up in Asia (APNIC, 1993), North America (ARIN, 1997), Latin America (LACNIC, 2002), and Africa (AFRINIC, 2004). These facts point to a level of active indigenous European support for TCP/IP-based networking that would seem to be at odds with any suggestion of a continent united in support of OSI against a less-attractive standard being pushed by an insular foreign organization.

Thus, regardless of whether DeNardis' concerns about institutions and power relations are well-founded, her intuitions about the division of contestants in the great protocol power struggle clearly are not.[2]

Market Contrast

Another question that DeNardis raises, obliquely but repeatedly, relates to the possibility of "free markets" as an alternative mechanism for defining, selecting, and distributing Internet protocols and the virtual resources that they create.

In no less than a dozen separate passages scattered across each of the chapters in the book, DeNardis sharply contrasts a range of IETF and RIR institutional processes to the workings of the "free market." For example, she observes that the value of IP addresses is unknown because they have never been exchanged in free markets (p. 16); that Internet addresses have never been exchanged in free markets (pp. 23, 190); that the privacy potential of Internet technologies is enhanced by selection pressures from free markets (p. 74); that the IETF refused to countenance an IPng protocol selection made by free markets (p. 51); that the selection of IPv6 happened outside the realm of free markets (p. 69); that widespread adoption of IPv6 is impeded by the absence of a free market for protocols (p. 137); that IETF philosophy holds that it would be inappropriate to exchange protocol resources in free markets (pp. 163, 183–184); that the *Internet Assigned Numbers Authority* (IANA) refused to relinquish IP addresses to free markets (pp. 163, 164); that traditional opposition to the exchange of protocol resources in free markets fortified and centralized the IETF's institutional control (p. 184); and that exchanging IPv4 in free markets has pragmatic appeal, if only as a temporary stopgap (p. 228), although such exchanges might have unintended consequences (p. 229).

Given this frequency of repetition, it is impossible to avoid forming a strong impression of DeNardis' underlying opinion about the intrinsic merits of "free markets" as compared to the seemingly market-antithetical goals and practices of the IETF and the other TCP/IP-centric standards-setting and technical coordination bodies. However, even if one stipulates that "free markets" would by definition represent a superior alternative to the enumerated protocol design and distribution mechanisms, DeNardis never provides any clear indication of where a model for such "free markets" might be found—whether in Europe, the United States, or anywhere else, now or anytime in the past.

Even her own description of that fateful moment in networking history when IPv6 was selected clearly suggests that the alternative to the IETF process that ultimately prevailed was itself neither "free" nor especially market-like:

> "… congruent with OSI directives of the United States. The selection of IPv6, an expansion of the prevailing IPv4 protocol *over such a politically sanctioned OSI alternative* solidified and extended the position of the Internet's traditional standards-setting establishment as the entity responsible for the Internet's architectural direction." (p. 61, emphasis added).

Arguably, the non-inclusion of a pure "free market" example is not merely a coincidence, but rather reflects a more fundamental problem inherent in the concept itself. Further, if one grants that the market mechanism that is *most free* is the one that fosters the broadest participation in those activities that make markets attractive—including openness to participation, exercise of individual choice, competition, accelerated innovation, and wealth creation—then one might interpret the two-plus orders-of-magnitude growth in the number of independent network services providers operating on both sides of the Atlantic since that time as a solid indicator that markets have not suffered too badly from the 1994 decision to extend the lifetime of TCP/IP through IPv6.

Clearly the looming inflection point in IP addressing will provide many irresistible opportunities to revisit that choice in the days ahead. Meanwhile, the question of whether the embrace of an OSI-friendlier IPng by the IETF would have been sufficient to offset the varied negative externalities that might have accompanied such a choice must forever remain unanswered. Would an IETF endorsement have trumped the as-yet incomplete state of OSI standards, as well as OSI's tighter associations with non-standards-based operating systems, proprietary hardware platforms, and the connection-oriented networking technologies favored by then Internet-averse incumbent *Public Switched Telephone Network* (PSTN) operators? Would that choice alone have created or been likely to foster a freer market, or to have led to a more enthusiastic, widespread embrace of a different post-IPv4 addressing format—or alternately would it have led to the appearance of books like *Protocol Politics,* albeit written from the opposite perspective, and possibly a decade sooner? Contrary to the popular adage, hindsight is not 20/20, any more than is our vision of where to go from here.[3]

Beyond the Clash of Idealizations

Writing a book review is an inherently risky undertaking, one that is vulnerable to many of the same human biases and errors that have unquestionably informed both the selection and development of various technical standards, just as they have influenced the embrace, rejection, or modification of various market arrangements throughout history.

Even when people (book reviewers, for example) recognize that real-world decisions and their consequences tend to be irreducibly complex—or perhaps precisely because they recognize that complexity—they nevertheless tend to gravitate toward explanatory frameworks and cognitive models that promise to invest their perceptions and choices with the kind of absolute certitude that is very rarely found outside of the physical world (and only infrequently found there).

The problem, of course, is that many such explanatory frameworks can be found to fit quite nicely with the same set of human experiences, even though some of those models may be mutually orthogonal, and some may be quite mutually and actively antagonistic. In this sense, the juxtaposition of pure, frictionless "free markets" alongside the idea of absolutely pure scientific or technical decision making divorced from all other human considerations, while well-calibrated to inflame passions, represents less a contrast of opposites than a rather less illuminating pairing of two deeply unrealistic ideal types. Distilling a book as rich and informative as *Protocol Politics* down to one possible review-sized essence is much easier to accomplish from just such a privileged vantage point, and no doubt this particular review suffers from the all-too-predictable effects described herein. However, with that caveat firmly established, a few more things about *Protocol Politics* deserve to be mentioned here.

First, *Protocol Politics* is an important book. It is the well-written and informative, and is the first to be written for a general audience that draws on the right historical sources (or at least most of the right ones that remain accessible) to cover this critical period in the development of the Internet's core addressing and routing protocols. Even those who are least likely to be sympathetic to its findings are likely to find *Protocol Politics* to be a thoughtful and engaging read.

Second, IPJ readers and other technologists should not dismiss the inherently political, power-oriented framework that DeNardis employs in *Protocol Politics*. In general, the most honest and effective response to an assertion of *systemic* political or institutional bias is not to claim an equally absolute, otherworldy detachment from the affairs of man, but rather to remind the critic that in a world where all institutions are regarded as manifestations of somebody's will to power, specific targeted criticisms based *solely* on that fact lose all coherence. Would-be institutional critics who espouse such views thus have no choice but to make a positive argument as to which arrangement, among all of the equally power-tainted institutional arrangements that are possible, should be regarded as the preferable outcome, for whom, and why. Judged in this light, this reviewer feels that "the IETF way" still stands up pretty well, foibles and all. There is always room for improvement, but just as in matters of code, a concrete proposal for improvement is worth a thousand critiques of the past.

Finally, the careful reader may notice a pattern within this review, one composed of points highlighted here even though they may not be equally central to the story presented in *Protocol Politics* (for example, about the role of technical expertise in Internet governance, the dynamic limitations of routing system carrying capacity, the possibility of free market alternatives to current Internet address distribution arrangements, and so on).

Each of these points merits special attention because taken together they help to illuminate the existence of an identical set of critiques that have reappeared periodically in the course of another, much older (actually, centuries-old) debate that parallels the as-yet unresolved debates outlined by DeNardis in *Protocol Politics*.

In both instances, the question at issue involves the relative merits of nonmarket, technical expert-based systems as a means of managing resources that are uniquely central to economic growth, and for mitigating the systemic risks that can threaten that growth. In that other debate, arguments in favor of pure free market solutions have generally been dismissed as extreme and unrealistic for more than a century, ever since the last real-world implementation of such a system finally succumbed to its own chronic instabilities and was replaced by a nonmarket coordination arrangement. More recently, however, a resurgence of extreme turmoil in that parallel industry has undermined belief in expert management, if not in the underlying "hard realities" that were supposed to constitute the managers' technical domain of expertise. In turn this turmoil has sparked renewed interest in the long-marginalized pure free market proposals, as well as in alternative remedies involving much tighter industry control by nonmarket authorities.

How the current chapter in either of these parallel stories will play out remains to be written. However, those who are eager to anticipate the kind of language that is likely to play a central role in both outcomes will find that a close reading of *Protocol Politics* provides a wealth of possibilities to consider, and more than a few to keep one up at night.

—Tom Vest, Consultant
tvest@eyeconomics.com

References

[1] DeNardis makes several references to the idea that *Code is Law,* which was first articulated by Larry Lessig in *Code and Other Laws of Cyberspace* (1999) [Editor's note: *Code* was reviewed in IPJ Volume 11, No. 3]. Here the phrase is juxtaposed with David Clark's famous paean to "rough consensus and running code," which DeNardis describes as an "articulation of the IETF's core philosophy" (p. 47), and amended with a paraphrasing of an early (c. 1992) observation made by Marshall Rose about a common problem encountered when attempting to implement code to satisfy a non-operationally developed standard. The original staying was, "The problems of the real world are remarkably resilient to administrative fiat."

[2] Several formerly obscure insights on the events of this period were recently illuminated by RIPE co-founders Rob Blokzijl and Daniel Karrenberg, during RIPE's 20th Anniversary Commemoration at the RIPE 58 meeting in Amsterdam (May 2009). Some of these are available at:
http://www.ripe.net/ripe/meetings/ripe-58/content/presentations/Blokzijl-RIPE-20-years.pdf

and

http://www.ripe.net/ripe/meetings/ripe-58/content/presentations/the-origins-of-ripe.pdf

[3] Those wishing to investigate these questions further may benefit substantially from yet another unique historical resource that has recently been made available online. Thanks to the Charles Babbage Institute and the Institute of Technology at the University of Minnesota, the entire ten-year archive of *ConneXions—The Interoperability Report* (1987–1996) is now available online at: **http://www.cbi.umn.edu/hostedpublications/Connexions/index.html**

In keeping with its mandate to track the interoperability of emerging network technologies, *ConneXions* published more than sixty substantive articles on OSI and GOSIP during the period leading up to and following the IPng debates recounted in *Protocol Politics*.

Fragments

Lorenzo Colitti (L) and Erik Kline
Photo: Matsuzaki Yoshinobu

Colitti and Kline Receive First Itojun Service Award

The first *Itojun Service Award* was presented at the recent IETF meeting in Hiroshima, Japan to Lorenzo Colitti and Erik Kline of Google for their outstanding contributions to the development and deployment of IPv6.

The award honours the memory of Dr. Jun-ichiro "Itojun" Hagino, who passed away in 2007, aged just 37. Established by the friends of Itojun and administered by the *Internet Society* (ISOC), the award recognises and commemorates the extraordinary dedication exercised by itojun over the course of IPv6 development.

"The sustained efforts of Lorenzo and Erik have tangibly increased the availability of Web-based services that use IPv6, reflecting the Itojun Service Award's focus on pragmatic contributions in the spirit of serving the global Internet's continued evolution," said Jun Murai of the Itojun Service Award committee and Director of the WIDE Project. "The award aims to recognize how important both the development of IPv6 and related protocols and efforts to advance their deployment are to ensuring the Internet continues to serve as a platform for innovation around the world."

The award, expected to be presented annually, includes a presentation crystal, a US$3,000 honorarium and a travel grant.

Lorenzo Colitti, Network Engineer at Google said, "This is a great honour. Itojun is a legend in the IPv6 community, and the Internet is indebted to him. Without his foundational work, none of what we achieved with IPv6 would be possible—we stand on the shoulders of giants. Itojun has been a source of inspiration, and I regret never being able to meet him, to show him our work, and show him that we too shared his vision of bringing IPv6 to the users of the Internet."

Erik Kline, IPv6 Software Engineer at Google said, "It's humbling to be sharing the Itojun Service Award, having achieved by comparison only a small fraction of the impact of his widely influential body of work. For me personally, Google's IPv6 efforts are not just for the Internet and its future, but also a way to honor his vision, dedication, and passion."

More information on the Itojun Service Award is available at: `http://www.isoc.org/itojun`

ISOC Donation to Support Evolution of W3C Organization

ISOC and the *World Wide Web Consortium* (W3C) recently announced a donation from ISOC for the purpose of advancing the evolution of W3C as an organization that creates open Web standards. Citing strongly aligned views on the value of an open global Internet and support for the current Internet governance and management model, ISOC pledged to support W3C efforts to implement a more agile, inclusive, and flexible organizational structure.

"ISOC and W3C have worked together for years in a number of areas, and have deeply shared values about the Internet's development," said Lynn St. Amour, President and CEO of ISOC. "Our support to the W3C in their transition efforts demonstrates our commitment to ensuring the Internet continues to be a global platform for innovation. What's at stake is the Internet's openness, which is a critical enabler of new products and services to billions of users worldwide."

"ISOC and W3C have a long history of cooperation and the Internet ecosystem has benefited from our shared yet independent voices," said Tim Berners-Lee, W3C Director. "The W3C staff, Members, and community continue to work on making W3C more relevant and valuable to the Web and Internet communities. ISOC support will allow W3C to evolve its structure to ensure we continue to forge solid working relationships with the increasing numbers of developers and users, worldwide."

The two organizations will continue to operate independently, and will maintain their long-standing, informal collaboration. ISOC's pledge of support is for three years, with both organizations working to ensure progress. A FAQ with additional information is available on both the ISOC site and the W3C site, see `http://www.isoc.org` and `http://www.w3.org`

DNSSEC Deployment in the Root Zone

In December 2009, ICANN and VeriSign began to deploy DNSSEC across the root server system and launched a website that provides information about DNSSEC for the root zone. The website is a repository for the documentation relating to the deployment of DNSSEC, and it includes information such as technical status updates and the full timetable for the deployment osf DNSSEC.
See: `http://www.root-dnssec.org/`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

March 2010 Volume 13, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

Previous articles in IPJ have described *Domain Name System Security Extensions* (DNSSEC), the security system for the *Domain Name System* (DNS). DNSSEC introduces security into the DNS through the use of cryptographic keys and digital signatures. Interest in DNSSEC has grown in recent months, as the *Internet Corporation for Assigned Names and Numbers* (ICANN) and VeriSign have undertaken a phased program to deploy DNSSEC across the root server system in the first half of 2010. In an article by four DNS practitioners, we will explore some side effects of DNSSEC, and examine what happens in two widely used DNS resolver implementations when DNS clients lag behind in synchronizing their local copy of trust keys with the master keys used by the zone administrators to sign their DNS data.

Several articles in IPJ have dealt with various concerns related to scaling of the Internet. In this issue, Paul Francis and Xiaohu Xu describe *Virtual Aggregation,* a new routing technology being developed by the GROW working group of the IETF to reduce the size of the *Forwarding Information Base* (FIB) held in memory by routers.

The *Request For Comments* (RFC) Series has been the main publication channel for Internet standards and related documents for more than 40 years. The RFC Editor function is in the process of being restructured and moved from its original home at the *University of Southern California Information Sciences Institute* (USC/ISI). Leslie Daigle describes the history and future of the RFC Editor mechanism.

If you are reading this online and did not receive the March 2010 edition of IPJ, it may be because your subscription has expired. You can still renew your subscription by visiting the "Subscriber Services" section of our webpage at **www.cisco.com/ipj**. Enter your subscription ID and e-mail address to gain access to your subscription record. If you don't know your subscription ID or have changed e-mail address recently, just send a message to **ipj@cisco.com** and we will take care of the renewal and update for you.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Rolling Over DNSSEC Keys

by George Michaelson, APNIC, Patrick Wallström, .SE, Roy Arends, Nominet, Geoff Huston, APNIC

As we are constantly reminded, the Internet can be a very hostile place, and public services are placed under constant pressure from a stream of probe traffic, attempting to exploit any one of numerous vulnerabilities that may be present at the server. In addition, there is the threat of *Denial of Service* (DoS)[1] attacks, where a service is subjected to an abnormally high traffic load that attempts to saturate and take it down. This story starts with the detection of a possible hostile DoS attack on *Domain Name System* (DNS) servers, and narrates the investigation as to the cause of the incident, and the wider implications of what was found in this investigation.

Detecting the Problem

The traffic signature in Figure 1 is a typical signature of an attempted DoS attack on a server, where the server is subjected to a sudden surge in queries. In this case the traffic log is from a secondary DNS Name Server that is authoritative for a number of subdomains of the **`in-addr.arpa`** zone; the traffic surge shown here commenced on December 16, 2009. The traffic pattern shifted from a steady state of some 12 Mbps to a new steady state of more than 20 Mbps, peaking at 30 Mbps.

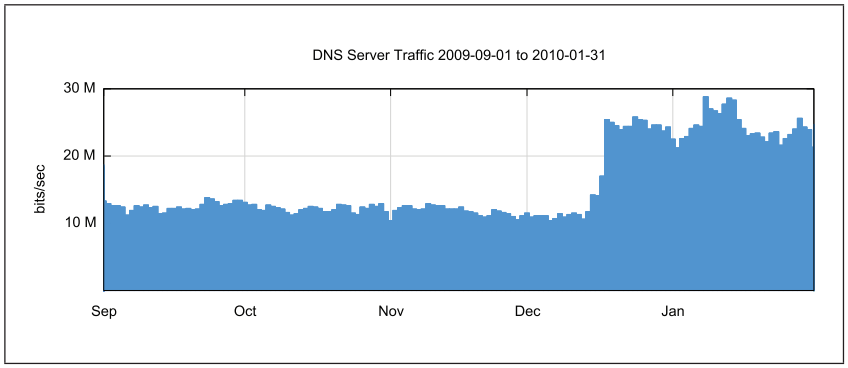


Figure 1: Traffic Load for ***`in-addr.arpa`*** *Server (provided by George Michaelson)*

Because the traffic shown in Figure 1 is traffic passed to and from a Name Server, the next step is to examine the DNS traffic on the Name Server, and in particular look at the rate of DNS queries that are being sent to the Name Server (Figure 2). The bulk of the additional query load is for DNSKEY *Resource Records* (RRs), which are queried as part of the operation of *Domain Name System Security Extensions* (DNSSEC)[2].

Because this zone is a DNSSEC signed zone, DNSKEY queries will cause the server to respond with a DNSKEY RR and the related RRSIG RR in response to each query. This pair of RRs generates a response that is 1,188 bytes in this case. At a peak query rate of some 3,000 DNS queries per second, a traffic response from the server in excess of 35 Mbps will be generated.

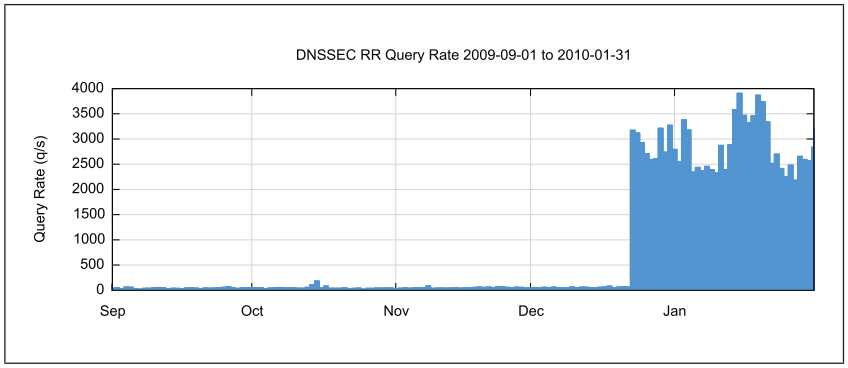


Figure 2: Query Rate for ***in-addr.arpa*** *Server (provided by George Michaelson)*

There are many possibilities as to what is going on here:

- This problem could be caused by a DoS attack directed at the server, with the attacker attempting to saturate the server by flooding it with short queries that generate a large response.
- This problem could be caused by a DNS reflection DoS attack, where the attacker is placing the address of the intended victim or victims in the source address of the DNS queries and attempting to overwhelm the victim with this DNS response traffic.

Although it is good to be suspicious, it is also useful to remember the old adage that we should be careful not to ascribe to malice what could equally be explained by incompetence, so numerous other explanations should also be considered, including:

- This problem could be a DNS resolver problem, where the resolver is not correctly caching the response, and some local event is triggering repeated queries.
- This problem could be a bug in an application where the application has managed to wedge itself in a state of rapid-fire queries for DNSKEY RRs.

The next step is to examine some of these queries more closely, and, in particular, look at the distribution of query source addresses to see if this load can be attributed to a small number of resolvers that are making a large number of queries, or if the load is spread across a much larger set of resolvers. The server in question typically sees on the order of 500,000 to 1,000,000 distinct query sources per day.

Closer inspection of the query logs indicates that the additional load is coming from a relatively small subset of resolvers, on the order of 1,000 distinct source addresses, with around 100 "heavy hitters." In other words, all this DNS traffic is being generated by some 0.01% of the DNS clients. The sequence of queries from one such resolver that is typical of the load being imposed on the server is shown in Figure 3.

Figure 3: DNS Query Sequence Packet Capture

1. Client requests the Delegation Signer (DS) records for the `211.89.in-addr.arpa` zone.
2. Reply says "no such delegation," and sends *DNSSEC Signature* (RRSIG) and *Next-Secure record* (NSEC) from the parent zone, and surrounding records.
3. Client requests DNSKEY from the parent zone.
4. Server sends DNSKEY and RRSIG set for the parent zone.

Having established an initial query state and the DNSKEY and signature set over the original request, the client then paradoxically repeatedly re-queries the parent-zone DNSKEY state. This process is elided as follows because the query and response do not differ during this exchange:

5. Client repeats the DNSKEY request.
6. Server repeats the DNSKEY and RRSIG response.
7. Client repeats the DNSKEY request.
8. Server repeats the DNSKEY and RRSIG response.

This exchange of DNSKEY request and DNSKEY and RRSIG response is repeated a further 6 times.

If this additional query load had appeared at the server over an extended period of time, it would be possible to ascribe this problem to a faulty implementation of a DNS resolver, or a faulty client application. However, the sudden onset of the additional load tends to suggest that something else is happening. The most likely explanation is that some external "trigger" event exacerbated a latent behavioral bug in a set of DNS resolver clients. And the most likely external trigger event is a change of the contents of the zones being served.

So we can now refine our set of possible causes to concentrate consideration on the possibility that:

- Something changed in the zones being served by this secondary server that triggered a pathological query response from a set of resolvers.

And indeed the contents of the zones did change on the day when the traffic profile changed, with a key change being implemented on that day.

DNSSEC Key Management

It is considered good operational practice to treat cryptographic keys with a healthy level of respect. As RFC 4641[3] states: "The longer a key is in use, the greater the probability that it will have been compromised through carelessness, accident, espionage, or cryptanalysis." Even though the risk is considered slight if you have chosen to use a decent key length, RFC 4641 recommends, as good operational practice, that you "roll" your key at regular intervals. Evidently it is a popular view that fresh keys are better keys.

The standard practice for a "staged" key rollover is to generate a new key pair, and then have the two public keys coexist at the publication point for a period of time. This practice allows relying parties, or clients, some period of time to pick up the new public key. Where possible during this period, signing is performed twice, once with each key, so that the validation test can be performed using either key. After an appropriate interval of parallel operation, the old key pair can be deprecated and the new key can be used exclusively for signing.

This key rollover process should be a routine procedure, without any intended side effects. Resolvers that are using DNSSEC should refresh their local cache of zone keys in synchronization with a published schedule of key rollover, and ensure that they load a copy of the new key within the period when the two keys coexist. In this way when the old key is deprecated, responses from the zone servers can be locally validated using the new key.

The question here is why did this particular key rollover for the signed zone cause the traffic load at the server to spike? And why is the elevated query rate sustained for weeks after the key rollover event? The key had changed 6 months earlier and yet the query load prior to this most recent key change was extremely low.

DNSSEC DNS Resolver Behavior with Outdated Trust Keys

It is possible to formulate a theory as to what is going on from this collection of information. It could be that one or more DNS resolver clients has been using a local *Trust Anchor* that has been manually downloaded from the zone administrator prior to the most recent key rollover, but has not been updated since. When the key rollover occurred in December 2009, these clients could no longer validate the response with their locally stored Trust Anchors.

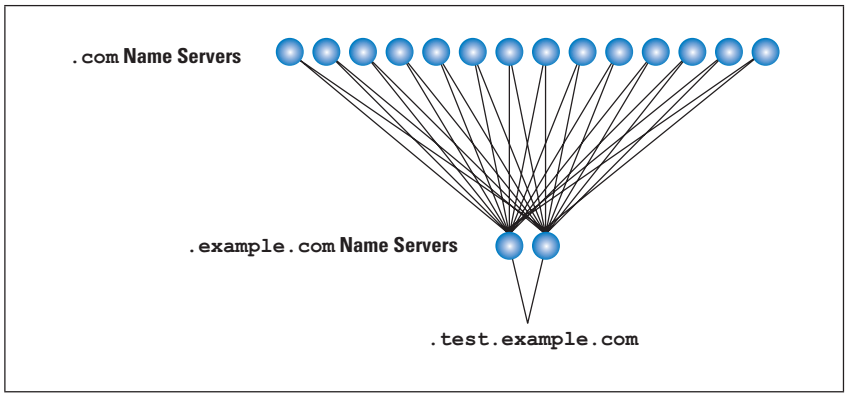
Upon detecting an invalid signature in the response, the client appears to have reacted as if there were a "man-in-the middle" injection attempt, and immediately repeated the request in an effort to circumvent the supposed attack by rapidly repeating the query. If this instance were really a man-in-the-middle injection attack, this response would be plausible, because there is the hope that the query will still reach the authoritative server and the client will receive a genuine response that can be locally validated.

Why does the client really perform this repeated query pattern? In this case the contributory factor is the use of multiple name servers in the DNS. When the DNS client performs a key validation, it performs a bottom-up search to establish the trust chain from the initial received query to a configured Trust Anchor.

Example DNSSEC Validation

As a hypothetical example, assume a TXT RRset for **test.example.com** in a signed **example.com** zone. The zone **example.com** resides on two Name Server addresses. The **example.com** zone has a *Key Signing Key* (KSK), which is referred to by the DS record in the **.com** zone. The **.com** zone is signed, and it resides on 14 addresses (11 IPv4 and 3 IPv6). The **.com** zone has a KSK, which is referred to by a Trust Anchor in the local configuration of the resolver (Figure 4).

Figure 4: Example Configuration



Assume that the locally held Trust Anchor for **.com** in the resolver has become stale. That is, the DS record for **.com** in the root zone validates, but there are no DNSKEYs in **.com** that match the DS record in the root zone.

When a client is resolving a query relating to **test.example.com**, the following search occurs:

- *Berkeley Internet Name Domain* (BIND)[9] resolves the **test.example.com** RRset. It attempts to validate it. To do so, it needs the **example.com** DNSKEY RRset.
- It resolves the DNSKEY RRset for **example.com** from a Name Server of **example.com**. It attempts to validate it. To do so, it needs the **example.com** DS RRset.
- It resolves the DS RRset for **example.com** from a Name Server of **.com**. It attempts to validate it. To do so, it needs the **.com** DNSKEY RRset.
- It resolves the DNSKEY RRset for **.com** from a Name Server of **.com**. It attempts to validate it with the locally configured Trust Anchor.

However, the resolver cannot validate the **.com** DNSKEY RRset because it does not have the proper Trust Anchor for it. It queries all remaining 13 **.com** servers for the DNSKEY RRset for **.com**. Then the resolver still does not have the proper **.com** DNSKEY, and tracks back one level:

- It resolves the DS RRset for **example.com** from the next authoritative Name Server. It attempts to validate it. To do so, it needs the **.com** DNSKEY RRset. The search goes forward again.
- It resolves the DNSKEY RRset for **.com**. It attempts to validate it with the locally configured Trust Anchor.

Because the DNSKEY RRset for **.com** has not changed, this attempt will fail as well.

The complete in-depth first search consists of:

- TXT records on 2 **example.com** servers, signed by:
- DNSKEY records on 2 **example.com** servers, referred to by:
- DS records on 14 **.com** servers, signed by:
- DNSKEY records on 14 **.com** servers.

When all possible paths are exhausted, the client will have sent the following:

- 784 (2 × 2 × 14 × 14) **.com** DNSKEY requests to 14 **.com** Name Servers
- 56 (2 × 2 × 14) **example.com** DS requests to 14 **.com** Name Servers
- 4 (2 × 2) **example.com** DNSKEY requests to 2 **example.com** Name Servers

In other words, in this example scenario with stale Trust Anchor keys in a local client's resolver, a single attempt to validate a single DNS response will cause the client to send a further 844 queries, and each **.com** Name Server to receive 56 DNSKEY RR queries and 4 DS RR queries.

The breadth and level of the search is important here, because the longer the validation chain and the more the number of authoritative Name Servers for those zones that lie on the validation chain path, the more queries that will be sent in an effort to validate a single initial response. In this example, the level of search is three deep, and terminates at **.com**. If the **.com** zone were signed by the root Name Servers and the client were using a stale root zone key, then the 20 distinct root zone server addresses (13 in IPv4 and 7 IPv6 addresses) would also be queried:

- 313,600 (2 × 2 × 14 × 14 × 20 × 20) root DNSKEY requests to 20 root Name Servers
- 15,680 (2 × 2 × 14 × 14 × 20) **.com** DS requests to 20 root Name Servers

It is worthwhile noting in this context that reverse trees and enum trees in the **.arpa** zone are longer on average. Though delegations in those subtrees might span several labels, it is not uncommon to delegate per label. Note also that the entire effort is done per incoming query—the entire search is repeated for each query.

Though this example shows an enormous query load, there are a few ceilings. In commonly used validating resolvers, such as BIND 9.7rc2, every search is performed in serial, and each search is halted after 30 seconds.

The *Unbound* client[4] also appears to have a similar request behavior, although it is not as intense because of the cache management in this implementation. Unbound will "remember" the query outcome for a further 60 seconds, so repeated queries for the same name will revert to the cache. But the DNSSEC key validation failure is per zone, and further queries for other names in the same zone will still exercise this re-query behavior. In effect, for a zone that has sufficient "traffic" of DNS load in subzones or instances inside that zone, the chain of repeated queries is constantly renewed and kept alive.

If one such client failed to update its local trusted key set, then the imposed server load on DNSSEC key rollover would be slight. However, if a larger number of clients were to be caught out in this manner, then the load signature of the server would look a lot like Figure 2. The additional load imposed on the server comes from the size of the DNSKEY and RRSIG responses, which are 1,188 bytes per response in the specific failure case that triggered this investigation.

So far we've been concentrating attention on the **`in-addr.arpa`** zone, where the operational data was originally gathered. However, it appears that this problem could happen to any DNSSEC signed domain where the zone keys are published so as to allow clients to manually load them as trust points, and where the keys are rolled on a regular basis.

It is likely that one possible cause for this situation is in the way in which some DNSSEC distributions are packaged with operating systems. For example, the *Fedora*[5] Linux distribution has bundled numerous trust keys with its packaging of a DNS resolver client and local Trust Anchor key set. When the keys associated with sub zones of **`in-addr.arpa`** rolled over in December 2009, users of this version of the Fedora Linux distribution would have been caught with stale trust keys.

So there appears to be a combination of three factors that are causing this situation:

- The use of prepackaged DNSSEC distributions that included preloaded keys in the distribution
- The use of regular key rollover procedures by the zone administrator
- Some implementations of DNS resolvers that react aggressively when there is a key validation failure by performing a rapid sequence of repeat queries, with either a very slow, or in some cases no apparent back-off in query load

This combination of circumstances makes the next scheduled key rollover for **in-addr.arpa**, scheduled for June 2010, appear to be quite an "interesting" event. If there is the same level of increase in use of DNSSEC with manually managed trust keys over this current 6-month interval as we've seen in the previous 6 months, and if the same proportion of clients fails to perform a manual update prior to the next scheduled key rollover event, then the increase in the query load imposed on **in-addr.arpa** servers at the time of key rollover promises to be truly biblical in volume.

Signing the DNS Root

There is an end in sight for this situation for the subzones of **in-addr.arpa**, and for all other such subzones that currently have to resort to various forms of distribution of their zone keys. The *Internet Corporation for Assigned Names and Numbers* (ICANN) has announced that on July 1, 2010, a signed root zone for the DNS will be fully deployed[6]. Assuming that the **.arpa** and **in-addr.arpa** zones will be DNSSEC-signed in a similar time frame, the situation of escalating loads being imposed on the servers for delegated subdomains of **in-addr.arpa** at each successive key rollover event will be curtailed. It would then be possible to configure the client with a single trust key, the public key signing key for the root zone, and allow the client to perform all signature validation without the need to manually manage other local trust keys.

There are two potential problems with this scenario.

The first is that for those clients that fail to remove the local Trust Anchor key set, these repeated queries may not go away. When there are multiple possible chains of trust, the resolver will attempt to validate using the shortest validation chain. As an example, if a client has configured the DNSKEY for, say, **test.example.com** into its local Trust Anchor key set, and it then subsequently adds the DNSKEY for **example.com**, the resolver client will attempt to validate all queries in **test.example.com** and its subzones using the **test.example.com** DNSKEY.

A more likely scenario is where an operator has already added local Trust Anchor keys for, say, **.org** or **.se**. When the root of the DNS is signed, the operator may also add the keys for the root to the local Trust Anchor set. If the operator fails to remove the local copies of the **.org** and **.se** Trust Anchor keys, in the belief that this root key value will override the **.org** and **.se** local keys, then the same validation failure behavior will occur. In such a case, when the local keys for these second-level domains become stale, their resolver will exhibit the same re-query behavior, even when they maintain a valid local root Trust Anchor key.

As a side note, the same behavior may occur when *DNSSEC Lookaside Validation* (DLV) is used. If the zone key management procedures fall out of tight synchronization with the DLV repository, it is possible to open a window where the old key remains in the DLV repository, but is no longer in the zone file. This situation can lead to a window of vulnerability where the keys in the DLV repository are unable to validate the signed information in the zone file, a situation that, in turn, introduces the same problem with re-query.

The second potential problem lies with the phase-in approach of signing the root. The staged rollout of DNSSEC for the root zone envisages a sequenced deployment of DNSSEC across the root server clusters, and through this sequence the root will be signed with a key that has no valid published public part, creating a *Deliberately Unvalidatable Root Zone* (DURZ).

What happens when a client installs this key in its local Trust Anchor set and performs a query into the root zone?

As an experiment, this DURZ key was installed into an instance of BIND 9.7.0rc2, with a single upstream root, pointing at the "L" root, the only instance of the 13 authoritative root servers enabled with DNSSEC signed data in February 2010. On startup the client made 13 consecutive DNSKEY requests, one to each of the root zone server addresses. When the client started its first query in a subzone, the client issued a further 156 DNSKEY queries in a period of 19 seconds, making 12 queries to each of the 13 root zone server addresses.

This scenario should sound familiar, because it is precisely the same query pattern as happened with the **`in-addr.arpa`** servers and the **`.se`** servers, although the volume of repeated DNSKEY queries is somewhat alarming. When the client receives a response from a subdomain that needs to be validated against the root, and when the queries to the root are not validatable against the local trust key, the client goes into a sequence of repeated queries that explore each potential validation path. Anchoring the local resolver with a key state that invalidates the signatures of all authoritative servers of the zone—but authoritatively (absent DNSSEC) confirms them as valid servers of the zone—places the client instance in an unresolvable situation: no authoritative Name Server that it can query has a signature that the client can validate, but the root zone informs it that only these Name Servers can be used.

Further tests of this behavior show that the client does not cache the outcome that the DNSKEY cannot be validated for a zone, and the client reinitiates this spray of repeated queries against the zone Name Servers when a subsequent DNSSEC query is made in a subzone. Therefore the behavior is promiscuous in two distinct ways. First it is evident that any Name Server so queried is repeatedly queried. Second, it is evident that all Name Servers of a zone are queried. The other part of the client response is not to cache validation failure for the zone in case this repeated query phase does not provide the client with a locally validated key.

After all, the data is provably false, so caching it would be to retain something that has been "proven" to be wrong.

The emerging picture is that misconfigured local trust keys in a DNS resolver for a zone can cause large increases in the DNS query load to the authoritative Name Servers of that zone, where the responses to these additional queries are themselves large, of the order of 1,000 bytes in every response. This situation can occur for any DNSSEC signed zone.

The conditions for the client to revert to a rapid re-query behavior follow:

- The *DNSSEC OK* (DO) bit is honoured by the server.
- The DNS data appears to be signed.
- The signature check fails.
- The client does not cache the validation failure for this zone.

The conditions being set up for the DURZ approach for signing the root follow:

- The DO bit is honoured by the server.
- The DNS data appears to be signed.
- The signature check fails.
- The client does not cache the validation failure for this zone.

What is to stop the DNS root servers from being subjected to the same spike in the query load?

The appropriate client behavior for this period of DNSSEC deployment at the root is not to enable DNSSEC validation in the resolver. Although this advice is sound, it is also true that many resolvers have already enabled validation in their resolvers, and are probably not going to turn off for the next 6 months while the root servers gradually deploy DNSSEC using DURZ.

But what load will appear at the root servers if a subset of the client resolvers starts to believe that these unvalidatable root keys should be validated?

What If…?

The problem with key rollover and local management of trust keys appears to be found in around 1 in every 1,500 resolvers in the **`in-addr.arpa`** zones. With a current client population of some 1.5 million distinct resolver client addresses each day for these **`in-addr.arpa`** zones, there are some 1,000 resolvers who have lapsed into this repeated query mode following the most recent key rollover of December 2009. Each subzone of **`in-addr.arpa`** has six Name Server records, and all servers see this pathological re-query behavior following key rollover.

The root servers see a set of some 5 million distinct resolver addresses each day, and a comparable population of nonupdated resolvers would be on the order of some 3,000 resolvers querying 13 zone servers, where each zone server would see an incremental load of some 75 Mbps.

Because the re-query behavior is caused by the client's being forced to reject the supposedly authoritative response because of an invalid key, and because DURZ is by definition an invalid key, the risk window for this increased load is the period during which DURZ is enabled, which for the current state of the root signing deployment is from the present date until July 2010. Because not all root servers have DNSSEC content or respond to the DO bit—and therefore do not return the unvalidatable signatures—the risk is limited to the set of DNSSEC-enabled roots, which is increasing on a planned, staged rollout. It has been reported that a decision to delay deployment of the DNSSEC/DURZ sign state to the "A" root server instance was made because this root server receives a noted higher query load for the so-called "priming" queries, made when a resolver is reinitialized and uses the offline root "hints" file to bootstrap more current knowledge. It is therefore likely that the "A" root server would also see increased instances of this particular query model, if the priming query is implicated in this form of traffic.

Arguably, this situation is unlikely. For most patterns of DNS query, failure to validate is immediately apparent. After all, where previously you receive an answer, you now see your DNS queries time out and fail.

However, because the typical situation for a client host (including *Dynamic Host Configuration Protocol* [DHCP] initialized hosts in the customer network space, the back office, etc.) is to have more than one listed resolver, there is the possibility of a misconfiguration being unnoticed during the period of a rolling deployment of DNSSEC-enabled services. In this situation if only one of the resolver's "nserver" entries is DNSSEC-enabled, either it is not queried or it is queried, but then passed over by the resolver timeout setting. Users see slower DNS resolution, but can attribute it to network delay or other local problems.

A second argument is that installation of hand-trust material is not normal, so the servers in question will be immediately known because a nonstandard process has to be invoked. Unfortunately, this situation is demonstrably not true. For example, the *Fedora*[5] release of Linux has included a simple DNSSEC-enabling process including a preconfigured trust file covering the reverse-DNS ranges. Because a previous release of this software included now stale keys (which have since been withdrawn in subsequent releases), any instance of *Fedora* for this release state being enabled will not only be unable to process reverse-DNS, it may also invoke this re-query mode of operation that places the server under repeated load of DNSKEY requests.

Because reverse-DNS is the "infrastructure" DNS query that is typically logged, but not otherwise used, unless the server in question is configured to block service on failing reverse (unlikely, given than more than 40 percent of reverse-DNS delegations are not made for the currently allocated IP address ranges), the end user simply might never notice this behavior. The use of so-called "Live CDs" can exacerbate this problem of pre-primed software releases that include key material that falls out-of-date. Even when the primary release is patched, the continued use of older releases in the field is inevitable. So perhaps this second argument is not quite as robust as originally thought.

Lastly, distinct from hand-installed local trust is the use of DNSSEC look-aside validation, which is known as DLV. This DNS namespace is privately managed and has been using the ICANN-maintained *Interim Trust Anchor Repository*, or ITAR. The DLV service is configured to permit resolvers to query it, in place of the root, to establish trust over subzones that exist in a signed state, but cannot be seen as signed from the root downward before the deployment of a signed root. There is now evidence that part of this query space exists, covering zones of interest to this situation. The **`.se`** zone key, for instance, is in the ITAR, as are the **`in-addr.arpa`** spaces signed by the RIPE NCC. Evidence suggests that if the DLV chain is being used and a key rollover takes place, some variants of BIND resolver clients fail to reestablish trust over the new keys until the client is rebooted with a clean cache state. This theory is difficult to confirm because as each resolver is restarted, the stale trust state is wiped out and the local failure is immediately resolved.

Post DURZ

Of course this phase is transitory, and even if there are concerns in terms of DURZ and queries to the root servers, all will be resolved when the root key is rolled to a validatable key on July 1, 2010.

Yes? Maybe not.

The current plan is to roll the root zone Key Signing Key every 2 to 5 years. The implication is that sometime every 2 to 5 years all DNS resolvers will need to ensure that they have fetched a new root trust key and loaded it into their resolver's local trust key cache.

If this local update of the root trust key does not occur, then the priming query for such DNSSEC-enabled resolvers will encounter this problem of an invalid DNSKEY when attempting to validate the priming response from the root servers. The fail-safe option here for the resolver client is to enter a failure mode and shut down, but there is a strong likelihood that the resolver client will try as hard as it can to fetch a validatable DNSKEY for the root before taking the last resort of a shutdown, and in so doing will subject the root servers to this intense repeated query load that we are seeing on the **`in-addr.arpa`** zone.

A reasonable question to ask follows: "Are there any procedural methods to help prevent stale keys from being retained during key rollover?" Reassuringly, the answer is "Yes." There is a relatively recent RFC, "Automated Updates of DNS Security (DNSSEC) Trust Anchors," RFC 5011[7], which addresses this problem.

RFC 5011 provides a mechanism for both signaling that a key rollover needs to take place and forward declaring the use of keys to sign over the new trust set to permit in-band distribution of the new keys. Resolvers are required to be configured with additional keying, and a level of trust is placed on this mechanism to deal with normal key rollover. RFC 5011 does not solve initial key distribution problems, which of course must be made out of band, nor does it attempt to address multiple key failures. Cold standby equipment, or decisions to return to significantly older releases of systems (for example, if a major security compromise to an operating system release demands a rollback) could still potentially deploy resolvers with invalid, outdated keys. However, RFC 5011 will prevent the more usual process failures, and it provides an elegant in-band rekeying method that obviates a manual process of key management that all too often fails through neglect or ignorance of the appropriate maintenance procedures to follow.

It is unfortunate that RFC 5011-compliant systems are not widely deployed during the lifetime of the DURZ deployment of the root, because we are definitely going to see at least one key rollover at the end of the DURZ deployment, and we can expect a follow-up key rollover within a normal operations window. The alternative is that no significant testing of root trust rollover takes place until we are committed to validation as a normal operational activity—a situation that invites the prospect of production deployment across the entire root set while many production operational processes associated with key rollover remain untested. The evidence from past concerns in resolver behavior is that older deployments have a very long lifetime for any feature under consideration, and because BIND 9.5 and older prerelease BIND 9.7 systems can be expected to persist in the field in significant numbers for some years to come, it is likely a significant level of pathological resolver behavior in re-querying the root services by active resolvers will have to be tolerated for some time.

It is also concerning that aspects of the packet traces for the **`in-addr.arpa`** zone suggest that for all key rollovers, albeit at very low levels of query load, some of the resolvers have simply failed to account for the new keys—and may never do so. Therefore, with increasing deployment of key validation, it is possible that a substantial new traffic class that grows, peaks, and then declines, but always declines to a slightly higher value than before, has to be borne, and factored into deployment scaling and planning.

Because this traffic is large—generating a kilobyte of response per query and potentially generally prevalent—it has the capability to exceed the normal response requirements for "normal" DNS query loads by at least one, if not two orders of magnitude. This multiplication factor of load is defined by the size of the resolver space and the number of listed Name Servers for the affected zone.

Mitigation at the server side is possible if this problem becomes a major one. The pattern of re-query here (the sequence of repeated queries for DNSKEY RRs) appears a potential signature for this kind of problem. Given that for any individual server the client times its repeat queries on the reception of the response from the previous query, delaying the response of the server to the repeated query will further delay the client's making its repeated query to this server. If the server were in a position to delay such repeated responses, using a form of exponential increase in the delay timer or similar form of time penalty, then the worst effects of this form of client behavior in terms of threats to the integrity of the ability of the server to service the "legitimate" client load could be mitigated.

Conclusion

It is an inherent quality of the DNSSEC deployment that in seeking to prevent lies, an aspect of the stability of the DNS has been weakened. When a client falls out of synchronization with the current key state of DNSSEC, it will mistake the current truth for an attempt to insert a lie. The subsequent efforts of the client to perform a rapid search for what it believes to be a truthful response could reasonably be construed as a legitimate response, if indeed this instance was an attack on that particular client. Indeed, to do otherwise would be to permit the DNS to remain an untrustable source of information. However, in this situation of slippage of synchronized key state between client and server, the effect is both local failure and the generation of excess load on external servers—and if this situation is allowed to become a common state, it has the potential to broaden the failure state to a more general DNS service failure through load saturation of critical DNS servers.

This aspect of a qualitative change of the DNS is unavoidable, and it places a strong imperative on DNS operations and the community of the 5 million current and uncountable future DNS resolvers to understand that "set and forget" is not the intended mode of operation of DNSSEC-equipped clients.

For Futher Reading

[0] A longer version of this article can be found in our online companion publication, *The Internet Protocol Forum*, `http://www.ipjforum.org/?p=226#more-226`

[1] Charalampos Patrikakis, Michalis Masikos, and Olga Zouraraki, "Distributed Denial of Service Attacks," *The Internet Protocol Journal*, Volume 7, No. 4, December 2004.

[2] Miek Gieben, "DNSSEC: The Protocol, Deployment, and a Bit of Development," *The Internet Protocol Journal*, Volume 7, No. 2, June 2004.

[3] O. Kolkman and R. Gieben, "DNSSEC Operational Practices," RFC 4641, September 2006.

[4] **http://www.unbound.net**

[5] **http://fedoraproject.org**

[6] **http://www.root-dnssec.org**

[7] M. St. Johns, "Automated Updates of DNS Security (DNSSEC) Trust Anchors," RFC 5011, September 2007.

[8] Geoff Huston, "Resource Certification," *The Internet Protocol Journal*, Volume 12, No. 1, March 2009.

[9] **https://www.isc.org/software/bind**

[10] **https://www.isc.org/community/blog/201002/signed-root-coming-and-what-means-you**

GEORGE MICHAELSON has a B.Sc. from the University of York and is a research scientist at the *Asia Pacific Network Information Centre* (APNIC), the Regional Internet Registry serving the Asia Pacific region. George explores problems in Internet Number Resource management, Internet standards, and network measurement by collaborative research. George has more than 28 years experience in computer science, networking, ICT administration, and research conducted in Australia and the UK. He participates in standards development in the IETF and has been a working group chair as well as an RFC author. He is a member of the *British Computer Society*. E-Mail: **ggm@apnic.net**

PATRIK WALLSTRÖM is a senior researcher at .SE, the Internet registry for SE domain names, and has been with the registry for eight years developing registry systems and working with the deployment of DNSSEC. Patrik is currently working on the *OpenDNSSEC* project, producing tools for a wider deployment of the technology. At .SE he is also managing the *Healthcheck* project, a new open source platform for measuring the quality of DNS, E-mail, Web and IP within Sweden. Patrik is also a board member of the *Swedish Network Users' Society* (SNUS). E-mail: **pawal@iis.se**

ROY ARENDS is a senior researcher at Nominet UK, the Internet registry for UK domain names. He co-authored several IETF standards on DNSSEC, resides on the board of DNS-OARC, is a member of ICANN's *Security and Stability Advisory Committee*, and is part of IETF's DNS-Directorate. As an expert on DNS and DNSSEC, Roy has co-initiated several DNS-related open source projects, such as *Unbound* and *OpenDNSSEC*. In the past, Roy was a member of, and chaired, CERT-NL. E-mail: **roy@nominet.org.uk**

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. The author of numerous Internet-related books, he is currently the Chief Scientist at APNIC. He was a member of the *Internet Architecture Board* (IAB) from 1999 until 2005, and served on the Board of the Internet Society from 1992 until 2001. E-mail: **gih@apnic.net**

Extending Router Lifetime with Virtual Aggregation

by Paul Francis, Max Planck Institute for Software Systems, and Xiaohu Xu, Huawei Technologies

Biologists believe that human life is limited by the number of times cells can replicate; noncancerous cells have a kind of internal counter that prevents them from replicating forever. Even if humans are kept healthy in every respect, they will eventually die simply because their cells will cease to replicate. Internet routers also have a finite lifetime. They are built with a fixed amount of hardware memory for storing the forwarding table (the memory structure that tells the router where to forward any IP packet, also called the *Forwarding Information Base* [FIB]). As the Internet global routing table grows, it eventually overflows the FIB, and the router ceases to be able to hold the full routing table. Even if the router is healthy in every respect (all of its hardware components still operate), it can no longer function as a router in the Internet *Default-Free Zone* (DFZ), where no default routes can be used.

In the past, router vendors have been reasonably good at predicting how long FIBs will last because the growth of the global DFZ routing table has stayed fairly predictable. As a result, *Internet Service Providers* (ISPs) can plan their capital budgets, and where necessary use a set of tricks (discussed in the next section) to squeeze additional life out of routers even after their "FIB death." But there are two problems.

First, these tricks work only in limited situations, they require extra configuration, and they can lead to increased traffic loads. Second, and potentially much more serious, the rate of routing table growth may dramatically accelerate in the near future, thus shrinking the lifetime of the installed router base. This expected acceleration is due to the imminent exhaustion of IPv4 addresses. In the past, address authorities such as the *American Registry for Internet Numbers* (ARIN) could assign large contiguous blocks of addresses to ISPs, which in turn assigned smaller blocks to their customers. Therefore, routers in other ISPs' networks need only a single routing table entry—that of the large block—to route to destinations in the ISP. This approach to scaling is called *address aggregation*. There is a fear that, as IPv4 addresses become increasingly unavailable, ISPs will start buying and selling smaller and smaller blocks of IP addresses from each other in an effort to squeeze out as many addresses as possible. These small blocks will appear all over the Internet thus significantly increasing the size of the routing table.

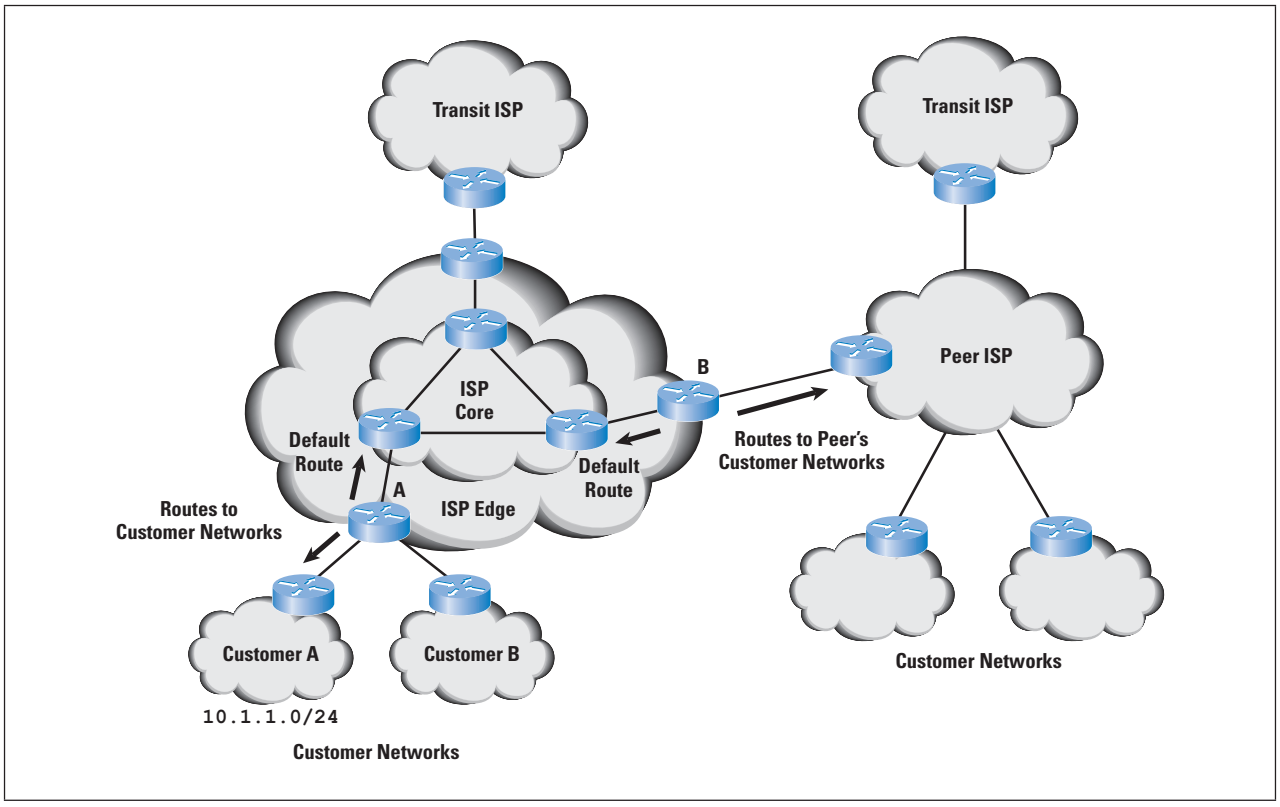
This article describes a new routing technology, called *Virtual Aggregation* (VA), which mitigates these problems. It makes extending the lifetime of old routers much easier, and makes it possible for existing routers to absorb a surge in the routing table size. Virtual Aggregation is a working item in the *Global Routing Operations Working Group* (GROW) working group of the IETF[7], and is documented in `draft-ietf-grow-va`[6] and related drafts.

Tricks for Keeping Old Routers Deployed

ISPs frequently want to extend the usefulness of a router beyond its "FIB death," and there are many tricks for doing just this. The most common is to structure the ISP in a core-edge arrangement. In this setup, a core of routers forms the backbone of the network. Edge routers connect to other networks and feed into the core. In many cases these edge routers do not need to know how to route to everything in the Internet. Rather, they often need to know only what addresses are reachable in their directly connected networks.

For instance, Figure 1 shows an ISP whose edge routers connect to three types of other networks: customer networks, peer ISP networks, and transit ISP networks. Each customer network has only one or a small number of address prefixes. The edge routers connecting customer networks must know what addresses are reachable in the customer networks, but everything else can be "default routed" to the core. Likewise, the routers connected to peer ISPs need to know how to route to the peers' customer addresses. Everything else can be defaulted to the core. The core routers and the edge routers that connect to transit ISPs, however, need to know how to route to everything.

Figure 1: With a core-edge style of deployment, some routers need to keep full routing tables, while others can keep partial routing tables and default route everything else to the ISP core.



A common practice is for ISPs to delegate FIB-dead routers to the customer or peer edges, and to have the core routers filter the routing information given to the edge routers. For instance, router A in Figure 1 learns the addresses reachable in customer network A (say, `20.1.1.0/24`) and conveys them to the ISP core, but the core tells router A only that "everything else" is reachable through it (`0.0.0.0/0`). But what if customer A itself wants the full DFZ routing table? For instance, customer A might be multihomed to some other ISP, and might want to know which Internet destinations are best reachable through each ISP. To do this, it needs to receive the whole routing table from each ISP, a situation that, of course, cannot happen if the core withholds routes from router A.

As another example, what if two peer ISPs later decide that they want to offer transit service to each other? Now additional routes need to be conveyed to the peer-connected edge routers (router B), and this process may not be possible with limited FIB.

Another way an ISP can shrink its routing table is to default route to its transit ISPs. For instance, routers keep track only of how to route to customers and peers, and everything else is defaulted to the transit ISPs. When this default routing is done, even an ISP's core routers do not need the full routing table. A simple approach is for an ISP to send all defaulted packets to the nearest transit ISP. This process, however, may result in many packets taking a longer Internet path than necessary. Reference [1] describes a more complicated approach where the ISP maintains "semidefaults" for different transit networks in order to improve its global routing while reducing routing table size by about half. This approach, however, can be hard to manage.

In addition, any form of ISP-level default (simple or complex) results in sending extra traffic to the transit ISPs. A substantial amount of Internet traffic is targeted to nonroutable prefixes. When an ISP has the full routing table, it can identify this traffic and drop it before sending it to its transit ISPs. When an ISP defaults, it sends this traffic to its transit ISPs, and pays for it.

To summarize, dealing with FIB-dead routers leads to more complex management, limitations in business arrangements with peers and customers, poor paths over the Internet, and increased traffic load.

The Idea of Virtual Aggregation

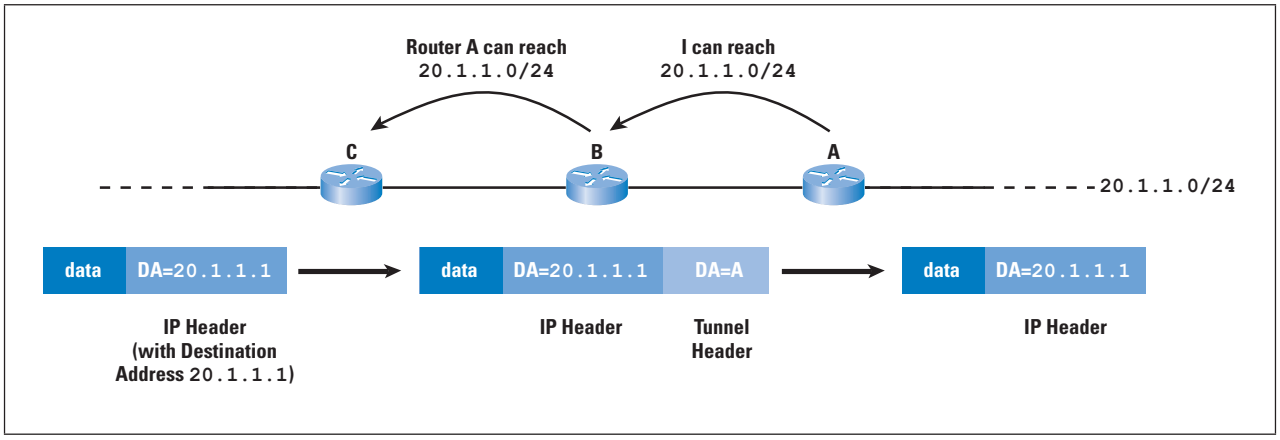
In its simplest form, Virtual Aggregation allows an ISP to use FIB-dead routers as edge routers, in any edge router position (neighbor is a transit provider, a peer, or a customer) without limiting what routing information is exchanged. Configuration requirements are minimal. In a more complex form, Virtual Aggregation allows all ISP routers (not just edge routers) to be FIB-dead routers, without requiring ISP-level default routing.

Virtual Aggregation uses two basic mechanisms, FIB suppression and tunneling. Before discussing FIB suppression, a small amount of background is needed. Internet routers have a "data plane" and a "control plane." The data plane is what forwards packets, and includes such functions as header parsing, FIB lookup, queuing, and packet transmission. The control plane operates the background protocols that gather much of the information needed by the data plane. Examples include routing protocols such as the *Border Gateway Protocol* (BGP) and *Open Shortest Path First* (OSPF), and tunnel establishment protocols such as the *Label Distribution Protocol* (LDP).

The idea of FIB suppression is that the control plane operates as normal, but that certain routing table entries are not loaded into the FIB. This idea exploits the fact that it is (data plane) FIB memory, not control plane routing table memory that is the more severe bottleneck. By allowing the control plane to operate as normal, no changes are required to routing protocols or, for the most part, the management of routing protocols.

Tunneling is used to pass packets through routers that have suppressed FIB entries. The principle is illustrated in Figure 2. Here router A tells router B that it can reach `20.1.1.0/24`. Router B in turn tells router C that router A can reach `20.1.1.0/24`. As a result, router C tunnels packets destined for `20.1.1.0/24` to router A through router B. In other words, it wraps the IP header in another IP or a *Multiprotocol Label Switching* (MPLS) header that first gets the packet to router A. Router A strips that header, and sends the packet toward the destination. Notice that router B can suppress the route to `20.1.1.0/24` from the FIB—it only needs to know how to route the packet to router A. In other words, even though router B fully participates in the control plane, it is able to shrink its FIB through FIB suppression and tunneling.

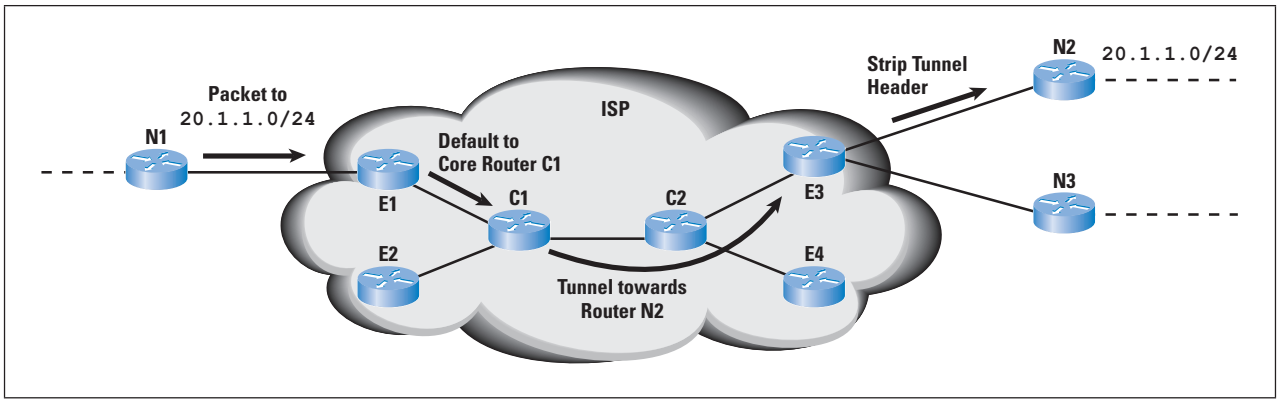
Figure 2: Because router C tunnels the packet to router A, router B does not need to know how to forward packets with addresses in ***`20.1.1.0/24`****.*



Virtual Aggregation in Practice, Simple Version

In the simplest version of Virtual Aggregation, a core-edge configuration is used. The core routers maintain full FIB tables. The edge routers FIB-install at least a default route to the core, and potentially additional routes if there is space in the FIB. This process is illustrated in Figure 3. Here there are two core routers, C1 and C2, and four edge routers, E1, E2, E3, and E4. The edge routers have external neighbors, N1, N2, and N3, as shown.

Figure 3: Packets can be delivered to **`20.1.1.0/24`** *even if none of the edge routers has a FIB entry for* **`20.1.1.0/24`**.



The operation is best explained by example. Suppose that N2 advertises a route to destination `20.1.1.0/24` to E3 using *External BGP* (eBGP) and giving itself as the next hop. E3 in turn advertises this route to the other internal routers using *Internal BGP* (iBGP), with the next hop still as N2. The core routers install this route in their FIBs, with an indication that packets matching the route should be tunneled to the next hop, N2. Assume for now that all edge routers FIB-suppress the entry. When a packet for say `20.1.1.1` arrives at E1 from N1, E1 does not find an entry for `20.1.1.0/24`, but does find the default route `0/0` telling it to forward the packet to its core router C1. C1 looks into its FIB and indeed finds an entry for `20.1.1.0/24` telling it to tunnel the packet to N2. C1 wraps the packet in another header, typically IP or MPLS, addressed to N2. When the packet reaches E3, however, E3 notes that the header directs it to send the packet to N2, strips off the outer header, and sends the packet to N2. E3 can do this without a FIB entry for `20.1.1.0/24`.

MPLS already has all the mechanisms needed to perform this packet forwarding. E3 can use LDP to signal a *Label Switched Path* (LSP) to N2, and *Penultimate Hop Popping* can be used to strip off the MPLS header before forwarding the packet to the external neighbor N2 (as described in section 4.1.4 of [4]).

Alternatively, stacked MPLS label technology can be used; for example, the inner label is signaled with BGP (see "Carrying Label Information in BGP-4"[3]) while the outer label is signaled with LDP. Here E3 sets itself as the next hop for all the routes learned from external neighbors (for example, `20.1.1.0/24`) when advertising them to its iBGP peers, and uses the inner label to identify the external neighbor (see section 4.3, "Label Stacks and Implicit Peering" of [4]). IP-in-IP tunneling can also be used, in this case signaled with softwires BGP attributes[5].

Now let's see what happens if a packet to `20.1.1.1` is received by E3 from external neighbor N3. If E3 has not FIB-installed the route for `20.1.1.0/24`, it uses its default entry and forwards the packet to C2. C2 finds its entry for `20.1.1.0/24`, which instructs it to tunnel the packet to N2. The packet is sent back to E3, which strips off the outer header and delivers the packet to E2. In this case, the packet has traveled an extra hop and back, a process that is not acceptable if done too much. As long as there is space in the FIB, however, routers are free to FIB-install additional routes. A good policy is to always install routes when external neighbors are the next hop. This policy avoids the longer path. In some cases, such as edge routers that connect to transit networks, there may not be enough FIB space to hold all routes from all external neighbors. In this case, the router may FIB-install the routes for which the most traffic is forwarded. Studies have shown that a small number of routes account for majority of the traffic, making Virtual Aggregation a very efficient solution[2].

Note that this simple form of Virtual Aggregation is very easy to configure. Essentially all that is needed is to tell the routers that they are using simple Virtual Aggregation, and to tell them if they are a core or an edge router. The routers can automatically configure everything else. Virtual Aggregation requires configuration of tunnels from every router to every other router, but these configurations also can be automatic. In any event, increasingly these tunnels are created anyway for the purpose of traffic engineering.

Simple Virtual Aggregation solves most of the problems described earlier. It can save FIB on any edge router without having to compromise BGP service to customers or flexibility in using peer networks for some transit. It also allows FIB-dead routers to be used as edge routers with transit ISPs. Finally, it prevents the need for ISP-level default routing to transits, thus avoiding unnecessarily sending unroutable traffic to the transit. And it does all this with much less configuration than is required to operate with FIB-dead routers today.

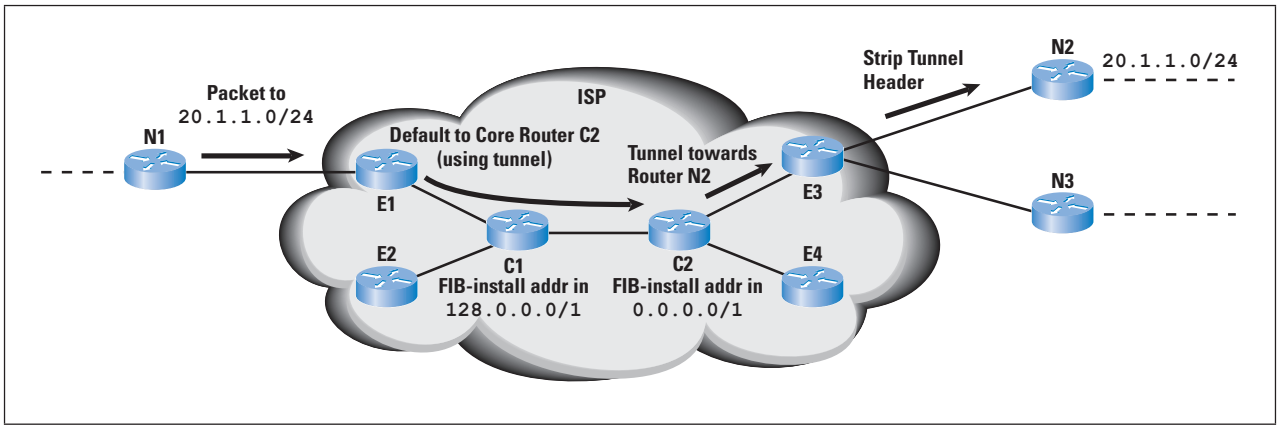
Virtual Aggregation in Practice, Complex Version

The simple version of Virtual Aggregation is satisfactory for edge routers, but it does nothing to reduce FIB size on core routers. What if an ISP wishes to also extend the lifetime of its core routers? Or wants to move away from a core-edge model, and rather connect all edge routers directly through a Layer 2 substrate like MPLS?

What if indeed there is a surge in routing table growth, thus causing ISPs all over the world to suddenly find themselves FIB-starved? There is a version of Virtual Aggregation that allows for FIB reduction in any and all routers in an ISP network.

The basic idea is to divide the address space so that different routers maintain full routes within different parts of the address space. So for instance, rather than have all core routers responsible for all of the address space, you could have half of the core routers responsible for the lower half of the address space, and the other half of the core routers responsible for the upper half of the address space. Figure 4 shows how this setup would look for the simple topology of Figure 3, keeping in mind that this example is rather simplistic.

Figure 4: In a complex version of Virtual Aggregation, even core routers do not need to hold the full routing table.



Assume that C2 FIB-installs only the lower half of the address space (`0.0.0.0/1`) and C1 FIB-installs the upper half (`128.0.0.0/1`). With this arrangement, the edge routers have two defaults instead of one. Packets to addresses in `0.0.0.0/1` are defaulted, through a tunnel, to C2, and packets to addresses in `128.0.0.0/1` are defaulted to C1. These defaults are learned simply by having C1 and C2 advertize their respective default routes with themselves as the next hop in iBGP.

As with the previous example, assume that router N2 advertises a route to `20.1.1.0/24`, with itself as the next hop, to E3. E3 advertises this route to all other routers using iBGP. Only C2, however, FIB-installs this route—C1 suppresses it. When a packet to `20.1.1.1` arrives at E1, it looks in its FIB, finds a matching route to `0.0.0.0/1`, and so tunnels the packet to C2. C2 terminates the tunnel, finds its FIB entry for `24.1.1.0/24`, and tunnels the packet toward N2. E3 uses the tunnel information to know to forward the packet to N2, strips away the tunnel header, and forwards the packet to N2.

Now suppose that a packet for `20.1.1.1` arrives at E3 from N3. Ideally E3 has already automatically FIB-installed the route for `24.1.1.0/24` either because its external neighbor provides the next hop, or because the route is a high-volume destination. In this case, of course, the packet is directly forwarded to N2. However, if E3 has not FIB-installed the route, then its best match is the default to `0.0.0.0/1`, and it tunnels the packet to C2. C2 in turns tunnels the packet back toward N2 through E3 as described before. Worse, if C1 rather than C2 FIB-installed the lower half of the address space, the packet would have detoured all the way to C1. Clearly these routes are not optimal, and so we must ask how nonoptimal would the complex version of Virtual Aggregation be in real ISPs.

The USENIX NSDI paper[2] answers this question for one large transit ISP. In this study, both the topology and the traffic matrix of the ISP are considered. The deployment strategy is substantially more complex than the simplistic example given previously. An upper limit is placed on the maximum increase in latency (5 ms) for any path through the ISP. There is a requirement that within a *Point of Presence* (PoP) at least two routers must cover the same address space. The number and size of address partitions are engineered to spread FIB load evenly. The "additional" routes installed in the FIB are designed to cover high-traffic destinations to the extent possible.

With these requirements in mind, this study found that FIB size could be reduced in all routers by at least an order of magnitude with a negligible increase (1–2%) in overall traffic load due to the occasional extra hops from the detours. This result ultimately translated into an increased router lifetime of easily 10 years.

The management requirements for the complex version are substantially greater than those for the simple version. The address partitions must be chosen, the routers assigned to address partitions must be chosen, and possibly some strategy for deciding what "additional" routes should be FIB-installed is needed. Whether this added configuration and the associated difficulties due to, for instance, misconfiguration are worth the cost savings for extending router lifetime is up to each ISP. Virtual Aggregation at least provides an option that was not previously available.

Status

Virtual Aggregation is a working-group item in the *Global Routing Operations Working Group* (GROW) in the IETF. The primary draft is `draft-ietf-grow-va`[6]. This draft has gone through several revisions, and is very close to its final form. Huawei is currently implementing Virtual Aggregation. A second open-source implementation has been built by Paul Francis' research group for the *Quagga* open-source routing platform, and is still being enhanced.

Acknowledgements

The authors would like to thank the co-authors of the Virtual Aggregation drafts, Hitesh Ballani, Dan Jen, Robert Raszuk, and Lixia Zhang. In particular, it was Robert who suggested the simple version of Virtual Aggregation.

References

[1] Andre Chapuis, "BGP Filtering," Presentation from SWINOG7, **www.swinog.ch/meetings/swinog7/BGP_filtering-swinog.ppt**

[2] Hitesh Ballani, Paul Francis, Tuan Cao, and Jia Wang, "Making Routers Last Longer with ViAggre," USENIX NSDI 2009, April 2009.

[3] Y. Rekhter and E. Rosen, "Carrying Label Information in BGP-4," RFC 3107, May 2001.

[4] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, January 2001.

[5] P. Mohapatra and E. Rosen, "BGP Encapsulation SAFI and BGP Tunnel Encapsulation Attribute," RFC 5512, April 2009.

[6] P. Francis, X. Xu, H. Ballani, D. Jen, R. Raszuk, and L. Zhang, "FIB Suppression with Virtual Aggregation," October 2009, **draft-ietf-grow-va-01**.

[7] IETF Global Routing Operations Working Group (GROW), **http://www.ietf.org/dyn/wg/charter/grow-charter.html**

PAUL FRANCIS is a faculty member at the Max Planck Institute for Software Systems in Germany. He has been active periodically in the IETF for nearly 20 years. Dr. Francis has held research positions at Cornell University, ACIRI, NTT Labs, and Bellcore, and was Chief Scientist at Fast Forward Networks and Tahoe Networks.
E-mail: **francis@mpi-sws.org**

XIAOHU XU graduated from Beijing University of Posts and Telecoms in 2000. He has been working in the telecom industry for about 10 years and now is a research engineer with IP Advanced Technology Research Department of Huawei Technologies. Before joining Huawei at the end of 2004, he was the chief engineer of the Technical Support Department for Harbour Networks.
E-mail: **xuxh@huawei.com**

RFC Editor in Transition: Past, Present, and Future

by Leslie Daigle, ISOC

In April 2009, the *Request For Comments* (RFC) Editor published RFC 5540[1], "40 Years of RFCs," which summarized the publication history of the RFC Series. The series has been the technical publication series for Internet technology since long before there was an *Internet Engineering Task Force* (IETF). Although the RFC Series is the publication vehicle for the IETF, it has been, and remains, scoped more broadly than that (refer to RFC 4844[2], "The RFC Series and RFC Editor").The RFC Series is the archival series dedicated to documenting Internet technical specifications, including general contributions from the Internet research and engineering community as well as standards documents.

For the past three of the four decades of the history of the series, the RFC Editor work has been carried out at the *University of Southern California Information Sciences Institute* (USC/ISI). The RFC Editor role now faces another evolutionary step: The work involved in managing the overall series is being split up to recognize the different components of the editing, production, and archiving activities and to lay the groundwork to ensure its continued success, as outlined in RFC 5620[3], "RFC Editor Model (Version 1)."

At the IETF 76 plenary in Hiroshima, Japan, in November 2009, USC/ISI and the role it has played in supporting the RFC Editor over the past 30 years were given special recognition. Some members of the team will move from USC/ISI to the RFC Editor's new home, where they will continue their work. We took the opportunity to talk with current and future RFC Editor staff and advisory board members, including current RFC Editor staff members Bob Braden, Sandy Ginoza, and Alice Hagens, as well as Bob Hinden, who is a member of the RFC Editor advisory board.

The People Behind the RFC Editor

Jon Postel was the first RFC Editor, starting the position in 1969 as an activity to keep track of RFC Series documents. Bob Braden, who was then part of the *Advanced Research Project Agency Network* (ARPANET) research program, told how he got started with the RFC Series: "I wrote my first RFC in the early 1970s, when it was somewhere around RFC 100. I was at that point manager of programming for the Computing Center at the *University of California, Los Angeles* (UCLA), and *Advanced Research Projects Agency* (ARPA) wanted to connect it to ARPANET as a resource." This was all pre-TCP/IP, and Bob's staff had to implement file transfer and Telnet. At the same time, Jon was a graduate student at UCLA, and Bob worked with him as a colleague. It was before Jon got his Ph.D. and moved to SRI in 1973–1974. In 1980, Jon moved to USC/ISI, taking the RFC editorship with him. Joyce Reynolds went to work for Jon at USC/ISI. She did much of the actual editing and became an important part of making the RFC Editor activity viable.

Jon was responsible for quality control, running the operation, and generally being the series editor. When Jon died suddenly in 1998, Bob, who joined USC/ISI in 1986, and Joyce both felt a keen sense of loss. "Jon was a very remarkable guy in many ways," Bob said. "We knew how much the RFC Series meant to Jon, and we volunteered to carry it on."

Sandy Ginoza joined USC/ISI to work on the RFC Editor activity in 1999, just after Jon passed away. Alice Hagens came onboard in 2005, taking on more of the computer-oriented aspects of the work.

RFC Series

Although we tend to reference and read individual RFC documents, it is important to understand that there is significant value in the collection of published RFCs as a *series*. On the importance of the RFC Series, Bob Hinden said, "This community is IETF-focused, but to the larger world not centered around the IETF; it's really the RFCs that are how you build the Internet. One of the things that made the Internet possible was the RFC Series: that you could build things and deploy things without coming to IETF meetings was valuable." Bob went on to outline his own experiences, such as meeting engineers in Taipei, for whom it was the first time they had ever met anyone who had written an RFC. Even the notion of going to an IETF meeting was in another dimension. "The RFC Series is what enables people to build products, networks, and the Internet," he said.

And it is quite an active series. Currently, some 300 documents (10,000 pages) are published every year, and although it might be interesting to review the material to detect trends or arcs of work in the Internet technical community, that type of activity is beyond the current scope of the RFC Editor. Focusing on consistency of the series, Bob Braden wondered, "Will we eventually have good enough statistics from the errata system to gauge our error rate?"

The intent of the RFC Series is to serve the broader Internet community; it is not just for or by the IETF. Sandy's perspective on the value of the *Independent Stream* of RFCs is that "it offers an alternate view than what happens in the IETF and what working groups have decided to take on as part of their chartered activities. It's good to document that work was done, results were generated, lessons learned, etc. 'We tried it; don't do it this way.' We often get asked why it's called RFC when we're not really requesting comments anymore, but that is the genesis, and the Independent Stream keeps some of that alive."

Bob Braden offered his own perspective on the Independent Stream.

"Historically, the RFC Series is supposed to be larger than the IETF, and while Jon was alive, the editor did whatever he thought he ought to do; the community didn't question it much."

However, in the absence of Jon as an authority figure, the community began to ask questions and build its own set of beliefs, eventually coming to believe that RFCs were only for the IETF. That matter was resolved with RFC 4846[4], which explained that there is a separate set of independent submissions that do not come through the IETF.

"It's not a big stream, not a lot of documents, but it is important philosophically," Bob added. "The Internet community is bigger than the IETF."

The RFC Series is, nevertheless, entwined with the IETF and its activities. For instance, the discussion of (IETF) *Intellectual Property Rights* (IPR) has led to an impasse in assigning boilerplate to RFCs that allow the continued publication of the Independent Stream documents. That subject is being worked on and resolved, but it offers an example of some of the complexities—and frustrations—that can arise as part of the RFC Editor process. "The current situation—that the independent submissions cannot be published because we don't know what the boilerplate is—is just terrible," said Bob Braden.

Bob Hinden, who has been tracking the IPR work from the IETF side, agreed and elaborated on some important lessons learned: "The IETF created a process in the IPR working group that focused on trying to provide a solution to what they perceived as a problem. But they lost sight of the complexity and cost of implementing that solution compared with the actual risk of something bad happening. We have learned a lot about doing this in the future. This isn't like a protocol spec where you fix a bug in the finite state machine. This has a real effect on people doing stuff. When you ask for legal opinions you get the answer about how to solve the problem, but that's not the end of the process. You need to balance the cost of solving the problem with the risk of what you're trying to avoid. Lawyers are supposed to give you the lowest-risk answer. You need to follow through with questions about likelihood and consequences. This is all great hindsight, and I hope we can apply it in the future." Hinden also said he believes the current impasse could have been avoided if the new procedure had specified that it go into effect when appropriate supporting conditions were met, instead of on a specific flag day, such as the date of publication of the RFC.

The effects of entwining the RFC Series and the IETF go both ways. For example, the RFC Series recognizes three levels of standards documents: *Proposed, Draft,* and *Full*. The expectation, documented in the IETF standards process, is that standards-track specifications should be published as Proposed and then advanced to Draft and Full as the specification gets tested commercially and acknowledged as appropriately mature to move to the next stage.

In reality, as observed at the IETF 76 plenary, many of the important specifications that form the basis of the operating Internet are still published only as Proposed Standard. Bob Braden explained the history of the standards-track RFC maturity system this way: "Labels were invented whole cloth by the original *Internet Architecture Board* (IAB), who were a bunch of academics. At that point the Internet had not been commercialized—there were no commercial pressures—so we imagined that it made sense to step through progressions in a theoretical world. In the real world, companies are putting out products. There is no financial incentive for people to spend time advancing documents. Plus, the IETF is so large and there are so many working groups that we try to dispatch them as fast as we can; there is no one around to advance a document." There have been, and will continue to be, proposals for moving important, current standards (such as the *Border Gateway Protocol* [BGP]) forward in maturity or for collapsing the maturity scale and labeling system.

On the fun side of the RFC Series, there remains a tradition of "April 1st" RFCs. "That people want to participate in that is cool," said Sandy. "And we get to see the runners-up and the really-not-so-good ideas!"

Alice agreed, adding that "there are high standards for straight-faced satire."

RFC Editor

Traditionally, the RFC Editor has not only populated the series with new (approved) documents but also kept all the threads together in the RFC Series. Describing the origins of the role, Bob Braden pointed out that "Originally, Jon was prince of his kingdom. As RFC Editor, he was an honorary member of the IAB informally called the *Protocol Czar.* He used the RFC Editor position to actively prevent bad ideas from getting pushed. Jon imposed a consistency of style on the document series. You pick up RFC 1001 and compare it with 2001, and they look very similar." Jon believed, and the RFC Editor continues to believe today, that consistency was a worthwhile attribute, promoting stability in the series.

Reflecting back, Bob Braden said, "In discussions over the last five years, people have expressed the view that we don't need an RFC Editor—just take an Internet Draft and publish it. That notion drives me crazy. The implication is that it doesn't matter whether it is good English, correctly referenced, consistent, etc. I can't stand that view." One of the arguments for such an approach to IETF document publishing is that editing can inadvertently alter, and thereby introduce errors to, text. But the RFC Editors understand that.

Alice said changes to text can be problematic, "partly because of the technical content and partly because it is a group process. It's agreed-upon text. The idea is how precious the text is and how a slight change can make a large difference."

Sandy agreed, adding that "for as many changes that get pushed back upon, there are many that make it through the process: for as many people as look at the document before it gets to us, there are things that escape them; there is often missing text, missing words." According to Alice, with working group documents, people often focus on getting the technical ideas right, but nobody has read the text from beginning to end. In addition, many in the community are not native English speakers. It all comes back to the consistency and professionalism of the output of the series.

RFC Editing Process

As the RFC Series has grown, achieving consistency has required the creation and refining of processes. "When Joyce and I took over," said Bob Braden, "we built the website and regularized a lot of things, and the community began to ask, 'Why do you do it that way?'" In response, the editors started publicizing the *Style Manuals* they used. Joyce and Bob generated a lot of rules that have become institutionalized.

Of course, there is continuing evolution. Bob Braden noted that the addition of errata was his idea, although "it has turned out to be a much, much bigger deal than ever imagined, as is often the case," he said, laughing. "Now we're talking about adding image files to solve the problem of incorporating graphics in an ASCII RFC. John Klensin and I generated a plausible solution for that, and we hope to get it installed soon."

It is important to note that there are some edits the RFC Editor will not make. According to Sandy, the RFC Editor tries to ensure consistency of terminology and to make recommendations that improve consistency within a document, both in a technical sense and within the series. "We don't change the active/passive voice," she said.

"We might suggest it, but we are concerned that it would affect the author's intent." Being conservative is critical. Sandy said she was surprised by how "simple grammatical changes can have a serious technical effect; placement of a comma can make a big difference in how people read the document and what they implement."

Working with authors is an important part of making the editing process successful. Innovations such as having the *RFC Editor Help Desk* at IETF meetings and making the AUTH48[5] (final check of the RFC Editor's edits) more of an interactive dialogue have helped build community and create awareness of how to build a better document that conveys the meaning as intended. "It is extremely useful to get discussions started earlier, which lessens problems during AUTH48," said Alice. She added that it has also been useful to have face time with the developers of community-created tools, such as *xml2rfc*[6] and the *Augmented Backus–Naur Form* (ABNF) checker, which have been instrumental in improving RFC production. Office hours, building relationships, and face time "all help make it about working together," said Sandy.

Looking forward, Sandy said she would like to see the RFC editing process (and series) "grow and continue to be more consistent, with better community relations and more transparency so authors can look at our site and better understand the process, instead of thinking their document has gone into a black box."

On the Verge of Major Change

As this article is written, the RFC world is on the brink of major structural change. Following IAB-led community discussion, there is a new model for recognizing the components of activity that make up the RFC activities. ISI is handing off the RFC Editor activity, which will be taken up by separate organizations working together. In February 2010, the IAB appointed Nevil Brownlee as the *Independent Submissions Editor* (ISE) and Glenn Kowack as the Transitional *RFC Series Editor* (RSE). In October 2009, *Association Management Solutions* (AMS) was awarded 2-year contracts to manage the RFC Production Center and the RFC Publisher.

Sandy will be joining AMS as RFC Production Center director and Alice will be joining as senior editor and information technology development project manager. To the question of whether the current RFC advisory board will carry forward in the current format or will change, Bob Braden answered, "The current board serves two functions: It provides a supply of experienced people who review independent submissions, but it also gives the RFC Editor advice on policy matters. Some members of the advisory board are very strong members of the IETF in terms of policy advice. In forming the board, I tended to identify a subset of people within the IETF who have long IETF and publishing experience. In the new world there will be an *RFC Series Advisory Group* (RSAG), which will take over the policy discussions that are currently being conducted by the editorial board. In practice it will be the same people, at least for a while, but with separate duties. That separation is useful."

In considering the change of organizations, Sandy said the biggest thing in moving to AMS is that it is a more service-oriented environment. "In the new model," she said, "it is important that the ISE and RSE be respected individuals who are granted some of the independence the RFC Editor had at ISI."

Alice added that the institutional memory of the RFC Editor function will not be lost with the move to AMS. "Sandy has worked side by side with Bob Braden for 10 years, and much of the process is written down in the document series. I'm confident that the continuity of the series won't be lost by the move to AMS."

Bob Hinden offered another perspective. "I think one of the positive things that has come out of the new model that has gotten lost is this: A lot of people in the IETF didn't understand where the series had come from, or why the IETF chose to use it," he said.

"It is the formalization that there are different streams that have different rules. Before, this was confused with the IETF standards process. Going forward we'll have the opportunity to use the RFC Series for other relevant Internet publication streams that have not been part of IETF. Now we have a framework that would allow that."

Although it is on the verge of major changes, the RFC Series and RFC Editor functions are clearly continuing what has been a long process of constant evolution and change. This transition is just a new chapter in the history of the series.

[Ed.: This article is composed of interviews conducted by Leslie Daigle and Lucy Lynch, and notes compiled by Mat Ford. The original version was published in *The IETF Journal,* Volume 5, Issue 3, January 2010 and has been been updated for use in IPJ. *The IETF Journal* can be obtained from `http://isoc.org/ietfjournal/`]

For Further Reading

[1] RFC Editor, "40 Years of RFCs," RFC 5540, April 2009.

[2] L. Daigle, Ed., Internet Architecture Board, "The RFC Series and RFC Editor," RFC 4844, July 2007.

[3] O. Kolkman, Ed., IAB, "RFC Editor Model (Version 1)," RFC 5620, August 2009.

[4] J. Klensin and D. Thaler, Eds., "Independent Submissions to the RFC Editor," RFC 4846, July 2007.

[5] `http://www.rfc-editor.org/pubprocess.html`

[6] Marshall T. Rose and Carl Malamud, "Writing Internet Drafts and RFCs Using XML," *The Internet Protocol Journal,* Volume 10, No. 1, March 2007.

Alice Hagens, Bob Braden, and Sandy Ginoza are recognized at IETF 76 for their work with the RFC Editor. (Photo: Internet Society)

Fragments

IETF Outcomes Wiki Launched

As an organization, the *Internet Engineering Task Force* (IETF) measures its success by its publication of RFCs (see previous article). It does not explicitly ask itself whether published work is adopted and used by the greater Internet community. The IETF's dialogue about success started to change with the production of RFC 5218, "What Makes for a Successful Protocol?"[1] which documented case studies and empirical data about some of the factors that appear to correlate with success, in terms of community uptake for IETF work.

Taking a different approach in assessing long-term IETF impact, another tool is now available: A wiki that lets community participants list the success or failure of significant standards. The *Outcomes Wiki*[2] divides listings according to the "areas" used for managing technical work in the IETF, such as Applications or Transport. Outcomes are rated according to a 6-point scale, ranging from "complete failure" to "massive adoption, plus extensive derivative work."

The wiki began in June 2009, as an independent effort among a small set of IETF participants, to test its feasibility and evolve its design. For example, it quickly became clear that the single attribute of success vs. failure needed to be qualified by another attribute that indicates who the work is intended for, called "Target Segment." Work that is intended to support the internal operations of an *Internet Service Provider* (ISP) is not necessarily visible to the billions of Internet users and will, at best, be part of only a few thousand organizations. In terms of Internet scale, that is considered minuscule. However wide adoption of a tool among ISPs can have substantial benefit, and thereby qualify as "massive adoption."

The wiki can serve both as a means of recording the IETF's track record of successes and failures, as well as providing a means of encouraging community dialogue about the quality of different IETF efforts. In addition, it can provide a window onto completed IETF work for the broader Internet community.

[1] D. Thaler and B. Aboba, "What Makes for a Successful Protocol?" RFC 5218, July 2008.

[2] `http://trac.tools.ietf.org/misc/outcomes/`

Final Phase of Four-byte AS Number Policy Begins in APNIC Region

From 1 January 2010, the *Asia Pacific Network Information Centre* (APNIC) ceased to make a distinction between four-byte only and two-byte only *Autonomous System* (AS) numbers. Instead, all AS numbers are now considered to be four-byte AS numbers.

This change marks the third phase of the transition to four-byte AS numbers. For more information on the implementation phases of the four-byte AS number policy, please see "Policies for Autonomous System number management in the Asia Pacific region," section 6.3, "Timetable for moving from two-byte only AS numbers to four-byte AS numbers," available from:
`http://www.apnic.net/policy/asn-policy.html#6.3`

To learn more about how the transition to four-byte AS numbers may affect your network, see: `http://icons.apnic.net/asn`

Charting the Course for Future Internet Leaders

As the importance of the Internet grows in all aspects of modern life, so too do the challenges of those in positions of leadership and responsibility.

Responding to the need for well-qualified leadership, the *Internet Society* (ISOC) is now accepting applications from people seeking to join the new generation of Internet leaders to address the critical technology, policy, business, and education challenges that lie ahead.

Successful candidates in ISOC's *Next Generation Leaders Program* will gain a wide range of skills in a variety of disciplines, as well as the ability and experience to work with people at all levels of society.

This program, under the patronage of the European Commission, blends course work and practical experience to help prepare young professionals (aged from 20 to 40) from around the world to become the next generation of Internet technology, policy, and business leaders.

"The Internet Society's Next Generation Leaders Program is a unique opportunity to identify potential Internet leaders and help them accelerate their careers," said Bill Graham, responsible for strategic global engagement at ISOC.

The key to the Internet's success lies in the Internet Model of decentralized architecture and distributed responsibility for development, operation, and management. That model also creates important leadership opportunities, especially in those spaces where technology, policy, and business intersect.

"We have designed the Next Generation Leaders Program to prepare young professionals for leadership, bridging the boundaries between business, technical development, policy, and governance on local, regional, and international levels," said Graham.

Full details of the Next Generation Leaders Program are available at:
`http://www.isoc.org/leaders/`

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

June 2010 Volume 13, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: `www.cisco.com/ipj`

ISSN 1944-1134

From The Editor

Protocol changes are never easy, especially when they involve something as fundamental as the *Internet Protocol* (IP). This journal has published numerous articles about the depletion of IPv4 addresses and several articles about IPv6, including methods for a gradual transition from v4 to v6. A lot of energy has gone into the development, promotion, and deployment of IPv6, but in reality only a small fraction of the global Internet currently supports IPv6. Meanwhile, the *Internet Assigned Numbers Authority* (IANA) and the *Regional Internet Registries* (RIRs) will "soon" (12 to 24 months from now is predicted) run out of IPv4 addresses to allocate. Although this situation has some serious implications for new entrants to the *Internet Service Provider* (ISP) market, it does not spell the end of the Internet as we know it. Numerous *Network Address Translation* (NAT) solutions are already widely deployed, and the IETF is discussing other solutions. One example is *Address Sharing* as explained by Geoff Huston in our first article.

Changes to the *Domain Name System* (DNS) are also underway. The *Domain Name System Security Extensions* (DNSSEC) are being gradually deployed in the global Internet. As with any complex technology, implementation of DNSSEC is not without problems. Our second article, by Torbjörn Eklöv and Stephan Lagerholm, is a step-by-step guide for those considering implementing DNSSEC in their network.

By now you will be aware that we have implemented a renewal system for subscribers and will not be automatically extending your subscription unless you contact us via e-mail or use the online tool to renew your subscription. You can find your subscription ID and expiration date either on the back page of your copy or on the envelope that it came in. In order to access your record, click the "Subscriber Services" link on our webpage at `www.cisco.com/ipj`, and enter your e-mail address and the subscription ID. The system will send you a link that allows direct access to your record, and you will be able to update your address and renew your subscription. If you no longer have access to the e-mail you used when you subscribed, or have forgotten your subscription ID, just send a message to `ipj@cisco.com` and we will make the necessary changes for you.

—Ole J. Jacobsen, Editor and Publisher
`ole@cisco.com`

NAT++: Address Sharing in IPv4

by Geoff Huston, APNIC

In this article I examine the topic that was discussed in a session at the 74th meeting of the *Internet Engineering Task Force* (IETF) in March 2009, about *Address Sharing* (the SHARA BOF)[0], and look at the evolution of *Network Address Translation* (NAT) architectures in the face of the forthcoming depletion of the unallocated IPv4 address pool.

Within the next couple of years we will run out of the current supply of IPv4 addresses. As of the time of writing this article, the projected date when the *Internet Assigned Numbers Authority* (IANA) pool will be depleted is August 3, 2011, and the first *Regional Internet Registry* (RIR) will deplete its address pool about March 20, 2012.

Irrespective of the precise date of depletion, the current prediction is that the consumption rate of addresses at the time when the free pool of addresses is exhausted will probably be running at some 220 million addresses per year, indicating a deployment rate of some 170–200 million new services per year using IPv4. The implication is that the Internet will exhaust its address pool while operating its growth engines at full speed.

How quickly will IPv6 come to the rescue? Even the most optimistic forecast of IPv6 uptake for the global Internet is measured in years rather than months following exhaustion, and the more pessimistic forecasts extend into multiple decades.

For one such analysis using mathematical modelling techniques, refer to Jean Camp's work[1]. One of the conclusions from that 2008 study follows: "There is no feasible path which results in less than years of IPv4/IPv6 co-existence. Decades is not unreasonable."

The implication of this conclusion is that we will need to operate a dual-stack Internet for many years to come, and the associated implication is that we will have to make the existing IPv4 Internet span a billion or more new deployed services—and do so with no additional address space.

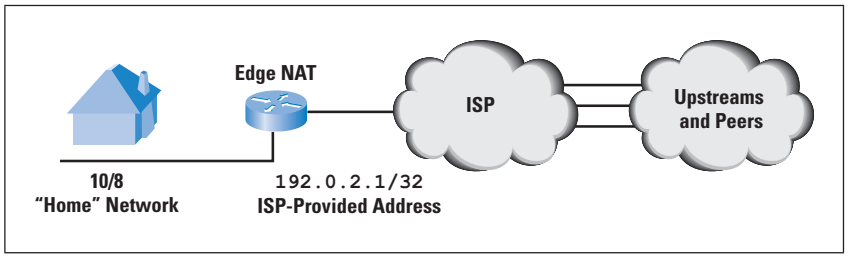
So how are we going to make the IPv4 address pool stretch across an ever-larger Internet?

Given that the tool chest we have today is the only one available, there appears to be only one answer to this question: Use *Network Address Translators,* or NATs.

For a description of how NATs work and some of the terminology used to describe NAT behavior, refer to the article "Anatomy: A Look Inside Network Address Translators," published in this journal[2].

Today NATs are predominately edge devices that are bundled with DSL modems for residential access, or bundled with routing and security firewall equipment for small to midsize enterprise use as an edge device. The generic model of NAT deployment currently is a small-scale edge device that generally has a single external-side public IP address and an internal-side private IP network address (often network 10). The NAT performs address and port translation to map all currently active sessions from the internal addresses to ports on the public IP address. This NAT deployment assumes that each edge customer has the unique use of a public IP address (refer to Figure 1).

Figure 1: Conventional NAT Deployment



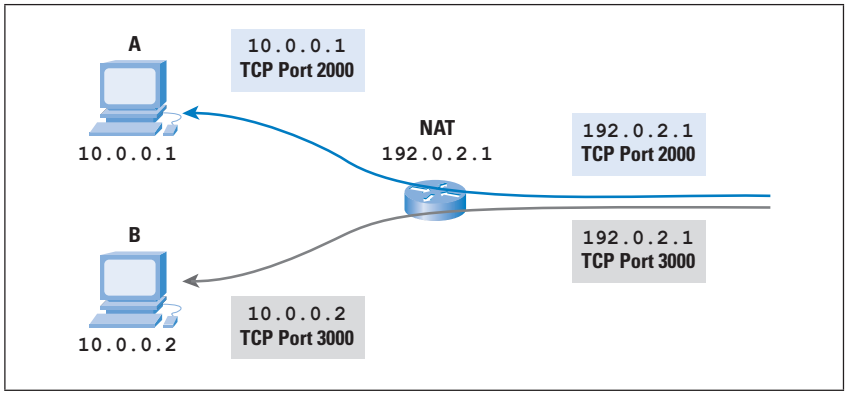
The question provoked by IPv4 address exhaustion is what happens when there are no longer sufficient IPv4 addresses to provide this 1:1 mapping between customers and public IPv4 addresses? In other words, what happens when there are simply not enough IPv4 addresses to allow all customers to have exclusive use of their own unique IPv4 address?

This question has only two possible answers. One is for no one to use IPv4 addresses at all, on the basis that the entire Internet has migrated to use IPv6. But this answer appears to be an uncomfortable number of decades away, so we need to examine the other answer: If there are not enough addresses to go around, then we will have to *share* them.

But isn't sharing IP addresses impossible in the Internet architecture? The IP address in a packet header determines the destination of the packet. If two or more endpoints share the same address, then how will the network figure out which packets go to which endpoint? It is here that NATs and the transport layer protocols, the *Transmission Control Protocol* (TCP) and the *User Datagram Protocol* (UDP), come together. The approach is to use the *port address* in the TCP and UDP header as the distinguishing element.

For example, in Figure 2, incoming TCP packets with TCP port address 2000 may need to be directed to endpoint A, while incoming TCP packets with TCP port address 3000 need to be directed to endpoint B. The incoming TCP packets with a port address of 2000 are translated to have the private IP address of endpoint A, and incoming TCP packets with a port address of 3000 are translated to have the private address of endpoint B.

Figure 2: Address Sharing with NATs



As long as you restrict yourself to applications that use TCP or UDP, you don't rely on receiving *Internet Control Message Protocol* (ICMP) packets, and you don't use applications that contain IP addresses in their payload, then you might expect this arrangement to function.

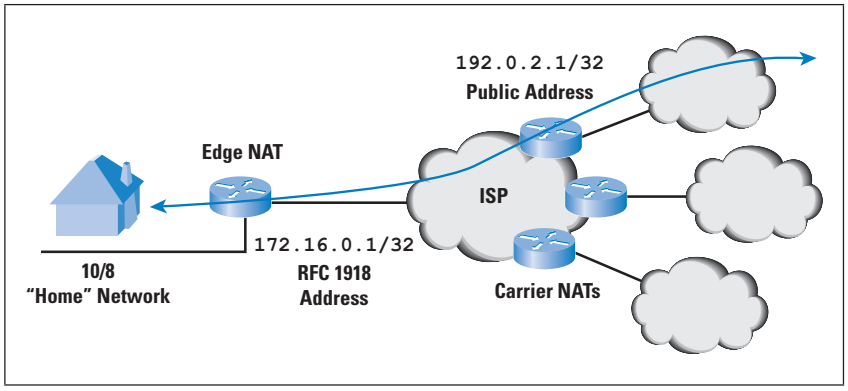
ICMP is a problem because the ICMP packet does not contain a TCP or UDP transport layer. All that a NAT sees in the ICMP packet is its own external address as the destination IP address. To successfully deliver an ICMP packet through a NAT, the NAT needs to perform a more complex function that uses the ICMP-encapsulated IP header to select the original outbound combined IP + TCP header or IP + UDP header in the ICMP payload. The source IP address and transport protocol port address in the ICMP payload are then used to perform a lookup into the NAT binding table and then perform two mappings: one on the ICMP header to map the destination IP address to the internal IP address, and the second on the payload header where the source IP address and port number are changed to the interior-side values, and the checksums altered as appropriate. Now in most cases ICMP really is not critical, and a conservative NAT implementation may elect to avoid all that packet inspection and simply discard all incoming ICMP messages, but one message that is important is the ICMP *packet-too-large-and-fragmentation-disabled* message used in IPv4 *Path MTU Discovery*[3].

Sharing IP addresses is fine in theory, but how can we achieve it in practice? How can many customers, already using NATs, share a single public IP address?

Carrier-Grade NATs

One possible response is to add a further NAT into the path. In theory the *Internet Service Provider* (ISP) could add NATs on all upstream and peer connections, and perform an additional NAT operation as traffic enters and leaves the ISP's network. Variations of this approach are possible, placing the ISP NATs at customer aggregation points within the ISP's network, but the principle of operation of the ISP NAT is much the same.

Figure 3: Carrier NATs



The edge NATs translate between private address pools at each customer's site and an external address provided by the ISP, so nothing has changed there. The change in this model is that the ISP places a further NAT in the path within the ISP network, so that a set of customers is then sitting behind a larger NAT inside the ISP's network, as shown in Figure 3.

This scenario implies that the external address that the ISP provides to the customer is actually yet another private address, and the ISP's NAT performs yet another transform to a public address in this second NAT. In theory this NAT is just a larger version of an existing NAT with larger NAT binding space, higher packet-processing throughputs, and a comprehensive specification of NAT binding behavior. In practice it may be a little more complicated because at the network edge the packet rates are well within the processing capability of commodity processors, whereas in the core of the network there is an expectation of higher levels of robust performance from such units. Because it is intended that such a NAT handle thousands of customers and large numbers of simultaneous data flows and peak packet rates, it requires a performance level well beyond what is seen at the customer edge and, accordingly, such a NAT has been termed a *Carrier-Grade NAT* (CGN), or a *Large-Scale NAT* (LSN).

From the inside of the two NATs, not much has changed with the addition of the CGN in terms of application behavior. It still requires an outbound packet to trigger a binding that allows a return packet through to the internal destination, so nothing has changed there. Other aspects of NAT behavior, notably the NAT binding lifetime and the form of *Cone Behavior* for UDP, take on the more restrictive of the two NATs in sequence. The binding times are potentially problematic in that the two NATs are not synchronized in terms of binding behavior. If the CGN has a shorter binding time, it is possible for the CGN to misdirect packets and cause application-level problems. However, this situation is not overly different from a single-level NAT environment where aggressively short NAT binding times also run the risk of causing application-level problems when the NAT drops the binding for an active session that has been quiet for an extended period of time.

However, one major assumption is broken in this structure, namely that an IP address is associated with a single customer. In this model a single public IP address may be used simultaneously by many customers at once, albeit on different port numbers. This scenario has obvious implications in terms of some current practices in filters, firewalls, "black" and "white" lists, and some forms of application-level security and credentials where the application makes an inference about the identity and associate level of trust in the remote party based on the remote party's IP address.

This approach is not without its potential operational problems as well. For the ISP, service resiliency becomes a critical concern in so far as moving traffic from one NAT-connected external service to another will cause all the current sessions to be dropped, unless the internal ISP network architecture uses a transit access network between the CGNs and the external transit providers. Another concern is one of resource management in the face of potentially hostile applications. For example, an end host infected with a virus may generate a large amount of probe packets to a large range of addresses. In the case of a single edge NAT, the large volumes of bindings generated by this behavior become a local resource management problem because the customer's network is the only affected site. In the case where a CGN is deployed, the same behavior starts to consume binding space on the CGN and, potentially, can starve the CGN of external address bindings. If this problem is seen to be significant, the CGN would need to have some form of external address rationing per internal client in order to ensure that the entire external address pool is not consumed by a single errant customer application. This "rationing" would have the unwanted effect of forcing the ISP to deny access to its customers.

The other concern here is one of scalability. Although the greatest leverage of the CGN in terms of efficiency of usage of external addresses occurs when the greatest numbers of internal edge-NAT-translated clients are connected, there are some real limitations in terms of NAT performance and address availability when an ISP wants to apply this approach to networks where the customer population is in the millions or larger. In this case the ISP is required to use an IPv4 private address pool to number every client. But if all customers use network 10 as their "internal" network, then what address pool can the ISP use for its private address space? One of the few answers that come to mind is to deliberately partition the network into numerous discrete networks, each of which can be privately numbered from the smaller private address pool of `172.16.0.0/12`, allowing for some 600,000 or so customers per network partition, and then use a transit network to "glue" together the partitioned elements, as shown in Figure 4.

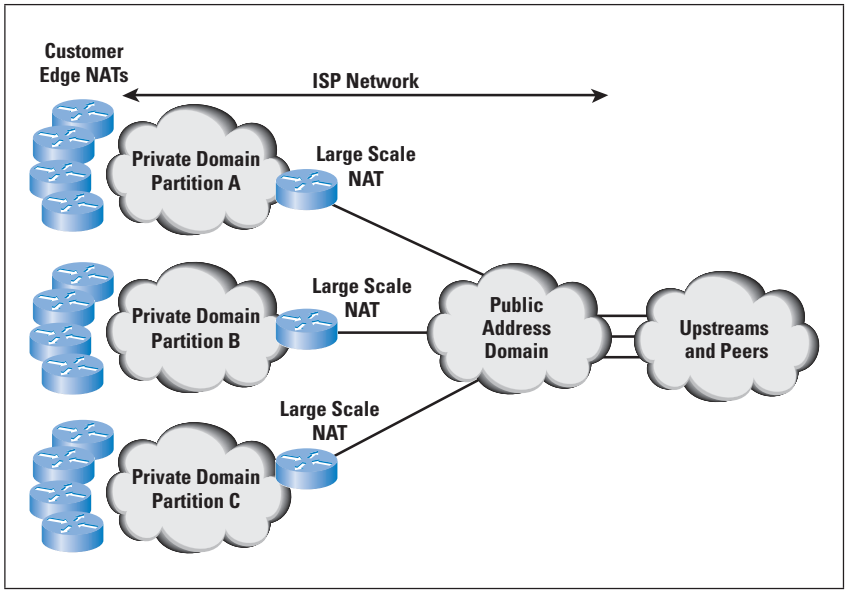


Figure 4: Multiple Carrier NAT Deployment Using Network Partitioning

The advantage of the CGN approach is that for the customer nothing changes. Customers do not need to upgrade their NAT equipment or change them in any way, and for many service providers this motivation is probably sufficient to choose this path. The disadvantages of this approach lie in the scaling properties when looking at very large deployments, and the problems of application-level translation, where the NAT attempts to be "helpful" by performing deep packet inspection and rewriting what it thinks are IP addresses found in packet payloads. Having one NAT do this rewriting is bad enough, but loading them up in sequence is a recipe for trouble!

Are there alternatives?

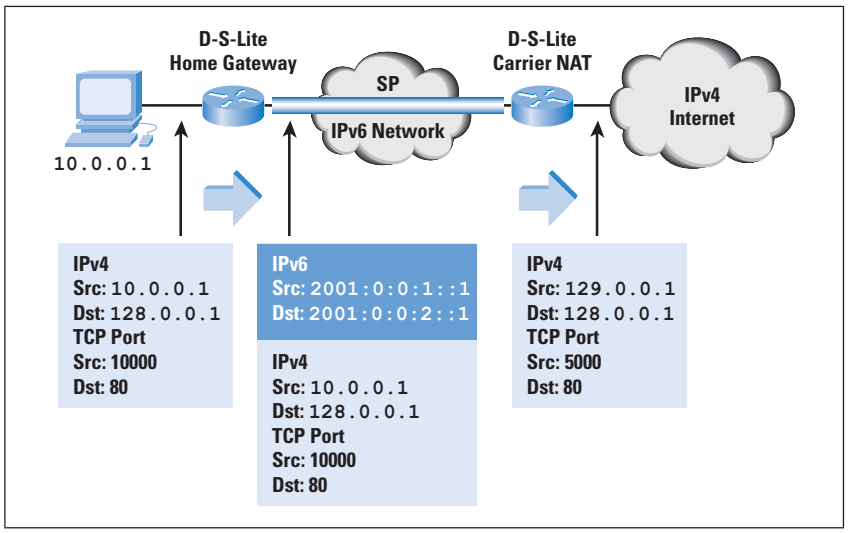
Dual-Stack Lite and Carrier-Grade NATs

One rather elegant alternative is described by Alain Durand and others in an Internet Draft "Dual-stack lite broadband deployments post IPv4 exhaustion"[4]. The assumption behind this approach is that the ISP's network infrastructure needs to support IPv6 running in native mode in any case, so is there a way in which the ISP can continue to support IPv4 customers without running IPv4 internally?

Here the customer NAT is effectively replaced by a tunnel ingress/egress function in the *Dual-Stack Lite Home Gateway*. Outgoing IPv4 packets are not translated, but are encapsulated in an IPv6 packet header, where the IPv6 packet header contains a source address of the carrier side of the home gateway unit and a destination address of the ISP's gateway unit. From the ISP's perspective, each customer is no longer uniquely addressed with an IPv4 address, but instead is addressed with a unique IPv6 address. The customer's interface to the ISP network, the Home Gateway, is configured with this IPv6 address as the customer end of the IPv4-in-IPv6 tunnel, where the other end of the tunnel is the IPv6 address of the ISP's Dual-Stack Lite Gateway unit.

The service provider's Dual-Stack Lite gateway unit performs the IPv6 tunnel termination and a NAT translation using an extended local binding table. The "interior" NAT address is now a 4-tuple of the IPv4 source address, protocol ID, and port, plus the IPv6 address of the home gateway unit, while the external address remains the triplet of the public IPv4 address, protocol ID, and port. In this way the NAT binding table contains a mapping between interior "addresses" that consist of IPv4 address and port plus a tunnel identifier and public IPv4 exterior addresses. This way the NAT can handle a multitude of network 10 addresses, because the addresses can be distinguished by different tunnel identifiers. The resultant output packet following the stripping of the IPv6 encapsulation and the application of the NAT function is an IPv4 packet with public source and destination addresses. Incoming IPv4 packets are similarly transformed, where the IPv4 packet header is used to perform a lookup in the Dual-Stack Lite gateway unit, and the resultant 4-tuple is used to create the NAT-translated IPv4 packet header plus the destination address of the IPv6 encapsulation header (refer to Figure 5).

Figure 5: Dual-Stack Lite



The advantage of this approach is that now only a single NAT is needed in the end-to-end path because the functions of the customer NAT are now subsumed by the carrier NAT. This scenario has some advantages in terms of those messy "value-added" NAT functions that attempt to perform deep packet inspection and rewrite IP addresses found in data payloads. There is also no need to provide each customer with a unique IPv4 address, public or private, so the scaling limitations of the dual-NAT approach are also eliminated. The disadvantages of this approach lie in the need to use a different *Customer Premises Equipment* (CPE) device, or at least one that is reprogrammed. The device now requires an external IPv6 interface and at a minimum an IPv4 or IPv6 tunnel gateway function. The device can also include a NAT if desired, but it is not required in terms of the basic Dual-Stack Lite architecture.

This approach pushes the translation into the middle of the network, where the greatest benefit can be derived from port multiplexing, but it also creates a critical hotspot for the service itself. If the carrier NAT fails in any way, the entire customer base is disrupted. It seems somewhat counter intuitive to create a resilient network with stateless switching environments and then place a critical stateful unit in the middle! So is there an approach that can push this translation back to the edges while avoiding a second NAT in the carrier's network?

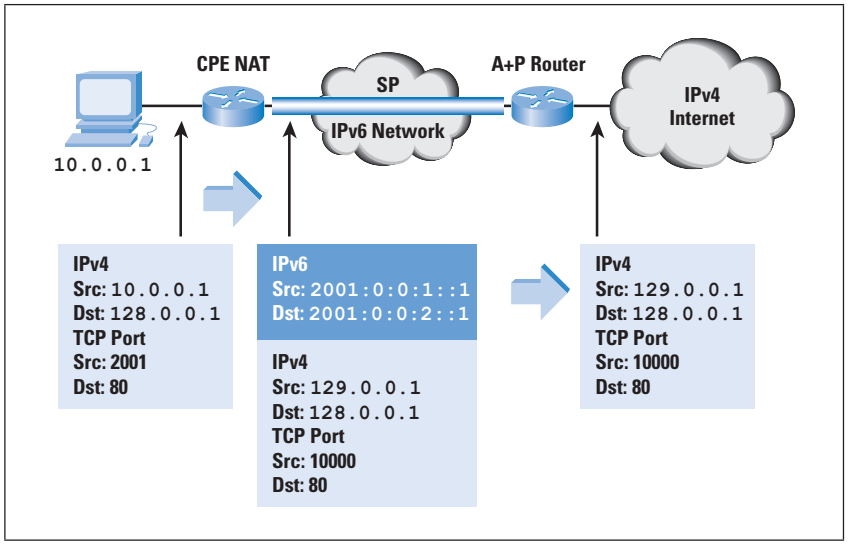
The Address Plus Port Approach

The observation here is that CPE NATs currently map connections into the 16-bit port field of the single external address. If the CPE NAT could be coerced into performing this mapping into 15 bits of the port field, then the external address could be shared between two edge CPE devices, with the leading bit of the port field denoting which CPE device. Obviously, moving the bit marker across the port field would allow more CPE devices to share the one address, but it would reduce the number of available ports for each CPE device in the process.

The theory is again quite simple. The CPE NAT is dynamically configured with an external address, as happens today, and a port range, which is the additional constraint. The CPE NAT performs the same function as before, but it is now limited in terms of the external ports it can use in its NAT bindings to those that lie within the provided port range, because some other CPE may be concurrently using the same external IP address with a different port range.

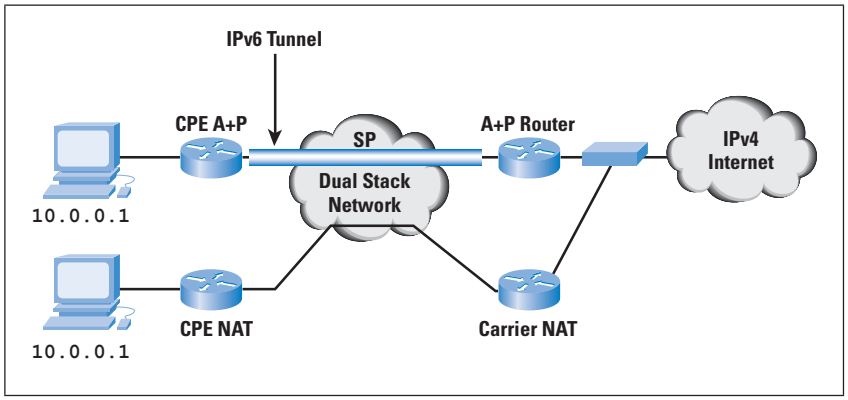
For outgoing packets this limitation implies only a minor change to the network architecture, in that the RADIUS[9] exchange to configure the CPE now must also provide a port range to the CPE device. However, the case of incoming packets is more challenging. Here the ISP must forward the packet based not only on the destination IP address, but also on the port value in the TCP or UDP header. A convenient way to forward the packet is to take the Dual-Stack Lite approach and use an IPv4-in-IPv6 tunnel between the CPE and the external gateway (Figure 6). This gateway, or *Address Plus Port* (A + P) router, needs to be able to associate each address and port range with the IPv6 address of a CPE device, which it can learn dynamically as it decapsulates outgoing packets. Corresponding incoming packets are encapsulated in IPv6 using the IPv6 destination address that it has learned previously. In this manner the NAT function is performed at the edge, much as it is today, and the interior device is a more conventional form of tunnel server.

Figure 6: Address Plus Port Framework



This approach relies on every CPE device being able to operate using a restricted port range, to perform IPv4-in-IPv6 tunnel ingress/egress functions, and to act as an IPv6 provisioned endpoint for the ISP network, which is perhaps an unrealistic hope. Further modifications to this model (Figure 7) propose the use of an accompanying CGN operated by the ISP to handle those CPE devices that cannot support these Address Plus Port functions.

Figure 7: Combined Address Plus Port and Carrier Grade NAT



If the port range assigned to the CPE is from a contiguous range of port values, then this approach could exacerbate some known problems with infrastructure protocols. There are *Domain Name System* (DNS) problems with guessable responses. The so-called "Kaminsky Attack" on the DNS[5, 6] is one such example where the attack can be deflected, to some extent, by using a randomly selected port number for each DNS query. Restricting the port range could mitigate the efficacy of such measures under certain conditions.

However, despite such concerns, the approach has some positive aspects. Pushing the NAT function to the edge has some considerable advantage over the approach of moving the NAT to the interior of the network.

The packet rates are lower at the edge, allowing for commodity computing to process the NAT functions across the offered packet load without undue stress. The ability for an end-user's application to request a particular NAT binding behavior by speaking directly with the local NAT using the *Internet Gateway Device Protocol,* as part of the *Universal Plug and Play* (UPnP)[7] framework, will still function in an environment of edge NATs operating with restricted port ranges. Aside from the initial provisioning process to equip the CPE NAT with a port range, the CPE, and the edge environment is largely the same as in today's CPE NAT model.

That is not to say that this approach is without its negative aspects, and it is unclear as to whether the perceived benefits of a "local" NAT function outweigh the problems associated with this model of address sharing. The concept of port "rationing" is a very suboptimal means of address sharing, given that after a CPE device has been assigned a port range those port addresses are unusable by any other CPE. The prudent ISP would assign to each CPE device a port address pool equal to some estimate of peak demand, so that, for example, each CPE device would be assigned 1,000 ports, allowing a single external IP address to be shared across only 60 such CPE clients. Neither the Carrier-Grade NAT approach nor the Dual-Stack Lite approach attempts this form of rationed allocation, allowing the port address pool to be treated as a common resource, with far higher levels of usage efficiency through dynamic management of the port pool.

The difference here is that in the dynamically managed approach any client can use the currently unused port addresses, whereas in the rationed approach each client has access to a fixed pool of port addresses that cannot be shared with any other client—even when the client does not need them. The difference here parallels the difference in network efficiency between time-division multiplexed synchronous circuits and asynchronous packets at Layer 2 in the network model. In the Address Plus Port framework the leverage obtained in terms of making efficient use of coopting these additional 16 bits of port address into the role of additional bits of client identifier address space is reduced by the imposition of a fixed boundary between customer and ISP use in the port address plan. The central NAT model of a CGN effectively pools the port address range and facilitates far more efficient sharing of this common port address pool across a larger client base.

Alain Durand reported to IETF 74 on a data-collection experiment using a *Cable Modem Termination System* (CMTS) with 8,000 subscribers where the peak port consumption level was 40,000 ports, or a maximum average port consumption of 5 ports per subscriber in each direction. As Alain noted, this average value needs to be compared with the hundreds of ports consumed by a single client browsing a Web 2.0 or *Asynchronous Java and XML* (AJAX) site, but he also noted that a central model of port sharing does yield far higher levels of address-sharing efficiency than the Address Plus Port advanced allocation model.[8]

The other consideration here is that this approach constitutes a higher overhead for the ISP, in that the ISP must support both "conventional" CPE and Address Plus Port equipment. In other words, the ISP must deploy a CGN and support customer CPE using a two-level NAT environment in addition to operating the Address Plus Port infrastructure. Unless customers would be willing to pay a significant price premium for such an Address Plus Port service, it is unlikely that this option would be attractive for the ISP as an additional cost after the CGN cost.

General Considerations with Address Sharing

The basic elements of any such approach to address sharing involve the CPE equipment at the edge, optionally some form of tunneling of traffic between the CPE and the carrier equipment, and carrier-provided equipment at the edge of the carrier's network (refer to Figure 8).

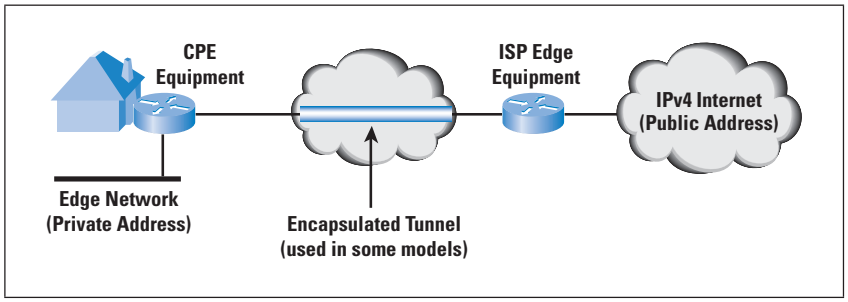


Figure 8: Generic Architecture for Address Sharing

A variety of technical solutions here involve these basic building blocks, so it is not true to say that this challenge is technically significant. But few ISPs have decided to proceed with large-scale deployment of any form of address-sharing technology for their IPv4 network infrastructure. So what is the problem here?

I suspect that the real concern is the consideration of the relevant business model that would guide this deployment. Today's Internet is large. It encompasses some 1.7 billion human users, a larger pool of devices, and hundreds of millions of individual points of control. If we want to change this deployed system, we will need copious quantities of money, time, and unity of purpose. So do we have money, time, and unity of purpose?

Money is missing: It could be argued that we have left the entire IPv6 transition effort to this late stage because of a lack of money. The main advantage of the Internet was that it was cheap. Packet sharing is intrinsically more efficient than circuit sharing, and the shift in functions of network service management from the network to the customer-owned and -operated endpoints implied further cost savings for the network operator. So the Internet model gained ascendency because for consumers it represented a cost-effective choice. It was cheap.

But what does IPv6 offer consumers? For existing Internet consumers it appears that IPv6 does not offer anything that they don't already have with IPv4—it offers mail, the web, various forms of voice services, and games. So consumers are not exactly motivated to pay more for the same services they already enjoy today.

In addition, it would appear that the ISP must carry this cost without incremental revenue from its customer base. But the ISP industry has managed to shave most of its revenue margins in a highly competitive industry, and at the same time lose control of services, their delivery, and their potentially lucrative revenue margins. Thus the ISP industry has been collectively idle in this area not because it cannot see the problem in terms of the imminent exhaustion of IPv4, but because it has little choice because of financial constraints that have prevented it from making the necessary longer-term investments in IPv6. So if the ISP industry has been unwilling to invest in IPv6 so far, then what incentive is there for it to invest in IPv6 and at the same time also invest in these IPv4 address-sharing measures? Is the lure of new, low-margin customers sufficient incentive to make such investments in this carrier-grade equipment? Or is the business case still insufficiently attractive?

Time is missing: The unallocated IPv4 address pool is already visibly waning. Without any form of last-minute rush, the pool will be around for the next 2 years, or until 2012 or so. But with any form of typical last-minute rush, this pool could be depleted in the coming months rather than in the coming years. Can we do what we need to do to get any of these approaches to a state of mass-market deployment in the next few months? All these approaches appear to be at the early stages of a timeline that starts with research and then moves on to development, prototyping, and trials; then to standards activity and industry engagement to orchestrate supply lines for end user equipment, ISP equipment, and definition of operational practices; then to product and service development; and finally, to deployment. For an industry that is the size of the Internet, "technical agility" is now an obsolete historical term. Even with money and unity of purpose this process will take some years, and without money—or even the lure of money—it becomes a far more protracted process, as we have seen already with IPv6 deployment.

And do we have *unity of purpose* here? Do we agree on an approach to address sharing that will allow players to perform their tasks? That will allow consumer product vendors to develop the appropriate product? That will allow application developers to develop applications that will operate successfully in this environment? That will allow the end user platform vendors to incorporate the appropriate functions in the operating system stacks? That will allow ISPs to integrate vendors' productions into their operational environments? Right now it is pretty clear that what we have is a set of ideas, each of which has relative merits and disadvantages, and no real unity of purpose.

It is easy to be pessimistic at this stage, given that the real concerns here appear to be related more to the factors associated with a very large industry attempting to respond to a very challenging change in the environment in which it operates. The question here is not really whether Address Plus Port routing is technically inferior to Dual-Stack Lite, or whether Carrier-Grade NATs are technically better or worse than either of these approaches. The question here is whether this industry as a whole will be able to sustain its momentum and growth across this hiatus. And, from this perspective, I believe that such pessimism about the future of the Internet is unwarranted.

The communications industry has undergone significant technological changes over the years, and this change is one more in the sequence. Some of these transformations have been radical in their effect, such as the introduction of the telephone in the late nineteenth century, whereas others have been more subtle, such as in the introduction of digital technology to telephony in the latter part of twentieth century, replacing the earlier analogue circuit model of telephony carriage. Some changes have been associated with high levels of risk, and we have seen a myriad of smaller, more agile players enter the market to lead the change while the more risk-averse enterprises stand back. On the other hand, other changes require the leverage of economies of scale, and we have seen market consolidation behind a smaller number of highly capitalized players.

My personal opinion is that the Dual-Stack Lite approach is the best one, because it appears to be technically elegant. I suspect, however, that the lowest-common-denominator fall-back position that this somewhat conservative industry will adopt will rely strongly on Carrier-Grade NATs, and the industry is likely to eschew the more complex support mechanisms required by the various permutations of Address Plus Port routing.

Further Reading

[0] The Address Sharing BOF was held at IETF 74 in March 2009. The presentations and a summary of the session can be found as part of the proceedings of that meeting:
http://www.ietf.org/proceedings/09mar/shara.html

[1] **http://www.ripe.net/ripe/meetings/ripe-56/presentations/Camp-IPv6_Economics_Security.pdf**

[2] Geoff Huston, "Anatomy: A Look Inside Network Address Translators," *The Internet Protocol Journal,* Volume 7, No. 3, September 2004.

[3] Jeff Mogul and Steve Deering, "Path MTU Discovery," RFC 1191, November 1990.

[4] **draft-ietf-softwire-dual-stack-lite-00.txt**

[5] **http://www.doxpara.com/DMK_BO2K8.ppt**

[6] **http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html**

[7] http://en.wikipedia.org/wiki/Universal_Plug_and_Play

[8] http://www.ietf.org/proceedings/09mar/slides/shara-8/shara-8.htm

[9] C. Rigney, S. Willens, A. Rubens, W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865, June 2000.

[10] Egevang, K., and P. Francis, "The IP Network Address Translator (NAT)," RFC 1631, May 1994.

[11] Srisuresh, P., and D. Gan, "Load Sharing Using IP Network Address Translation (LSNAT)," RFC 2391, August 1998.

[12] Srisuresh, P., and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," RFC 2663, August 1999.

[13] Tsirtsis, G., and P. Srisuresh, "Network Address Translation—Protocol Translation (NAT-PT)," RFC 2776, February 2000.

[14] Hain, T., "Architectural Implications of NAT," RFC 2993, November 2000.

[15] Srisuresh, P., and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," RFC 3022, January 2001.

[16] Holdrege, M., and P. Srisuresh, "Protocol Complications with the IP Network Address Translator," RFC 3027, January 2001.

[17] D. Senie, "Network Address Translator (NAT)-Friendly Application Design Guidelines," RFC 3235, January 2002.

[18] Srisuresh, P., J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan, "Middlebox Communication Architecture and Framework," RFC 3303, August 2002.

[19] Daigle, L., and IAB, "IAB Considerations for Unilateral Self-Address Fixing (UNSAF) Across Network Address Translation," RFC 3424, November 2002.

[20] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN—Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)," RFC 3489, March 2003.

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. The author of numerous Internet-related books, he is currently the Chief Scientist at APNIC. He was a member of the Internet Architecture Board (IAB) from 1999 until 2005, and served on the Board of the Internet Society from 1992 until 2001.
E-mail: gih@apnic.net

Operational Challenges When Implementing DNSSEC

by Torbjörn Eklöv, Interlan Gefle AB, and Stephan Lagerholm, Secure64 Software Corp.

As a reader of *The Internet Protocol Journal*, you are probably familiar with the *Domain Name System* (DNS) "cache poisoning" techniques discovered a few years ago. And you have most likely heard that *Domain Name System Security Extensions* (DNSSEC)[0, 13, 14, 15] is the long-term cure. But you might not know exactly what challenges are involved with DNSSEC and what experience the early adopters have gathered and documented. Perhaps you waited with your own rollout until you could gather more documentation about operational experiences when rolling out DNSSEC.

Stephan Lagerholm and Torbjörn Eklöv are DNS architects with significant DNSSEC experience. Torbjörn lives in Sweden and has helped several municipalities, as well as other organizations, sign their zones. Stephan Lagerholm lives in Dallas, Texas, and has been involved in implementing DNSSEC at several U.S. federal agencies. This article summarizes their experiences, including lessons learned from implementing the technology in production environments, and discusses associated operational concerns.

Background

A plethora of information about DNSSEC and cache poisoning attacks is available on the Internet[16], so we will not repeat it, but we think it is important to state where DNSSEC is today.

During the last few years the number of deployments, as well as the size and importance of the signed domains, has increased significantly. One of the main reasons for adoption of the DNSSEC during the past year was that the U.S. *Office of Management and Budget* (OMB) issued a mandate requiring the signing of the **.gov** domain in the beginning of the year. U.S. federal agencies were mandated to sign their domains by the end of 2009. Some agencies have already implemented the technology, whereas others are still working on it.[1]

Acceptance of DNSSEC technology is also reaching outside of the U.S. government. *Top Level Domains* (TLDs) around the globe have announced DNSSEC initiatives. To mention a few, Afilias signed .org and Neustar recently announced signing of **.us**. Several *County Code TLDs* (ccTLDs), including **.nl** and **.de**, announced that DNSSEC implementation is a work in progress. VeriSign has announced that it is working on signing the largest TLDs, namely **.com** and **.net**. Finally, the *Internet Corporation for Assigned Names and Numbers* (ICANN) along with VeriSign released a timeline for signing the root zone. And of course, the pioneer **.se** is on its fourth year as a signed TLD.

Several vendors have released software and products to support and make the signing of zones easier. A range of different products is now available on the market.

DNS professionals now have a broad choice of technology—from collections of open-source signing scripts to advanced systems with full automation and support for *Federal Information Processing Standard* (FIPS)-certified cryptography.

Operational Challenges

DNSSEC might significantly affect operations unless it is carefully implemented because it requires some changes to the underlying DNS protocol. Those changes are, in fact, the first significant changes that have been made to the DNS protocol since it was invented. Those changes might sometimes fool old systems into believing that the packets are illegal. DNSSEC also introduces new operational tasks such as rolling the keys and resigning the zone. Such tasks must be performed at regular intervals. Furthermore, as with any new technology, there are misconceptions about how to interpret the RFC standard.

The First Bug Reported

Late summer 2007, Torbjörn Eklöv convinced the municipality of Gävle in Sweden of the benefits of DNSSEC. He proudly signed what is believed to be the first municipality zone in the world, **`gavle.se`**. At first, everything worked fine. A week or so later, Gävle received reports from citizens who could not reach the municipality's websites. It turned out that a new version of *Berkeley Internet Name Domain* (BIND) was rolled out by a large service provider and that this version of BIND introduced a rather odd bug that affected DNSSEC. The result of the bug was that home users with some home routers and firewalls could not reach any signed domains.

Some people who heard about the problem at **`gavle.se`** wrongly believed that DNSSEC caused the problem and that DNSSEC is broken. However, this assumption is not true; DNSSEC worked as expected, but a bug in a particular version of BIND caused the problem. The problem triggered some research on how home routers handle DNSSEC. *Stiftelsen för Internetinfrastruktur,* the organization that runs the **`.se`** TLD, issued a report describing how commonly used home routers and firewalls handled the new protocol changes in DNS[2]. Later, Nominet, which administers the **`.uk`** TLD, issued a similar report[3]. In addition, DENIC, which administers the **`.de`** TLD, researched the same subject[4]. The results are all discouraging; only 9 out of 38 tested home gateways supported DNSSEC correctly in the most recent reports.

A *Birds of a Feather* (BoF) session was held at the 76th meeting of the *Internet Engineering Task Force* (IETF) in Hiroshima to discuss the problems involving home gateways[5]. We look forward to seeing progress in this area.

Preparing Your Firewall for DNSSEC

Most problems with DNSSEC are related to firewalls. Make sure to involve your security and networking administrators so that they can make the required changes before taking DNSSEC into production.

Two types of firewall problems are most common:

The first involves the *Transmission Control Protocol* (TCP). There is a misconception among firewall vendors and security administrators that DNS queries use the *User Datagram Protocol* (UDP) and that zone transfers use TCP. Unfortunately, this assumption is not entirely true. DNS queries first try UDP, but revert to TCP if no response is received for the initial UDP query or if the response lacks important information because it is truncated. The possibility of something in the path blocking the response to the initial query is much higher with DNSSEC because of the increased size of the responses.

> *For DNSSEC to work correctly, it is mandatory that you open your firewall for both TCP and UDP over port 53.*

The second problem is related to the *IP Buffer Reassembly* size. The authors of the DNSSEC standard realized that a potential problem might exist with TCP queries. TCP puts a higher burden on the DNS servers. (TCP is much more expensive to process than UDP.) To avoid too much TCP traffic, the authors made the EDNS0 extension mandatory for DNSSEC. EDNS0 is one of the *Extension Mechanisms for DNS* (EDNS), a standard that, among other things, allows a client to signal that it is capable of receiving DNS replies over UDP that are larger than the previous limit of 512 bytes. Some firewalls are not aware of the fact that the EDNS0 standard allows for larger packets and they either block any DNS packet using EDNS0, or block any DNS packet larger than the 512 bytes regardless of the EDNS0 signaling.

Other firewalls allow for the large packets by default, whereas a few vendors require the firewall to be manually configured to do so. Any device in the path that does packet inspection at the application layer must be aware of the EDNS0 standard to be able to make a correct decision about whether to forward the packet or not. ICANN has summarized the status of EDN0 support in some commonly used firewalls[6].

Note that it is not enough to test that your firewall allows large incoming DNS replies by sending DNS queries to the Internet[7]. You must also test that an external source can receive large DNS replies that your DNS server is sending. One way of doing so is to use an open DNSSEC-aware resolver[8, 9].

> *Test and configure your firewall to allow for use of EDNS0 and for DNS packets larger than 512 bytes over UDP.*

Preparing Your Slaves

Setting up DNSSEC involves substantial changes to the master name server so it can sign and serve the signed data. However, it is easy to foresee that the slaves must be upgraded, too. The slaves are much easier to upgrade and operate because they never produce signatures.

They are secondary systems that transfer data from the primary server and respond to DNS queries. But the slaves must understand how to respond to queries requesting signed data.

Slaves must be upgraded to BIND 9.3 or better to understand the *Next Secure* (NSEC)[14] standard. NSEC is a method to provide authenticated denial of existence for DNS resource records. The newer *Next Secure 3* (NSEC3)[10] standard introduces some additional requirements for the slaves. If you use NSEC3, you must upgrade the slaves to BIND 9.6 or later. Version 3 of *Name Server Daemon* (NSD)[17] and any version of *Secure64 DNS Authority/Signer*[18] can do both NSEC and NSEC3. Windows Server 2008 R2 for the x86-64 architecture supports DNSSEC as a master, slave, and validating resolver. However, we recommend limiting the use of the Windows platform to slaves and for domains using NSEC. Our opinion is that it is very hard to implement DNSSEC on Windows, and we suggest that you wait until Microsoft offers a sensible *Graphical User Interface* (GUI) and support for NSEC3. Note that the Itanium version of Windows 2008 R2 supports neither DNS nor DNSSEC.

> *Make sure your slaves can handle the version of DNSSEC you intend to use.*

If the slaves are administered by another party, contact the administrator before you begin DNSSEC implementation. Make sure the slaves are running a version capable of DNSSEC. Stephan helped a large U.S. federal agency sign its domains. The agency used one of the major federal contractors to run its slave servers. After multiple attempts to reach somebody that understood DNS and DNSSEC, Stephan finally learned that the slaves were running BIND 9.2.3 and that the contractor had no plans to upgrade. The only alternative for the agency was to in-source the slaves and run them itself.

> *If your slaves are administered by another party, make sure you know if and what version of DNSSEC that party supports before you start implementing.*

Communicate with Your Parent

TLDs allow you to communicate with them in two ways:

- *Registrant–Registrar–Registry Model:* In this, the most common model, the registrant (**`example.org`**) does not communicate directly with the registry (**`.org`**). Instead, a third-party registrar handles all communication related to DNS and DNSSEC. This model is, for example, used by the **`.se`** and **`.org`** TLDs.
- *Registrant–Registry Model:* This model is normally used by smaller TLDs such as **`.gov`**. It allows direct communication between the registrant (**`agency.gov`**) and the registry (**`.gov`**). The TLD acts as both a registrar and a registry in this model.

Most problems described in the following paragraphs apply to both models, but those involving multiple registries are obviously applicable only to the Registrant–Registrar–Registry model.

Establishing a *Chain of Trust* in DNSSEC involves uploading one or more public keys to the parent. Ultimately the parent publishes a *Delegation Signer* (DS) record, a smaller fingerprint that can be constructed from the DNSKEY record. To upload your keys, you must use a registrar that supports DNSSEC. If your registrar does not support DNSSEC, you need to move your domains to another registrar (or convince your current registrar to start supporting DNSSEC). It usually takes a few days or up to a week to move a domain from one registrar to another.

Make sure that your registrar supports DNSSEC. If it does not, move your domain to a registrar that supports DNSSEC before you begin signing your zone.

Some registrars allow registration under multiple TLDs. However, just because a registrar handles DNSSEC for one TLD does not mean that it handles DNSSEC for all TLDs it serves. For example, several registrars in Sweden support DNSSEC for **`.se`** but not for **`.org`** or **`.us`**.

Make sure that your registrar handles DNSSEC under the TLD in question.

Most registrars offer you the opportunity to use their name server instead of your own. The service is either offered for free or for an additional cost. The registrar typically provides a web interface where you can change your zone data. This service is a good and useful choice if your domains are uncomplicated and small. Larger and more complex domains are better operated on your own servers.

Some registrars that provide this type of service can handle DNSSEC only if you use their name servers and not your own name servers. These registrars can establish the chain of trust with the parent only if the zone is under their control. They lack a user interface for uploading a DS key that you generate on your own name servers.

If you intend to use your own name servers, make sure that your registrar supports this deployment model, and allows you to upload a DS record for further distribution to the registry.

In theory, the child zone system should create the DS record fingerprint and upload it to the parent. In practice, some registrars require you to upload the DNSKEY record to them. They then create the DS record for you. (This practice is bad because the registrar must know the hash algorithm used to construct the DS record, which it might not know.) The DNSKEY record comes in several different formats, depending on the platform you used to create the keys (BIND, Microsoft, NSD, Secure64, etc.). The formats have minor differences, and you might have to convert the DNSKEY into a format that the registrar accepts.

Not everything works smoothly, even with the correct DNSKEY format. The logic at one registrar's website was to deny uploading of DNSKEYs unless the optional *Time To Live* (TTL) field existed. (The TTL value is useless in the DNSKEY context because the parent overrides this value with its own TTL). You may have to manually change your DNSKEY before uploading it to comply with the checks that the registrar performs.

> *If your registrar requires you to upload the DNSKEY, make sure that your solution can generate the requested format. If not, you need to manually change the fields with a text editor.*

As noted previously, some registrars are performing too many checks and irrelevant checks before accepting and creating the secure delegation. Other registrars do not check at all or have limited checks that do not work as expected. For example, some registrars assume that your key is created using a certain algorithm, and they do not double-check it prior to creating a DS record. One registrar created a bogus DS record if you uploaded a DNSKEY with upper-case characters in the domain name. The bogus DS record looked valid, and troubleshooting to find this error took hours.

Another example is keys created with *Webmin*[11], a graphical tool that you can use for signing zones. Webmin defaults to using the less-common *Digital Signature Algorithm* (DSA) for its DNSKEYs. The registrar did not complain when uploading the Webmin key, and it created a bogus DS record by assuming that it was an RSA key.

It is hard for a registrant to do anything about errors at the registrars. The best you can do is to make sure that you upload the correct key with the correct parameters such as algorithm, key length, key-id, etc. If something goes wrong, you might have to change the keys in production. Rolling the keys to the same algorithm and key length is relatively easy—but changing your keys to another algorithm adds extra complexity. It is an interesting exercise to change to another algorithm in production, but it is something we recommend avoiding if possible.

> *Double-check the DNSKEY/DS so that it is created with the correct parameters prior to uploading it.*

Communicate with Your Children

If you have sub-domains in your domain, you must make sure that you can accept and publish the DS records that your children upload to you. This situation is not a problem if you use zone files in text format—you can simply insert the DS record using your favorite editor. But it might be a problem if you are using an *Internet Protocol Address Management* (IPAM) system. In that case make sure that it can insert DS records into the zones that are managed by the system. Some IPAM systems do not support insertion of DS records correctly.

Make sure that your IPAM system can insert DS records into your zones.

A common strategy among organizations with high-availability requirements for their critical servers is to use a global load balancer, which is basically a DNS server that responds differently depending on the status of the service in question. For example, assume a load balancer can respond to a question for `www.example.com` with `192.0.2.1` and `192.0.2.2` if both web servers are up. If `.1` becomes unavailable, the load balancer notices a failure and responds only with `.2`. In order to use a global load balancer, you must delegate `www` as a sub-domain to its own DNS process.

When DNSSEC is implemented, you must make sure that the load balancer can handle DNSSEC (and not that many do); otherwise it is impossible to sign the responses for those resources. Unfortunately, these resources are the most critical ones for your environment and would benefit the most from DNSSEC signing.

Make sure that your load balancers support DNSSEC. If they do not, have an alternative strategy.

Rolling the Keys

You should change the DNSKEYs regularly and when you think the keys are compromised. The process of doing so is called *rolling the keys*. There are normally two different keys in DNSSEC, the *Key Signing Keys* (KSKs) and the *Zone Signing Keys* (ZSKs). Rolling the ZSK is an internal process and does not require communication with the parent. Rolling the KSK, on the other hand, requires the parent to publish a new DS record.[12]

There is no standard yet that describes how the communication between the parent and the child should occur when a key is rolled. Early DNSSEC-capable registrants used a web interface that allowed their registrants to upload and manipulate the DNSSEC information. With a web interface, each domain must be handled separately and there is no easy way to automate the interaction.

The web interface works for a handful of domains but becomes very cumbersome when you have many domains. For those types of organizations, it is important to make sure that there is some kind of *Application Programming Interface* (API) or script access to the registrar. This interface allows the organization to upload new DS records during the rollover in a convenient way.

Make sure that your registrar supports automation through an API if you have many domains.

Scripting with an API as described previously is one way of communicating with the registrar. Another way of achieving the same type of automation is for the parent (or registrar) to monitor the child for any changes to the DNSKEY records.

Note that the chain of trust is still intact during a nonemergency rollover. The parent can securely poll the child and grab the new DNSKEY records and convert them into DS records. The polling from the parent to each signed child needs to occur regularly so that a rollover is picked up quickly. This regularity of polling makes the scheme best for domains with fewer delegations (in the order of thousands, not millions—consider how much bandwidth an hourly polling of 15 million children would require).

Automation is a good thing, but make sure you understand the implications when opting for automatic detection of key rollovers. The automation scripts are not fail-safe. It has been reported that early versions of such scripts under some circumstances wrongly assumed that a key rollover occurred and deleted the DS record, thus breaking the chain of trust.

Understand the implication when opting for automatic detection, addition, and deletion of DS records.

Management of DNSSEC

Without DNSSEC, you are not bound to any particular registrar; you can switch to a new registrar fairly easily. With DNSSEC, this situation changes. First of all, if you let the registrar sign the zone on your behalf, the registrar will be in charge of the key used to sign your zone. Extracting your key so that it can be imported to another registrar is not always straightforward (also remember that there is really no incentive for your previous registrar to help you because you just discontinued its service). An alternative is to unsign the zone before you change registrars, but that option might not always be a viable one. The lack of standards makes it hard to change registrars on a signed domain that is in production.

You must tell your new registrar that you are using DNSSEC, and you must make sure that the registrar supports it. If not, the registrar might accept the transfer but be unable to publish the DNSKEY records. The result would be a DS record published by the registry but no corresponding DNSKEY records at the child, making the zone "security lame" and causing failed validation.

The same types of problems exist if you are running your own name servers. If you change your master server, make sure that you transfer the secret keys as well. Signing with new keys will not work unless you flush out the old keys with rollovers and upload a new DS record to your parent.

Have a plan ready for how to transfer your keys to a new master server.

Timers

It is important to adjust your signature validity periods and the *Start of Authority* (SOA) timers so that they match your organizational requirements and operational practices. SOAs expire and signature validity periods all too often are too short.

Unless you are restricted by guidelines saying otherwise, you should strive to set the timers reasonably high. Set the timers so that your zones can cope with an outage as long as the longest period that the system might be unattended.

For example, if you know that your top DNS administrator usually has three weeks of vacation in July, you could consider setting the times so that the zone can survive four weeks of downtime. If you are confident in your signing solution and are monitoring your signatures carefully, you might set it a little bit lower.

Signature lifetime is a trade-off between security (low signature lifetimes) and convenience (high signature lifetimes). Setting a really high signature lifetime is convenient from an operational perspective but is less secure. Some organizations such as the IETF use an excessive signature lifetime of one year (`dig ietf.org DNSKEY +dnssec | grep RRSIG`). This lifetime is clearly not recommended, and they should know better.

> *Carefully set your signature lifetimes and SOA times to reflect your organization's operational requirements and practices.*

A Note on Validation

This article has focused on the authoritative part of DNSSEC. That part includes signing resource records and serving DNS data. The operational challenges with signing data are much greater than the challenges of validating data. To validate data, the only thing you need to do regularly is update your trust anchor file. Make sure you do so. Torbjörn reports several outages when the `.se` DNSKEY used in the `.se` trust anchor expired in January 2010. We look forward to the work being done in this area to automate the process.

Summary

DNSSEC has been deployed and taken in production for several large and critical domains. It is not hard to implement DNSSEC, but doing so introduces some operational challenges. Those challenges exist both during the implementation phase when the zone is being signed for the first time and during the operation of the zone. Make sure you understand the possible effects of implementation and plan ahead. The following checklist summarizes the most important pitfalls with DNSSEC:

- Open your firewall for EDNS0 signaling and allow large DNS packets using UDP and TCP over port 53.
- Check the DNSSEC capabilities of all your masters and slave servers.
- Check the DNSSEC capabilities of your registrar and understand their requirements for the public key you are uploading.
- Make sure your IPAM system can handle secure delegations.

- Plan how to handle load balancers.
- Develop an automation strategy if you have a lot of zones.
- Plan how you will transfer your keys to a new master server if a disaster occurs.
- Implement a policy for DNSSEC timer settings.

Happy signing!

For Further Reading

[0] Miek Gieben, "DNSSEC: The Protocol, Deployment, and a Bit of Development," *The Internet Protocol Journal,* Volume 7, No. 2, June 2004.

[1] Carolyn Duffy Marsan, "80% of Government Web Sites Miss DNS Security Deadline," *Network World,* January 21, 2010, **http://www.networkworld.com/news/2010/012010-dns-security-deadline-missed.html**

[2] Jaokim Ålund and Patrik Wallström, "DNSSEC—Tests of Consumer Broadband Routers," **http://www.iis.se/docs/Routertester_en.pdf**

[3] Ray Bellis and Lisa Phifer, "Test Report: DNSSEC Impact on Broadband Routers and Firewalls," September 2008, **http://download.nominet.org.uk/dnssec-cpe/DNSSEC-CPE-Report.pdf**

[4] Thorsten Dietrich, "DNSSEC-Unterstützung durch Heimrouter," **http://www.denic.de/fileadmin/Domains/DNSSEC/DNSSEC_20100126_Dietrich.pdf**

[5] Broadband Home Gateway BoF, **http://tools.ietf.org/agenda/76/homegate.html**

[6] ICANN DNS Root Server System Advisory Committee (RSSAC) and Security and Stability Advisory (SSAC), "Testing Firewalls for IPv6 and EDNS0 Support," January 2007. **http://www.icann.org/en/committees/security/sac016.htm**

[7] Doman Name System Operations Analysis and Research Center (OARC)'s DNS Reply Size Test Server: **https://www.dns-oarc.net/oarc/services/replysizetest**

[8] OARC's Open DNSSEC Validating Resolver: **https://www.dns-oarc.net/oarc/services/odvr**

[9] Comcast DNSSEC Information Center, **http://www.dnssec.comcast.net/**

[10] Torbjörn Eklöv, "DNSSEC: Will Microsoft Have Enough Time?" *CircleID,* January 2010, **http://www.circleid.com/posts/dnssec_will_microsoft_have_enough_time/**

[11] **http://www.webmin.com/**

[12] George Michaelson, Patrik Wallström, Roy Arends, and Geoff Huston, "Rolling over DNSSEC Keys," *The Internet Protocol Journal,* Volume 13, No. 1, March 2010.

[13] Roy Arends, Rob Austein, Dan Massey, Matt Larson, and Scott Rose, "DNS Security Introduction and Requirements, RFC 4033, May 2005.

[14] Roy Arends, Rob Austein, Matt Larson, Dan Massey, and Scott Rose, "Resource Records for the DNS Security Extensions," RFC 4034, May 2005.

[15] Roy Arends, Rob Austein, Dan Massey, Matt Larson, and Scott Rose, "Protocol Modifications for the DNS Security Extensions," RFC 4035, May 2005.

[16] United States Computer Emergency Readiness Team (US-CERT), "Multiple DNS Implementations Vulnerable to Cache Poisoning," July 2008, **http://www.kb.cert.org/vuls/id/800113**

[17] **http://nlnetlabs.nl/projects/nsd/**

[18] **http://www.secure64.com/secure-DNS**

STEPHAN LAGERHOLM is a Senior DNS Architect with Secure64 Software Corporation—a software company offering high-performance DNS server software that makes the DNS trustworthy and secure. Secure64 DNS applications include key management and zone-signing software that make it easy to deploy DNSSEC securely and correctly as well as DNS server software that is always available. Stephan is a DNS and security expert with more than 11 years of international experience in the field. His background includes leadership positions at the largest networking and security system integrator in Scandinavia, and responsibility for designing hundreds of complex IT networks. Stephan is one of the few persons in the United States to have integrated DNSSEC into production environments. Stephan is CISSP-certified and holds a Master of Science degree in Computer Science and Mathematics from Uppsala University in Sweden. E-mail: **Stephan.Lagerholm@secure64.com**

TORBJÖRN EKLÖV is the founder and partner of Interlan Gefle AB, an IT consulting company in Sweden with 20 employees. He is a DNSSEC and IPv6 pioneer. All internal and external services at Interlan use both IPv6 and IPv4, and the company hosts about 200 DNSSEC-signed domains. Torbjörn has worked with Internet communication and security for 15 years, and is the founder and manager of Secure End User Connection (SEC), or Säker KundAnslutning (SKA) in Swedish, an organization that certifies products and broadband networks to protect subscribers from spoofing and hijacking. His favorite homepage is **http://test.ipv6.tk**. You can reach him at **Torbjorn.Eklov@interlan.se**

Book Review

The Art of Scalability

The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise, by Martin L. Abbott and Michael T. Fisher, ISBN-13: 978-0-13-703042-2, Pearson Education, 2010.

It is often claimed that the primary lesson of the Internet is one of "scaling." So the title of this book bodes well for relevance to Internet designers. A reader would likely expect discussion of hashing algorithms, fast-path coding, protocol latencies and chattiness, distributed redundancy design, and similar guidance for handling a billion users. The reader would largely be wrong, although some of the book is dedicated to technical performance. What is easily missed in the title is the word "organizations." It does not mean organization of modules. It means organizations within a *company.*

This book is very much a holistic one. It takes the painfully realistic position that well-designed protocols and software modules matter only if the company structure or team operation is tuned to growing and running a large-scale service. The book is comprehensive and primarily tailored for highly formal management, with substantial, bureaucratic procedures designed to ensure thorough consideration of scalability needs and implications. It is loaded with discussion of many different organizational and technical management tools that assist in making diligent decisions. For most readers and most companies, attempting to apply this level of formality is dramatic overkill. However, knowing about it is not.

The book is 533 pages, with 33 chapters and 3 appendices. The writing style is reasonably clean, but pedantic. Don't expect the type of entertainment-oriented writing that is common these days. The authors' experiences include *eBay* and *PayPal,* so scaling matters have been within their direct work responsibilities. As holds for any book attempting this kind of breadth, from technology design to organization management, discussion frequently is superficial and will be obvious to some readers, while the specific detail will in places be irrelevant to many others. Although these characteristics might be taken as negatives, they actually serve to demonstrate the utility of the book as an introduction and basic reference to the topic of scaling. A quick scan of the book helps the reader see how many different aspects of an organization's activities can aid or hinder large-scale operations. Exploring specific chapters can explain concepts and topics and suggest particular tools to help in planning or analysis.

Organization

Part I, "Staffing a Scalable Organization," comprises six chapters. It provides a tutorial on classic problems in structuring and staffing an organization for growth. Little is taken for granted. So there is guidance about the characteristics needed in a CEO, CFO, or CTO for aiding leadership in working to scale the company and the company's products. It even has a chapter on "Leadership 101."

For the most part, this section is likely to be useful only for readers with no management background, because the material is extremely basic. What distinguishes it is only the constant consideration of the way its topics are relevant to scaling. The likely utility of the section is in helping employees "manage up" so they can interact with management better when seeking support for changes needed to implement or maintain scalable development or operations. On the other hand, an interesting discussion explored why some simple and entirely logical choices for organizing a company work against accountability and scaling.

Part II, "Building Processes for Scale," at nearly 200 pages is 40 percent of the book. Whereas the first part concerned the people, this one concerns what they do. The first half of this part strongly emphasizes processes for anticipating and responding to scaling problems and for judiciously allocating limited resources. Hence there is even a chapter that considers "build versus buy." Technical topics discussed here are conceptual rather than concrete. They concern risk, performance, capacity, and failure recovery. Each is treated as a planning and design concern, with estimates and procedures. A warning: The word "architecture" shows up in the title of several middle paragraphs in this section, but don't be confused. It refers to groups that do architecture, not to the technical details of architecture.

Part III is "Architecting Scalable Solutions." Now at last, techies will start to get their geek fix. But perhaps with more abstraction than they will expect? Again, this book is more about properly organizing things than about algorithms. The section introduces "technology-agnostic design," with consideration of fault isolation and various growth factors, including repeated attention to cost, risk, scalability, and availability. There are chapters on database scaling and the use of caching for performance. The authors are fond of asynchronous and state-free interaction, with the view that it is more robust. The precise reason for this conclusion was not entirely clear to me, but presumably it is because it is easier to recover and retarget an exchange after an outage occurs during an interaction.

Two chapters of this part of the book are devoted to the "AKF Scale Cube," and indeed the Index has a large number of citations to it. (AKF refers to the authors' company.) For this analytic tool, the x-axis "...represents cloning of services and data with absolutely no bias." In other words, these graphs are pure replications of equivalent, parallel components or activities, used to distribute load. The y-axis "... represents a separation of work responsibility by either the type of data, the type of work performed for a transaction, or a combination of both... We often refer to these as service or resource oriented splits." The nature of the z-axis is described as "...biased most often by the requestor or customer... focused on data and actions that are unique to the person or system performing the request." I took this as meaning that the axis divides work according to tailored attributes.

Part IV is the catchall for remaining topics, with some requisite discussion of clouds and grids, application monitoring, and data center planning.

Summary

The book will be useful for architects who need to understand how to scale their own work and how to support their organization for long-term growth. It will also be useful for technical, operations, and other managers who need to understand the technical and operations scaling problems, support their own architects, and work with the rest of their organization to anticipate and satisfy scaling requirements.

—Dave Crocker, Brandenburg InternetWorking
dcrocker@bbiw.net

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

Call for Candidates for Itojun Service Award

The *Itojun Service Award* is presented every year to an individual or a group who has made outstanding contributions in service to the IPv6 community. The deadline for nominations for this year's award is July 12, 2010. The award will be presented at the 79th meeting of the *Internet Engineering Task Force* (IETF) to be held in November 2010 in Beijing, China.

The Itojun Service Award, established by the friends of Itojun and administered by the *Internet Society* (ISOC), recognizes and commemorates the extraordinary dedication exercised by Itojun over the course of IPv6 development. The award includes a presentation crystal, a US$3,000 honorarium, and a travel grant.

The award is focused on pragmatic technical contributions, especially through development or operation, with the spirit of servicing the Internet. With respect to the spirit, the selection committee seeks contributors to the Internet as a whole; open source developers are a common example of such contributors, although this is not a requirement for expected nominees. While the committee primarily considers practical contributions such as software development or network operation, higher-level efforts that help those direct contributions will also be appreciated in this regard. The contribution should be substantial, but could be immature or ongoing; this award aims to encourage the contributors to continue their efforts, rather than just recognizing well-established work. Finally, contributions of a group of individuals will be accepted as deployment work is often done by a large project, not just a single outstanding individual.

The award is named after Dr. Jun-ichiro "Itojun" Hagino, who passed away in 2007, aged just 37. Itojun worked as a Senior Researcher at *Internet Initiative Japan Inc.* (IIJ), was a member of the board of the *Widely Integrated Distributed Environment* (WIDE) project, and from 1998 to 2006 served on the groundbreaking KAME project in Japan as the "IPv6 Samurai." He was also a member of the *Internet Architecture Board* (IAB) from 2003 to 2005.

For additional information on the award, please visit:
http://www.isoc.org/awards/itojun/

Less than 10% of IPv4 Addresses Remain Unallocated, says NRO

The *Number Resource Organization* (NRO), the official representative of the five *Regional Internet Registries* (RIRs) that oversee the allocation of all Internet number resources, recently announced that less than 10 percent of available IPv4 addresses remain unallocated. This small pool of existing IP addresses marks a critical moment in IPv4 address exhaustion, ultimately impacting the future network operations of all businesses and organizations around the globe.

"This is a key milestone in the growth and development of the global Internet," noted Axel Pawlik, Chairman of the NRO. "With less than 10 percent of the entire IPv4 address range still available for allocation to RIRs, it is vital that the Internet community take considered and determined action to ensure the global adoption of IPv6. The limited IPv4 addresses will not allow us enough resources to achieve the ambitions we all hold for global Internet access. The deployment of IPv6 is a key infrastructure development that will enable the network to support the billions of people and devices that will connect in the coming years," added Pawlik.

The *Internet Protocol* (IP) is a set of technical rules that defines how devices communicate over a network. There are currently two versions of IP, IPv4 and IPv6. IPv6 includes a modern numbering system that provides a much larger address pool than IPv4. With so few IPv4 addresses remaining, the NRO is urging all Internet stakeholders to take immediate action by planning for the necessary investments required to deploy IPv6.

The NRO, alongside each individual RIR, has actively promoted IPv6 deployment for several years through grassroots outreach, speaking engagements, conferences and media outreach. To date, their combined efforts have yielded positive results in the call to action for the adoption of IPv6.

Given the less than 10 percent milestone, the NRO is continuing its call for Internet stakeholders, including governments, vendors, enterprises, telecoms operators, and end users, to fulfill their roles in IPv6 adoption, specifically encouraging the following actions:

- The business sector should provide IPv6-capable services and platforms, including web hosting and equipment, ensuring accessibility for IPv6 users.
- Software and hardware vendors should implement IPv6 support in their products to guarantee they are available at production standard when needed.
- Governments should lead the way by making their own content and services available over IPv6 and encouraging IPv6 deployment efforts in their countries. IPv6 requirements in government procurement policies are critical at this time.
- Civil society, including organizations and end users, should request that all services they receive from their ISPs and vendors are IPv6-ready, to build demand and ensure competitive availability of IPv6 services in coming years.

The NRO's campaign to promote the next generation of Internet Protocol continues to positively impact the Internet community. IPv6 allocations increased by nearly 30% in 2009, as community members continued to recognize the benefits of IPv6.

"Many decision makers don't realize how many devices require IP addresses—mobile phones, laptops, servers, routers, the list goes on," said Raul Echeberria, Secretary of the NRO. "The number of available IPv4 addresses is shrinking rapidly, and if the global Internet community fails to recognize this, it will face grave consequences in the very near future. As such, the NRO is working to educate everyone, from network operators to top executives and government representatives, about the importance of IPv6 adoption," added Echeberria.

IP addresses are allocated by the *Internet Assigned Numbers Authority* (IANA), a contract operated by the *Internet Corporation for Assigned Names and Numbers* (ICANN). IANA distributes IP addresses to RIRs, who in turn issue them to users in their respective regions. "This is the time for the Internet community to act," said Rod Beckstrom, ICANN's President and Chief Executive Officer.

"For the global Internet to grow and prosper without limitation, we need to encourage the rapid widespread adoption of the IPv6 protocol," he added.

The NRO is the coordinating mechanism for the five RIRs. The RIRs—AfriNIC, APNIC, ARIN, LACNIC, and the RIPE NCC—ensure the fair and equitable distribution of Internet number resources (IPv6 and IPv4 addresses and *Autonomous System* (AS) numbers) in their respective regions. The NRO exists to protect the unallocated Internet number resource pool, foster open and consensus-based policy development, and provide a single point of contact for communication with the RIRs.

Learn more about the NRO at **`www.nro.net/media`**

The five RIRs that make up the NRO are independent, not-for-profit membership organizations that support the infrastructure of the Internet through technical coordination. The IANA allocates blocks of IP addresses and ASNs, known collectively as *Internet number resources,* to the RIRs, who then distribute them to users within their own specific service regions. Organizations that receive resources directly from RIRs include *Internet Service Providers* (ISPs), telecommunications organizations, large corporations, governments, academic institutions, and industry stakeholders, including end users. The RIR model of open, transparent participation has proven successful at responding to the rapidly changing Internet environment. Each RIR holds one or two open meetings per year, as well as facilitating online discussion by the community, to allow the open exchange of ideas from the technical community, the business sector, civil society, and government regulators.

The five RIRs are:

- AfriNIC: **http://www.afrinic.net**
- APNIC: **http://www.apnic.net**
- ARIN: **http://www.arin.net**
- LACNIC: **http://www.lacnic.net**
- RIPE NCC: **http://www.ripe.net**

ISOC Funds Projects to Support Internet Access, Security, and Policy Development

The *Internet Society* (ISOC) recently announced it is funding community-based projects around the world addressing issues such as Internet leadership, education, core infrastructure, local governance, and policy development, with a strong focus on currently underserved communities.

"The diversity of projects awarded highlights the profound importance of the Internet in so many aspects of our lives, in all parts of the world," said Jon McNerney, Chief Operating Officer of the Internet Society. "The passion and creativity of those developing the projects within their communities drives the Internet Society's commitment to help bring the benefits of the Internet to people everywhere."

As part of the ISOC *Community Grants Program,* each project will receive up to US$10,000 for efforts that promote the open development, evolution, and use of the Internet for the benefit of all people throughout the world.

Projects funded in this round include:

- Training programs to build digital literacy within safe environments in India and Uganda
- Village-operated telecommunication services in East Timor
- Support for development of core Internet time infrastructure
- Policy and practical action in Kenya to improve online safety for women
- Online support for NGOs in Tunisia and more effective local governance in India
- Promotion of Internet leadership in Ecuador
- Development of important public policy resources in Georgia and Australia

ISOC Community Grants are awarded twice each year. The next round of the program will open on September 1, 2010. Additional information about the Community Grants Program and this round of award-winning projects can be found here:

https://www.isoc.org/isoc/chapters/projects/index.php

https://www.isoc.org/isoc/chapters/projects/awards.php?phase=11

RIPE Community Statement on the Internet Address Management System

At the May 2010 *Réseaux IP Européens* (RIPE) meeting in Prague, Czech Republic, the RIPE community issued the following statement:

"The RIPE community supports all efforts to assist in the deployment of IPv6, especially in developing countries.

However, we note concerns being expressed within the ITU by a few members, most recently in the ITU IPv6 Group, that the current address management system is inadequate.

The RIPE community mandates the RIPE NCC to work with the ITU IPv6 Group, individual ITU members, and the community to clearly identify these concerns and to find ways to address them within the current IP address management system."

This statement will be sent to the *International Telecommunications Union* (ITU) to reiterate the RIPE community's belief that the current address management system works. The RIPE NCC will continue to participate actively in the ITU IPv6 Group and report back to the RIPE community.

For more information see:

http://www.itu.int/ITU-T/othergroups/ipv6/
http://ripe.net/ripe/index.html
http://www.nro.net/documents/nro51.html

Upcoming Events

The *North American Network Operators' Group* (NANOG) will meet in San Francisco, California, June 13–16, 2010.
See **http://nanog.org**

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in Brussels, Belgium, June 20–25, 2010.
See **http://icann.org**

The *Internet Engineering Task Force* (IETF) will meet in Maastricht, The Netherlands, July 25–30, 2010 and in Beijing, China, November 7–12, 2010. See **http://www.ietf.org/**

APNIC, the *Asia Pacific Network Information Centre,* will hold its Open Policy meeting in the City of Gold Coast, Australia, August 24–28, 2010. See **http://www.apnic.net/meetings/30/**

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

September 2010 Volume 13, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

Technology advances—such as improvements in display technology, battery life, processor capabilities, and communications systems—have all contributed to making *mobile devices* the most important area for Internet growth. In order to fully support these devices, the IETF developed *Mobile IP* many years ago, and it has continued to work on the general area of IP mobility. We have covered some of this work in previous issues of IPJ, and this time we look at *Proxy Mobile IPv6* (PMIPv6), which is being standardized by the IETF. The article is by Ignacio Soto, Carlos J. Bernardos, María Calderón, and Telemaco Melia.

Deployment of IPv6 is progressing, albeit slowly. In several upcoming articles we will examine some transition technologies or implementation details that can make this deployment easier, and above all, transparent, to the end user. In our first article, Dan Wing and Andrew Yourtchenko explain the concept of "Happy Eyeballs" as applied to dual-stack IPv4/IPv6 systems.

Domain Name System Security Extensions (DNSSEC) have recently been applied to the Internet system of root servers. For details, see our "Fragments" section, where you will also find a statement from the *Number Resource Organization* (NRO) regarding the results of a recent IPv6 readiness study.

Once again, please remember to check your subscription expiration date and take the necessary steps if you wish to continue receiving this journal. It's not too late to renew and get back on the distribution list, even if your subscription expired some time ago. You can find your subscription ID and expiration date either on the back page of your copy or on the envelope that it came in. In order to access your record, click the "Subscriber Services" link on our webpage at `www.cisco.com/ipj` and enter your e-mail address and the subscription ID. The system will send you a link that allows direct access to your record, and you can update your address and renew your subscription. If you no longer have access to the e-mail you used when you subscribed or have forgotten your subscription ID, just send a message to `ipj@cisco.com` and we will make the necessary changes for you.

—Ole J. Jacobsen, Editor and Publisher
`ole@cisco.com`

PMIPv6: A Network-Based Localized Mobility Management Solution

by Ignacio Soto, Universidad Politécnica de Madrid; Carlos J. Bernardos, and María Calderón, Universidad Carlos III de Madrid; and Telemaco Melia, Alcatel Lucent Bell Labs

Traditional IP mobility procedures[4] are based on functions residing in both the mobile terminal and the network. Recently, we have been assisting in a shift in IP mobility protocol design, mostly focusing on solutions that relocate mobility procedures from the mobile device to network components. This new approach, known as *Network-Based Localized Mobility Management* (NetLMM), allows conventional IP devices (for example, devices running standard protocol stacks) to roam freely across wireless stations belonging to the same local domain. This property is appealing from the operator's viewpoint because it allows service providers to enable mobility support without imposing requirements on the terminal side (for example, software and related configuration). For this purpose the *Internet Engineering Task Force* (IETF) has standardized *Proxy Mobile IPv6* (PMIPv6)[1].

This article details the Proxy Mobile IPv6 protocol, providing a general overview and an exhaustive description of a few selected functions.

Why Network-Based Localized Mobility?

The ability to move while being connected to a communication network is very attractive for users, as demonstrated by the success of cellular networks. However, while designing the IP stack, mobility was not retained as a requirement and, as a consequence, IP does not natively support mobility. The reason is a very basic design choice adopted in IP, both in IPv4[2] and in IPv6[3], namely that addresses have two roles: they are used as locators and identifiers at the same time.[16]

IP addresses are *locators* that specify, by means of the routing system, how to reach the node (more properly, the *network interface*) that is using a specific destination address. The routing system keeps information about how to reach different sets of addresses that have a common network prefix, thus improving scalability of the system itself. However, IP addresses are also *identifiers* used by upper-layer protocols (for example, the *Transmission Control Protocol* [TCP]) to identify the endpoints of a communication channel. Additionally, names of nodes are translated by the *Domain Name System* (DNS) to IP addresses (which, in that way, play the role of node identifiers).

The linking of these two roles (*locators* and *identifiers*) is appealing because name resolution of the peer with whom we want to communicate and location finding translate to the same problem (that is, no translation mechanism is needed). However, the negative side effect is that supporting mobility becomes difficult.

Mobility implies separating the identifier role from the location one. From the identification standpoint, the IP address of a node should never change, but from the location point of view the IP address should change each time the node moves, showing its current location within the routing hierarchy (that is, the IP subnet to which the node is currently attached).

The IETF has studied the problem of terminal mobility in IP networks for a long time. It has developed IP-layer solutions for both IPv4 (Mobile IPv4[4, 5]) and IPv6 (Mobile IPv6[6]), enabling the movement of terminals and providing transparent service continuity. These solutions, being IP-based, are independent of the Layer 2 technologies. They provide Mobile Nodes with a permanent address (the *Home Address* [HoA]) to be used as identifier, and a temporal address (the *Care-of Address* [CoA]) to be used as locator. The CoA changes in each IP subnet visited by the Mobile Node. An entity in the network, the *Home Agent,* binds both addresses with the help of signaling generated by the Mobile Node. The Home Agent serving a Mobile Node must be placed in the subnet where the Home Address of that Mobile Node is topologically correct (the home network).

Although Mobile IP enables a host to move (that is, change the point of attachment in an IP network) while keeping session continuity, this ability is not sufficient for true mobility. Enabling efficient handoffs is an additional and critical requirement. Because the IP handoff latency is affected by the time required to exchange signaling between the Mobile Node and the Home Agent, a new family of solutions proposes to use a local Home Agent (that is, a Home Agent closer to the Mobile Node) to provide mobility in a local domain; that is, to provide localized mobility support. Changing the point of attachment within the local domain requires only signaling to the local Home Agent, allowing faster signaling messages exchange because it is limited within the local domain. This approach is attractive because users typically move in localized environments (for example, they commute between their living homes and their work places) that can be covered with localized domains. Examples of these types of solutions are "Regional Registrations for IPv4"[7] or "Hierarchical Mobile IPv6 for IPv6"[8]. Note that the term "localized" refers to a particular area from the point of view of the IP network topology, but depending on the access technology, geographically the area can be large, as happens when applying a localized mobility approach to cellular networks.

A common feature of Mobile IP and the localized mobility proposals mentioned previously is that all of them are *host-based*. Mobile Nodes must signal themselves to the network when their location changes and must update routing states in the Home Agent, in the local Home Agent, or in both. This situation also raises the problem of complex security configurations to authenticate those signaling exchanges and modifications of routing states.

Therefore, the IETF decided to work on a solution for NetLMM[10, 11], compounding the advantages of a network-based approach with the benefits of localized mobility management strategies. In NetLMM the network provides mobility support, although the Mobile Node does not participate in IP mobility procedures. That is, network operators can provide mobility support without requiring additional software and complex security configuration in the Mobile Nodes. Thus the deployment of network-based mobility solutions is greatly facilitated. Moreover, the Mobile Node can implement any global mobility solution, because the localized one is transparent and independent from it.

There are several target scenarios for Network-Based Localized Mobility Management[9]:

- Large campus networks with *Wireless Local-Area Network* (WLAN) access: Users move with IP standard devices (that is, no additional hardware or software is required) within a campus that provides WLAN access and mobility support.
- Advanced beyond-third-generation (3G) networks: Cellular operators have been important promoters in the development of the NetLMM solution in the IETF. *Universal Mobile Telecommunications System* (UMTS) and *General Packet Radio Service* (GPRS) networks use a proprietary network-based localized mobility mechanism to provide mobility support for user data traffic (typically IP). This mechanism is based on the GPRS Tunneling Protocol[11], a special-purpose solution developed for *Third-Generation Partnership Project* (3GPP) networks that uses TCP/IP application layer tunnels. A standardized NetLMM protocol for the Internet has important advantages:
  - Reduced costs in network management and in equipment supporting the technology (because of economy of scale)
  - Easier extension of mobility support to other technologies
  - Easier integration with other networks
- Other more-complex scenarios involving network mobility, as in automotive scenarios[12], could benefit from a NetLMM approach to support mobility.

With these advantages in mind, the IETF has standardized a protocol to provide Network-Based Localized Mobility support in IP networks, the *Proxy Mobile IPv6* (PMIPv6) protocol.

Operation of Proxy Mobile IPv6

The main idea of PMIPv6 is that the mobile node is not involved in any IP layer mobility-related signaling. The Mobile Node is a conventional IP device (that is, it runs the standard protocol stack). The purpose of PMIPv6 is to provide mobility to IP devices without their involvement. This provision is achieved by relocating relevant functions for mobility management from the Mobile Node to the network.

PMIPv6 provides mobility support within a localized area, the *Localized Mobility Domain* (LMD) or PMIPv6 domain. While moving within the LMD, the Mobile Node keeps its IP address, and the network is in charge of tracking its location. PMIPv6 is based on *Mobile IPv6* (MIPv6), reusing the Home Agent concept but defining nodes in the network that must signal the changes in the location of a Mobile Node on its behalf.

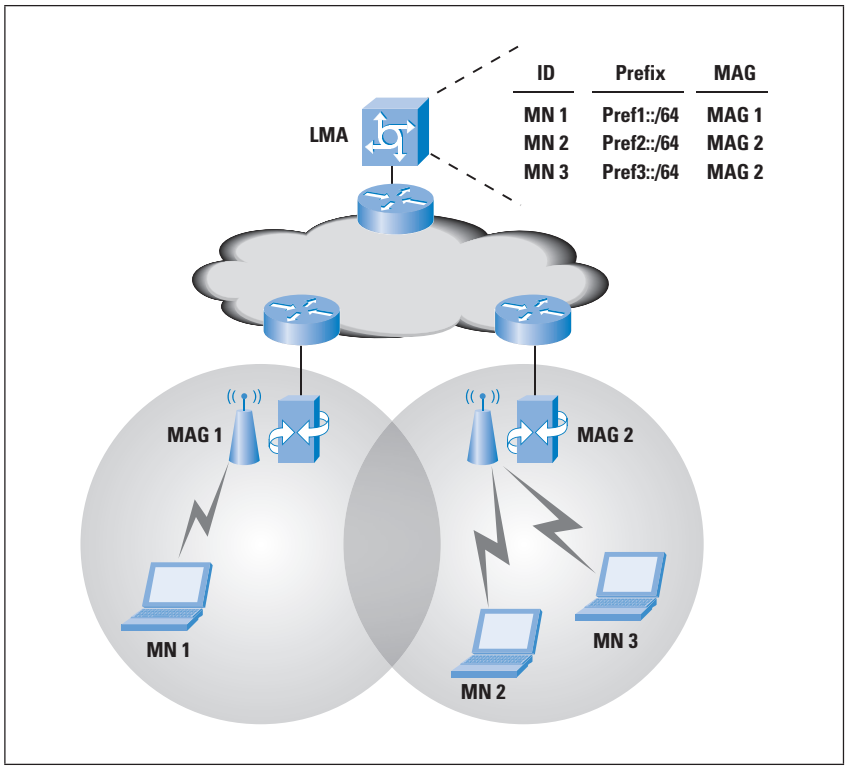


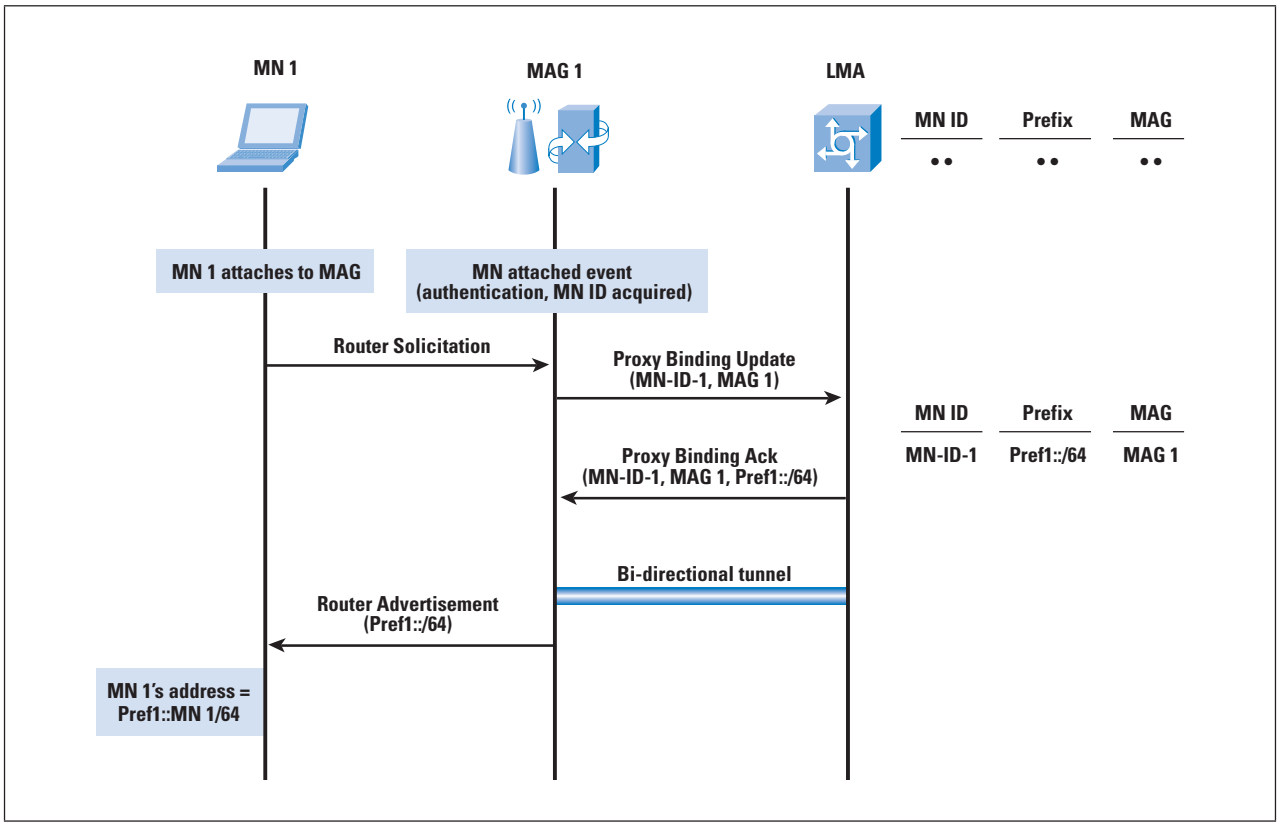
Figure 1: Network Entities in Proxy Mobile IPv6

The functional entities in the PMIPv6 network architecture (refer to Figure 1) include the following:

- *Mobile Access Gateway* (MAG): This entity performs the mobility-related signaling on behalf of the Mobile Nodes attached to its access links. The MAG is usually the access router for the Mobile Node, that is, the first-hop router in the Localized Mobility Management infrastructure. It is responsible for tracking the movements of the Mobile Node in the LMD. An LMD has multiple MAGs.
- *Local Mobility Anchor* (LMA): This entity within the core network maintains a collection of routes for each Mobile Node connected to the LMD. The routes point to MAGs managing the links where the Mobile Nodes are currently located. Packets sent or received to or from the Mobile Node are routed through tunnels between the LMA and the corresponding MAG. The LMA is a topological anchor point for the addresses assigned to Mobile Nodes in the LMD, meaning that packets with those addresses as destination are routed to the LMA.

The basic operation of PMIPv6 follows. When a Mobile Node enters a PMIPv6 domain, it attaches to an access link provided by a MAG. The MAG proceeds to identify the Mobile Node, and checks if it is authorized to use the network-based mobility management service. If it is, the MAG performs mobility signaling on behalf of the Mobile Node (see in Figure 2 the signaling when the Mobile Node enters the PMIPv6 domain). The MAG sends to the LMA a *Proxy Binding Update* (PBU) associating its own address with the identity of the Mobile Node (for example, its *Media Access Control* [MAC] address or an identifier related to its authentication in the network). Upon receiving this request, the LMA allocates a prefix to the Mobile Node. Then the LMA sends to the MAG a *Proxy Binding Acknowledgment* (PBA) including the prefix allocated to the Mobile Node. It also creates a *Binding Cache* entry and establishes a bidirectional tunnel to the MAG. The MAG sends *Router Advertisement* messages to the Mobile Node, including the prefix allocated to the Mobile Node, so the Mobile Node can configure an address (stateless autoconfiguration). The Mobile Node can alternatively use stateful address autoconfiguration mechanisms. For simplicity, we assume in the rest of the article that the stateless address autoconfiguration mechanism is used, except when indicated otherwise.

Figure 2: Signaling When a Mobile Node Connects to the PMIPv6 Domain



Whenever the Mobile Node moves, the new MAG updates the location of the Mobile Node in the LMA and advertises the same prefix to the Mobile Node (through Router Advertisement messages), thereby making the IP mobility transparent to the Mobile Node. In this way the Mobile Node keeps the address configured when it first enters the LMD, even after changing its point of attachment within the network, and the LMD appears as a single link from the perspective of the Mobile Node. It should be noted that all the MAGs configure the same link local address for a specific Mobile Node. That is, the Mobile Node will never see a change in its default route configuration.

The bidirectional tunnel between the LMA and the MAG and associated routing states in both LMA and MAG manage the Mobile Node data plane. Downlink packets sent to the Mobile Node from outside of the LMD arrive to the LMA, which forwards them through the tunnel to the serving MAG. The MAG, after decapsulation, sends the packets to the Mobile Node directly through the access link. Uplink packets that originated in the Mobile Node are sent to the LMA from the MAG through the tunnel, and then are forwarded to the destination by the LMA. Traffic originated inside the LMD and directed to a Mobile Node also inside the LMD follows a similar procedure, going through two tunnels from the originating MAG, to the LMA, and then to the destination MAG. It should be noted that PMIPv6 allows a MAG to short-circuit the tunneling in case two mobile nodes directly communicate through any of its interfaces.

Protocol Details

We next describe the PMIPv6 primary functions. Because PMIPv6 is based on the Mobile IPv6 protocol format, we will highlight the differences and extensions to MIPv6. Readers interested in knowing all protocol details should refer to the RFC[1].

Entering a PMIPv6 Domain

The Mobile Node enters the PMIPv6 domain by attaching to an access link. PMIPv6 defines a new functional entity, the MAG, typically residing in the access router. The MAG detects the attachment of the Mobile Node to the access link. The only access link types supported in PMIPv6 are point-to-point links; other types of links can be used as long as they are configured to emulate point-to-point links.

The MAG, upon detecting a Mobile Node attachment, verifies if the Mobile Node is eligible to the network-based mobility management service. Specific procedures to achieve this verification are out of the scope of the PMIPv6 standard. A Mobile Node that uses the mobility support service is identified by the network entities using a *Mobile Node Identifier* (MN-ID). The MN-ID must be stable and unique for the Mobile Node throughout the PMIPv6 domain, but the exact nature of this identifier is not specified. Possible examples are the Mobile Node MAC address or an identifier obtained as part of the Mobile Node authentication procedure.

After the MAG identifies the Mobile Node, authorizes its use of the NetLMM service, and acquires its Mobile Node Identifier, the MAG sends a PBU to the LMA; that is, it sends a registration request on behalf of the Mobile Node to the LMA. The PBU message is based on the MIPv6 *Binding Update* (BU) message with some extensions, but whereas the BU is sent by the Mobile Node, the PBU is sent by the MAG on behalf of the Mobile Node. A flag in the message is used to indicate that it is a PBU and not a BU. The PBU has as source address (and also in the alternate CoA option, if present) the global address configured in the egress interface of the MAG. This address is called *Proxy-CoA* in PMIPv6 terminology and is used by the LMA as locator of the Mobile Node. In the PBU, unlike in the BU, a Home Address destination option is not present; instead a *Mobile Node Identifier Option*[13] has to be included with the Mobile Node Identifier, which is used to identify the Mobile Node throughout the PMIPv6 domain.

The PBU also contains additional information, such as the access link technology, a handoff indicator, the requested lifetime for the registration, and other optional data. The *handoff indicator* is a new mobility option defined in PMIPv6 that allows the MAG to signal the LMA whether the PBU originated upon network attachment or upon handover of a Mobile Node (if known by some unspecified mechanisms), and that information could be useful to support advanced functions such as multihoming. Examples of values of the handoff indicator include: a Mobile Node entering the PMIPv6 domain, a reregistration to update the registration lifetime, a handoff between MAGs, or a handoff between interfaces of the Mobile Node.

Upon sending the PBU, the MAG creates a Binding Update List entry[6] for the Mobile Node. Note that this data structure in Mobile IPv6 is maintained by the Mobile Node to keep track of its bindings, but consequently to the PMIPv6 philosophy, the MAG maintains a *Binding Update List* (BUL) storing the bindings of the Mobile Nodes attached to it. The information in the Binding Update List allows the MAG to link the information about the Mobile Node, the interface in the MAG to which the Mobile Node is connected, and the LMA serving it, among others.

When the LMA receives the PBU sent by the MAG, it first checks that the message is correct according to the PMIPv6 specification, rejecting the registration otherwise. If the LMA accepts the PBU, it has to verify if its *Binding Cache* contains an entry for the Mobile Node identified in the PBU. When a Mobile Node first enters the PMIPv6 domain, the LMA cannot find an entry in its Binding Cache and has to create a new one. The Binding Cache entry is an extended version of the data structure defined for the Binding Cache entries in Mobile IPv6[6].

The entry in the Binding Cache has a flag to indicate that it is a proxy registration, and it links all the information related to the Mobile Node, including its identification and the MAG serving it; that is, the location of the Mobile Node. If there is no entry for the Mobile Node in the Binding Cache (that is, the Mobile Node is entering the PMIPv6 domain), the LMA allocates one or more network prefixes to the Mobile Node. These prefixes are called *Home Network Prefixes,* and it must be noted that at least one network prefix is assigned per Mobile Node.

If the LMA cannot allocate a network prefix to a Mobile Node, it has to reject the registration. The address(es) that the Mobile Node uses while inside the PMIPv6 domain are configured from those Home Network prefixes. The decision of allocating one or more network prefixes depends on a global policy in the PMIPv6 domain or a per-Mobile Node policy. When the registration request is accepted, the LMA creates a *Binding Cache Entry* (BCE) with the accepted values for the registration, including the Mobile Node Identifier, the Proxy CoA (the address of the MAG serving the Mobile Node), and the Home Network prefix(es) allocated to the Mobile Node.

Upon BCE creation, the LMA creates an IPv6-in-IPv6 bidirectional tunnel, if one does not already exist, to the MAG sending the PBU. The LMA sets up forwarding routes through the tunnel for any traffic received that is addressed to the Home Network prefixes of the Mobile Node. Finally, the LMA creates a *Proxy Binding Acknowledgment* (PBA) and sends it to the corresponding MAG. The PBA message is based on the MIPv6 *Binding Acknowledgment* (BA) message with a few more extensions, including a flag that indicates that the message is a Proxy Binding Acknowledgement. The PBA informs the MAG about the registration request result, if it has been rejected (and why, using a status code) or accepted. The PBA contains the Mobile Node Identifier and the Home Network prefixes allocated to the Mobile Node. Unlike the Binding Acknowledgment, the PBA does not include a type 2 routing header (that in the Binding Acknowledgment includes the Home Address of the Mobile Node). Also the PBA is received and processed by the MAG, and not by the Mobile Node.

If the PBA confirms that the registration request has been accepted for the Mobile Node, the MAG creates an IPv6-in-IPv6 bidirectional tunnel, if one does not already exist, to the LMA. The MAG sets up forwarding routes, through the tunnel, for uplink or downlink packets received or sent from or to the Mobile Node. The MAG also updates the Binding Update List entry to reflect the accepted binding registration values.

Upon network attachment and during the PBU or PBA procedure, the Mobile Node can send a *Router Solicitation* in the access link as part of the standard neighbor discovery procedures. The MAG should not reply to this Router Solicitation until the registration in the LMA has been successfully completed. When the MAG receives the PBA indicating a successful registration, the MAG sends a Router Advertisement to the Mobile Node announcing the Home Network prefix(es). The Mobile Node can then apply the stateless address autoconfiguration mechanism or the stateful one (using the *Dynamic Host Configuration Protocol* [DHCP]) according to the indication in the Router Advertisement. For supporting DHCP, a DHCP relay agent has to be present in every MAG in the domain, and the relay agent must include in the link-address field of the *Relay Forward* message an IPv6 address from the Home Network prefix, to indicate to the DHCP server the range of addresses it can assign.

The PMIPv6 specification, as mentioned previously, supports only point-to-point access links with the Mobile Nodes. An interesting use case is to have a broadcast access link and to emulate point-to-point links with the Mobile Nodes to be able to apply the PMIPv6 specification. This case raises the problem of sending Router Advertisements that should be received only by the corresponding Mobile Node, and not by other Mobile Nodes present in the broadcast link. There are several ways to send these advertisements. The Router Advertisements could be sent to the IPv6 link-local address of the Mobile Node that the MAG can learn from the source address of router solicitations sent by the Mobile Node, or by some other unspecified means. Another possibility is to send Router Advertisements to the all-nodes multicast address at the IP layer but to the Link Layer 2 address of the Mobile Node.

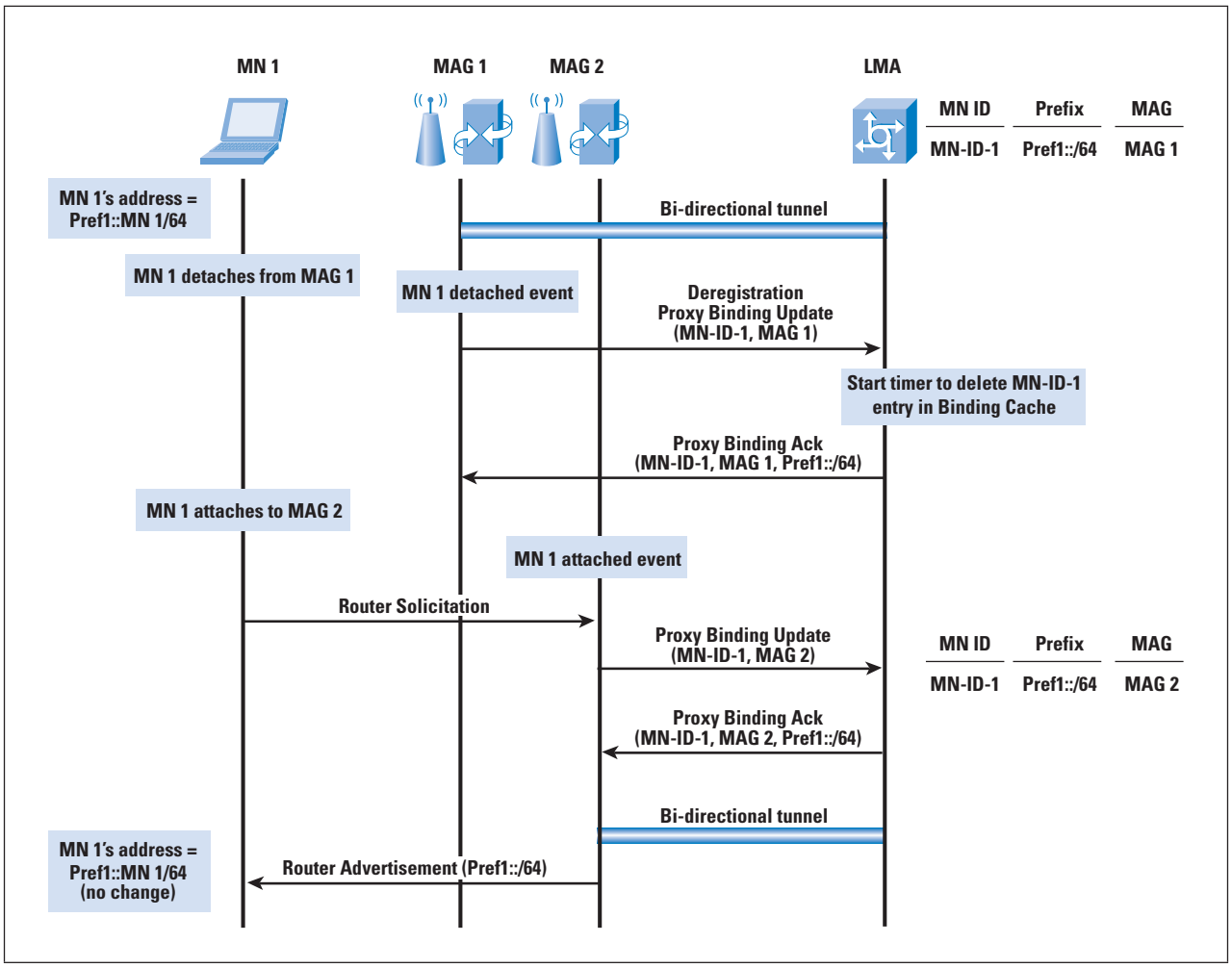
Changing MAG in a PMIPv6 Domain

The complete signaling for supporting the change of attachment by a Mobile Node in a PMIPv6 domain is described in Figure 3.

When a Mobile Node leaves a link, the event is detected by the corresponding MAG. The mechanism for Mobile Node movement detection is not specified in PMIPv6, but some possible options are link-layer events or an *IPv6 Neighbor Unreachability Detection* event. The MAG that detects that the Mobile Node has left the link must send a PBU with a Mobile Node de-registration request to the LMA. Upon receiving a PBA replying to the PBU or after a timer, the MAG deletes all the states associated with a specific Mobile Node.

When the LMA receives a PBU with a de-registration request for a Mobile Node with a valid entry in the Binding Cache, it sends the corresponding PBA and starts a timer. During the period defined by the timer the LMA drops any packets received for the Mobile Node. The use of this timer allows the LMA to receive a PBU from a new MAG updating the location of the Mobile Node. If the PBU is not received during that time, the LMA deletes the state associated with the Mobile Node.

Figure 3: Signaling When a Mobile Node Changes Point of Attachment



In a handoff situation the Mobile Node, after leaving a link, attaches to a new access link associated with a new MAG. The new MAG detects the Mobile Node and sends a PBU to the LMA on behalf of the Mobile Node. The LMA receives and processes the PBU, and detects that there is already a Binding Cache entry for that Mobile Node (the same Mobile Node Identifier). The LMA updates the Binding Cache entry with the new information, in particular with the Proxy CoA (egress IPv6 address) of the new MAG, updating also the tunnel and routing information for handling the traffic from or to the Mobile Node. The LMA sends a PBA to the new MAG in which it includes the Home Network prefix(es) already assigned to the Mobile Node. This scenario allows the new MAG to send a Router Advertisement with the same network prefix information as the Mobile Node received from the previous MAG. As stated before, the Mobile Node does not detect a link change and it keeps the same address(es). To make the change of link completely transparent to the Mobile Node, it must also continue receiving the Router Advertisements from the same link-local and link layer address; otherwise the Mobile Node would detect a change of default router. We describe how this problem is addressed in the next section.

Home Network Emulation and Address Uniqueness

MAGs must ensure that Mobile Nodes do not detect link changes when moving in a PMIPv6 domain; that is, MAGs must provide a home network emulation to the Mobile Nodes. To achieve this emulation, all the MAGs in the PMIPv6 domain must send, to a particular Mobile Node, Router Advertisements with the same network prefix information, as described previously. Additionally, the source IPv6 link-local address and the source link layer address in Router Advertisements sent to a Mobile Node must never change, independently of the MAG sending them. Therefore, the PMIPv6 specification requires all the MAGs to use, in any access link to which a particular Mobile Node attaches, the same link-local and link layer address.

PMIPv6 proposes two ways to meet this requirement:

- Configure a fixed link-local and link layer address to be used in all the access links in a PMIPv6 domain.
- Generate at the LMA the link-local address to be used by MAGs with a particular Mobile Node, and send it to the serving MAG through PMIPv6 signaling messages.

Both of these configuration methods are also helpful to guarantee address uniqueness in the access links of the PMIPv6 domain. The global addresses are always unique because all links are point-to-point and only one Mobile Node uses unicast global addresses over that link. Link-local addresses are used by the MAG and the Mobile Node on the link and a collision is possible. However, because the PMIPv6 specification requires that the link-local address used by the different MAGs with a particular Mobile Node is always the same while the Mobile Node moves across the PMIPv6 domain, the collision problem can happen only when the Mobile Node enters the PMIPv6 domain.

When a Mobile Node enters the domain, we must rely on *Duplicate Address Detection* (DAD) to detect a collision. If we use a globally unique link-local address for all the MAGs in the PMIPv6, then it is easy for the MAGs to respond to DAD requests from Mobile Nodes, because MAGs always know the address they must defend. If the link-local address to be used by the MAG with a Mobile Node is generated in the LMA, then it is desirable that the MAG learns that link-local address (that is, completes the PMIPv6 registration procedure) to defend it before the Mobile Node carries out the DAD procedure. You can ensure the MAG can learn this address by ensuring that the Layer 2 attachment is not completed until finishing the PMIPv6 signaling registration, or by configuring the PMIPv6 registration procedure in such a way that it is likely to be completed before the default waiting time of a DAD procedure.

Security Considerations

As with Mobile IPv6 signaling, PMIPv6 signaling is very sensitive to security threats, because it changes routing states of nodes in the network on behalf of the Mobile Nodes. PMIPv6 specification recommends using *IP Security* (IPsec) to protect the signaling exchanges between the MAGs and the LMA. A security association is needed between MAGs and the LMA, but how it is created is not defined. Two cases are possible:

- The network elements (LMA and MAGs) belong to the same operator.
- The elements belong to different operators with an agreement for roaming support.

In both scenarios, creating the security association is an affordable problem.

Traffic Handling in a PMIPv6 Domain

Traffic sent to any address belonging to a Home Network prefix is received by the LMA, the anchor point for those addresses. The LMA forwards the traffic through the tunnel to the MAG serving the Mobile Node, and the MAG decapsulates the packets and forwards them to the Mobile Node through the access link. Packets sent by the Mobile Node are forwarded by the MAG through the tunnel to the LMA. The LMA decapsulates the packets and forwards them to the destination. If a MAG has data traffic that originated in one of its access links and is destined to another of its access links, it can forward the traffic locally to avoid the forwarding through the LMA. This forwarding is done according to a policy configured in the MAG.

Performance Considerations

PMIPv6 presents two performance advantages compared with MIPv6. First, the LMA is a local network entity, so in principle the delay of sending signaling to the LMA will be lower than sending signaling to a remote Home Agent. And second, because the tunnel required to handle the traffic is terminated in the MAG instead of in the Mobile Node (as happens in MIPv6), we avoid the overhead of having a tunnel (two IP headers) over the radio interface. This overhead avoidance is relevant because bandwidth resources are scarcer over the air interface than in the backhaul network.

IPv4 Support Considerations

PMIPv6 acknowledges the existence of a dual-stack mobile host. To this end there are ongoing efforts to standardize IPv4 support for PMIPv6 operations. The extensions defined in [14] specify how to assign an IPv4 Home Address to a mobile host accessing the PMIPv6 domain. That is, the MAG—upon Mobile Node detection attachment and verification that the Mobile Node is eligible for PMIPv6 service—inserts in the PBU an "IPv4 Home Address Request Option."

The LMA, upon reception of the PBU message, assigns an IPv6 *Home Network Prefix* (HNP) or an IPv4 Home Address by attaching an "IPv4 Home Address Reply Option" to the PBA. How the information is delivered to the Mobile Node depends on the interface between the Mobile Node and the MAG, possible examples being DHCP or *Internet Key Exchange Version 2* (IKEv2). The Mobile Node—independent of the method deployed—configures the HNP and the IPv4 Home address assigned by the LMA, thus supporting both IPv4- and IPv6-based applications.

Conclusions

PMIPv6 is a promising specification that allows network operators to provide localized mobility support without relying on mobility functions or configuration present in the mobile nodes. This reality greatly eases the deployment of the solution.

The IETF is currently working in the *Network-Based Mobility Extensions* (netext) Working Group on extending the PMIPv6 specification to add functions such as enhanced multihoming and intertechnology handoff support, and localized routing for traffic between MAGs to avoid going through the LMA. Additionally, the *Multicast Mobility* (multimob) Working Group is working on the support of multicast in PMIPv6.

References

[1] S. Gundavelli (Ed.), K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6," RFC 5213, August 2008.

[2] Jon Postel, "Internet Protocol," RFC 791, September 1981.

[3] Stephen E. Deering and Robert M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1998.

[4] William Stallings, "Mobile IP," *The Internet Protocol Journal,* Volume 4, Number 2, June 2001.

[5] Charles E. Perkins, "IP Mobility Support for IPv4," RFC 3344, August 2002.

[6] David B. Johnson, Charles E. Perkins, and Jari Arkko, "Mobility Support in IPv6," RFC 3775, June 2004.

[7] E. Fogelstroem, A. Jonsson, and C. Perkins, "Mobile IPv4 Regional Registration," RFC 4857, June 2007.

[8] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, "Hierarchical Mobile IPv6 (HMIPv6) Mobility Management," RFC 5380, October 2008.

[9] J. Kempf (Ed.), "Problem Statement for Network-Based Localized Mobility Management (NETLMM)," RFC 4830, April 2007.

[10] J. Kempf (Ed.), "Goals for Network-Based Localized Mobility Management (NETLMM)," RFC 4831, April 2007.

[11] 3GPP TS 29.060, "GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface," 2009. Available at: **http://www.3gpp.org/ftp/Specs/html-info/29060.htm**

[12] Ignacio Soto, Carlos J. Bernardos, Maria Calderon, Albert Banchs, and Arturo Azcorra, "NEMO-Enabled Localized Mobility Support for Internet Access in Automotive Scenarios," *IEEE Communications Magazine,* Vol. 47, No. 5, May 2009.

[13] A. Patel, K. Leung, M. Khalil, H. Akhtar, and K. Chowdhury, "Mobile Node Identifier Option for Mobile IPv6 (MIPv6)," RFC 4283, November 2005.

[14] R. Wakikawa and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6," RFC 5844, May 2010.

[15] Carlos J. Bernardos, Ignacio Soto, and María Calderón, "IPv6 Network Mobility," *The Internet Protocol Journal,* Volume 10, Number 2, June 2007.

[16] Dave Meyer, "The Locator Identifier Separation Protocol (LISP)," *The Internet Protocol Journal,* Volume 11, Number 1, March 2008.

IGNACIO SOTO received a telecommunication engineering degree in 1993, and a Ph.D. in telecommunications in 2000, both from the University of Vigo, Spain. He was a research and teaching assistant in telematics engineering at the University of Valladolid from 1993 to 1999. In 1999 he joined University Carlos III of Madrid, where he was an associate professor from 2001 until 2010. In 2010, he joined Universidad Politécnica de Madrid as associate professor. His research activities focus on mobility support in packet networks and heterogeneous wireless access networks. E-mail: **isoto@dit.upm.es**

CARLOS J. BERNARDOS received a telecommunication engineering degree in 2003, and a Ph.D. in telematics in 2006, both from the University Carlos III of Madrid, where he worked as a research and teaching assistant from 2003 to 2008, and since then as an associate professor. His Ph.D. thesis focused on route optimization for mobile networks in IPv6 heterogeneous environments. He has published more than 30 scientific papers in prestigious international journals and conferences, and he also contributes to the IETF. He served as TPC chair of WEEDEV 2009 and as guest editor of *IEEE Network*. E-mail: **cjbc@it.uc3m.es**

MARÍA CALDERÓN is an associate professor at the Telematics Engineering Department of University Carlos III of Madrid. She received a computer science engineering degree in 1991 and a Ph.D. degree in computer science in 1996, both from the Technical University of Madrid. She has published more than 40 papers in the fields of advanced communications, reliable multicast protocols, programmable networks, and IPv6 mobility. E-mail: **maria@it.uc3m.es**

TELEMACO MELIA received his Informatics Engineering degree in 2002 from the Polytechnic of Turin, Italy, and his Ph.D. in Mobile Communications from the University of Goettingen in April 2007. From June 2002 to December 2007 he worked at NEC Europe Ltd. in Heidelberg, Germany, in the Mobile Internet Group. He worked on IPv6-based Mobile Communication focusing on IP mobility support across heterogeneous networks and resource optimization control. In September 2008 he joined Alcatel Lucent Bell Labs. He is currently working on interworking architectures spanning 3GPP, WiMAX forum, and IETF standardization bodies. His main research interests include wireless networking and next-generation networks. He his author of more than 20 publications and he actively contributes to the IETF. E-mail: **telemaco.melia@alcatel-lucent.com**

Improving User Experiences with IPv6 and SCTP

by Dan Wing and Andrew Yourtchenko, Cisco Systems

To be successful, new technologies must improve the user experience. In the process of finding the best way to deploy a new technology, several approaches are typically conceived, written down, tried, and possibly discarded. This article addresses two such approaches for *Internet Protocol Version 6* (IPv6) and the *Stream Control Transmission Protocol* (SCTP)[10].

Modern web browsers, web servers, and operating systems support IPv4 and IPv6, and several major content providers already support IPv6, including Google, NetFlix, and Facebook. However, their properties are not generally available over IPv6 because of a conflict between IPv6 technology and their business realities.

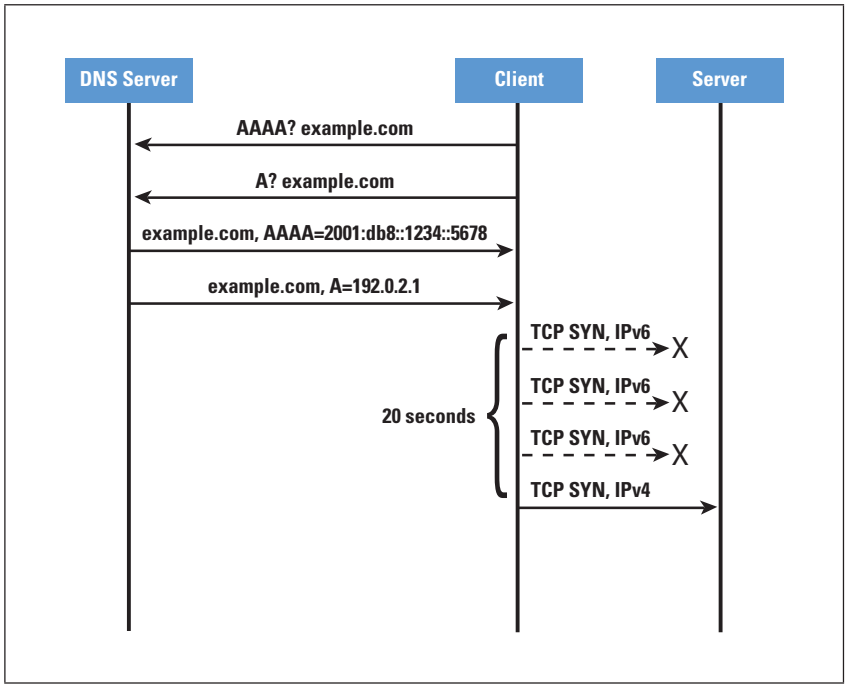
The technology in web browsers and operating systems involves doing *Domain Name System* (DNS) queries for *AAAA* and *A* resource records and then attempting to connect to the resulting IPv6 and IPv4 addresses *sequentially*. If the IPv6 path is broken (or slow), this connection can take a long time before it falls back to trying IPv4. This process is especially painful on typical websites that retrieve objects from different hosts—each failure incurs a delay. The combination of operating system and web browser results in delays from 20 seconds to several minutes if the IPv6 path is broken[2]. The typical message flow of a TCP client is shown in Figure 1. Clearly, this delay is unacceptable to users. Users avoid this delay by disabling IPv6[3] or avoiding IPv6-enabled websites.

The problem of broken IPv6 networks is relatively widespread[6]. Providing content is a business—either directly (for example, streaming movies) or indirectly (for example, selling advertising). If users suffer delays viewing IPv6-enabled content (because of the technology reasons described previously), they will have an incentive to visit other websites. This scenario means lost revenue and is unacceptable to the business. Considering that all of the customers on today's Internet can reach IPv4 content, it is a business risk to enable IPv6 because some customers will suffer delays attempting to view IPv6 websites. Major content providers have been monitoring the situation and have published results[7] showing that the IPv6 failure rate is too high to enable IPv6 *AAAA* for their content.

IPv6 problems have several causes. It is new technology, and monitoring of IPv6 connectivity is not yet on par with that of IPv4 because of single-point tunnels, unmanaged tunnels[11], accidentally misconfigured firewalls, and router and link failures can more easily cause outages on IPv6. Many applications remain IPv4-only, or network administrators are relying on dual-stack equipment to transparently fail over to IPv4 during IPv6 outages.

However, such failover is never transparent to users—it takes many seconds or minutes! To avoid these problems, the content provider has only one choice: don't provide *AAAA* records if users might experience broken or slow IPv6.

Figure 1: Behavior of a Typical Web Browser



To work around that problem, Google implements a white list of DNS servers that it will provide *AAAA* records for[8]. However, in its current incarnation, DNS white listing does not scale well because the *Internet Service Provider* (ISP) has to prove good IPv6 connectivity to Google, and then Google white lists the ISP's DNS servers to receive the *AAAA* records. The scaling problem is that there are thousands of ISPs around the world, and white listing and de-white listing them becomes a tiresome manual task for both ISPs and Google. Furthermore, if every content provider did DNS white listing, ISPs would have to work with several content providers in order to give value to the IPv6 network they have deployed to their subscribers! Content providers have started working together to consolidate requirements for DNS white listing and operate some sort of DNS white-listing service to slightly automate this process[5].

Yet, DNS white listing still does not guarantee a working IPv6 network or a fast IPv6 network, because there is not a direct relationship between good IPv6 connectivity and the DNS server of a user's ISP. Even with the best of intentions and network design, there will still be instances where an IPv6 path or IPv4 path is working when the other path is broken. The result will be excessive delays for IPv4-only clients or dual-stack clients, depending on what sort of breakage occurs.

This situation contributes to the user perception that the Internet, or the particular website being accessed, is "down." The user will visit a different site instead, possibly never returning to the site that was "down."

Happy Eyeballs

A different approach solves these problems. In this approach, rather than an application slowly trying to make a connection on IPv6 and then on IPv4, the application makes its connection attempts more aggressively over both IPv6 and IPv4. Initially, the connection attempts are made *simultaneously* (rather than serialized), in order to provide a fast user experience.

The simultaneous connection attempts consume a little extra network bandwidth and twice the connection attempts on the server. To reduce that chatter, a cache is also maintained to store the success or failure of connecting using IPv6 or IPv4. We nickname this approach "Happy Eyeballs"[1], because the "eyeballs" (users) are happier—their computer provides them immediate content, even if the network is suffering slow performance on IPv6 or IPv4 (Figure 2).

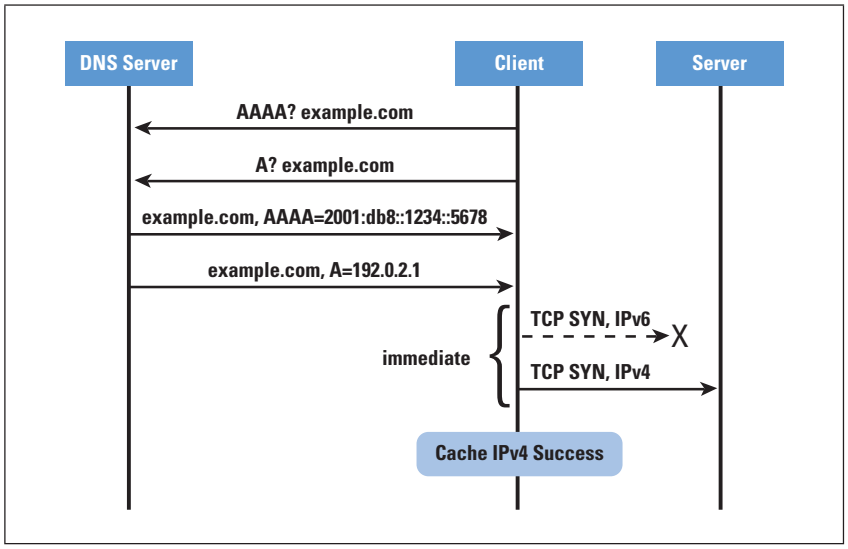


Figure 2: Dual-Stack Web Browser Implementing Happy Eyeballs

Obviously, sending a TCP SYN on both IPv6 and IPv4 doubles the number of connection attempts sent by the client. As discussed in [1], this chatter can be reduced by the application remembering if IPv6 (or IPv4) was successful in the previous connection attempt, and using that information for subsequent connection attempts. The sophistication of this cache is dependent on the memory (or disk) available, but even simple caching can be quite effective. When connecting to a new network (*third generation* [3G], different Wi-Fi network, or physical Ethernet), the connectivity of that new network can be determined and the cache of success or failure entirely or partially flushed, as necessary.

Thus, the doubling of connection attempts occurs only when connecting to a new network. Thereafter, initial connection attempts are delayed so that IPv6 (or IPv4) is tried first. But in all cases, significant user-noticeable delays are avoided when the IPv6 (or IPv4) is broken. The goal of Happy Eyeballs is to keep IPv6 enabled; that is, to make users unaware of IPv6 outages, so the user still visits IPv6-enabled websites without suffering any delay.

In this way, the user experiences a smooth migration from IPv4 to IPv6, and when necessary the fallback to IPv4 is almost immediate. This solution represents a significant improvement over today's web browsers. A drawback of this idea, however, is that it needs to be implemented in the application itself. Although it is a burden to upgrade those web browsers, there are only five major browsers[9], and the browsers receive the immediate benefit of the aggressive probing. Browsers are also commonly upgraded already for faster *JavaScript* engines and other new features.

Another idea to determine if IPv6 is working is to *ping* or send another simple request to an IPv6 resource on the Internet, and disable IPv6 on the host if that IPv6 request fails. This approach interferes with IPv6 traffic within the enterprise (which may be working fine, whereas IPv6 to the Internet is broken), and disabling IPv6 would break IPv6 features deployed in OSs (for example, *DirectAccess* in Windows or *Back to My Mac* in Mac OS X). An advantage of this approach is that if IPv6 is disabled, no application suffers the IPv6 outage and associated delay to fall back to IPv4.

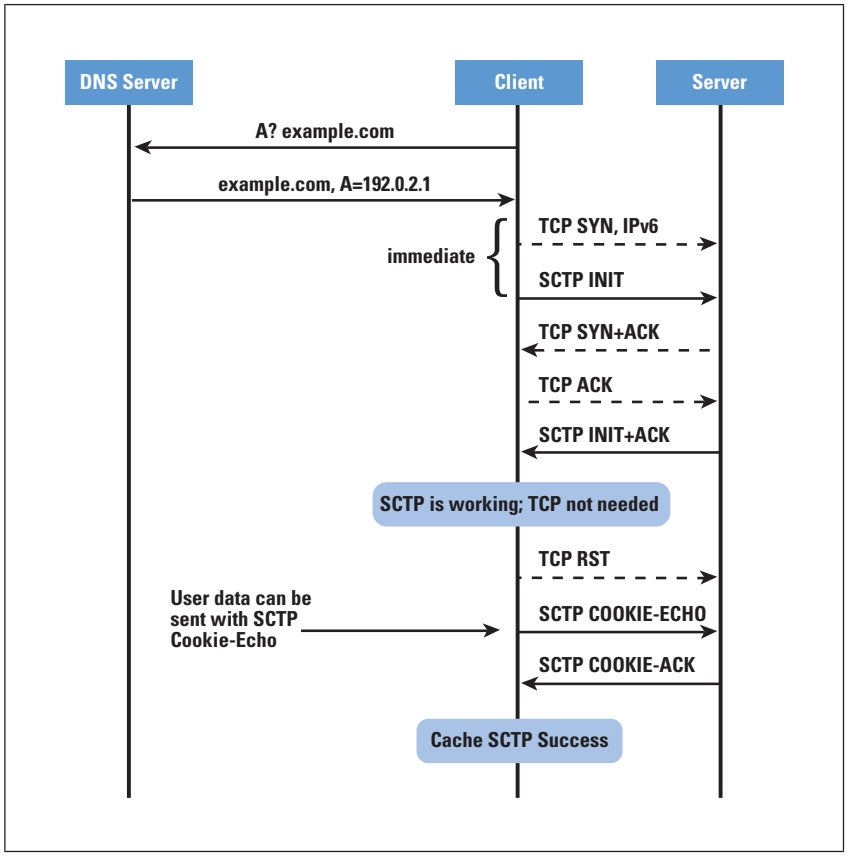
New Transport: SCTP

Besides the problem of network layer protocol selection, a similar task can be performed at the transport layer. Maybe surprisingly, one more transport protocol exists besides TCP, namely *Stream Control Transmission Protocol* (SCTP). SCTP provides significant advantages over TCP, and it was designed with some of the lessons learned by TCP implementations and deployment[4] in mind.

Unlike IPv6 and IPv4, which have different DNS resource records (*AAAA* and *A*), we don't have a resource record to indicate that an application could, or should, use a different transport protocol. But even if we could indicate support for SCTP in DNS, the path might block it, reducing the usefulness of a DNS resource record. The path could be blocked by a NAT or firewall that expects only TCP or *User Datagram Protocol* (UDP).

Happy Eyeballs also describes a technique where a client can simultaneously try connecting using both TCP and SCTP. By necessity, this attempt is done entirely in the application, and the application would prefer the transport that responded faster and cache that information to reduce network chatter for subsequent connections to that server. This scenario is shown in Figure 3.

Figure 3: Client Implementing Happy Eyeballs for TCP/SCTP Selection



By combining the IPv6/IPv4 technique with the SCTP/TCP technique, a web browser running on a computer connected to a new dual-stack network sends four packets—an IPv4 TCP SYN, an IPv6 TCP SYN, an IPv4 SCTP INIT, and an IPv6 SCTP INIT. Based on the responses, it decides which transport protocol and which address family (IPv6 or IPv4) it prefers, and abandons the other connections. As described previously, connection information is cached for subsequent use to avoid consuming network bandwidth and server resources for subsequent network connections.

Conclusion

New technology aimed at improving user experience will be successful only if it meets expectations—an improved user experience. Because many companies are deriving all of their revenue from the Internet, any reduction in service means a loss of revenue. Thus, deploying new technology must not negatively affect the user experience. This article described one of the mechanisms that implementers can use to avoid negative effects on the user experience.

References

[1] Dan Wing, Andrew Yourtchenko, and Preethi Natarajan, "Happy Eyeballs: Trending Towards Success (IPv6 and SCTP)," Internet-Draft, Work-in-Progress, July 2009: **http://tools.ietf.org/html/draft-wing-http-new-tech**

[2] "Broken IPv6 clients," Lorenzo Colitti, June 2010: **https://sites.google.com/site/ipv6implementors/2010/agenda**

[3] "Google Trends": **http://www.google.com/trends?q=enable+ipv6%2C+disable+ipv6**

[4] P. Natarajan, "Leveraging Innovative Transport Layer Services for Improved Application Performance," February 2009: **http://www.cis.udel.edu/~amer/PEL/poc/pdf/NatarajanPhDdissertation.pdf**

[5] Carolyn Duffy Marsan, "Google, Microsoft, Netflix in talks to create shared list of IPv6 users," *Network World,* March 2010: **http://www.networkworld.com/news/2010/032610-dns-ipv6-whitelist.html**

[6] Tore Anderson, "IPv6 brokenness experiment, November results," November 2009: **http://lists.cluenet.de/piper-mail/ipv6-ops/2009-December/002707.html**

[7] Igor Gashinsky, "IPv6 & recursive resolvers: How do we make the transition less painful?" March 2010: **http://www.ietf.org/proceedings/77/slides/dnsop-7.pdf**

[8] "Access Google services over IPv6": **http://www.google.com/intl/en/ipv6**

[9] "Usage share of web browsers": **http://en.wikipedia.org/wiki/Usage_share_of_web_browsers**

[10] R. Stewart, Ed., "Stream Control Transmission Protocol," RFC 4960, September 2007.

[11] Gunter Van de Velde, Ole Troan, and Tim Chown, "Non-Managed IPv6 Tunnels considered Harmful," July 2009: **http://tools.ietf.org/html/draft-vandevelde-v6ops-harmful-tunnels**

DAN WING has a B.S. in Computer Science from Central Washington University and has co-chaired the IETF's BEHAVE working group since 2006. He is a Distinguished Engineer at Cisco Systems, where he works on IPv6 transition technologies and has 30 patents issued or pending. E-mail: **dwing@cisco.com**

ANDREW YOURTCHENKO is a graduate of St. Petersburg Technical University in Russia, and has been in the networking industry since 1995. He is a Technical Leader at Cisco Systems in the network security area, and at IETF Andrew participates in the areas of security, TCP protocol, and IPv6 transition.
E-mail: **ayourtch@cisco.com**

Letter to the Editor

In response to "NAT++: Address Sharing in IPv4," in *The Internet Protocol Journal,* Volume 13, No. 2, June 2010:

Excellent article Geoff, so good I read it twice. While reading your article I was reminded of a recent experience that falls in the category of "unintended consequences." Since one of your situation descriptions was similar to the one I'm in, I thought I would relay my circumstance and experience and see if I can make my point.

A couple of months ago I signed up for an IPTV trial with my provider, and it was installed with a minimum of effort. The service is based on Cisco *Dial-on-Demand Routing* (DDR) and, of course, DSL service.

It worked fine for a couple of days; video feeds were good and all my computers and server worked just as before on a wireless network within my home. Then one day it appeared that I had lost *Domain Name System* (DNS) service, because I couldn't get name resolution to work but could route using the raw IPv4 addresses. So, I placed a trouble ticket and, of course, the provider's first request was to cold boot the DDR device and everything in the house, which I did. Sure enough, upon bringing all components back up (except one), everything was fine.

A day or two later I had to print something and powered on my HP 6510 wireless printer, printed what I needed to print, and then discovered I had lost DNS service again. I placed a trouble call and my provider came out and replaced the DDR device I went through the cold boot process (except one device) and everything was OK until I brought the printer online and the trouble returned. By now I had this nagging memory that wouldn't surface; something about that printer... With the printer powered off I rebooted the DDR, fired up SharkWire, and everything looked and worked OK.

Then I powered up the HP printer and saw another nagging memory; it immediately performed an *Address Resolution Protocol* (ARP) broadcast of the v4 address `169.254.65.206`—the famous black-hole address from RFC 3927[1]. Immediately after the ARP broadcast, the printer put out the normal *Dynamic Host Configuration Protocol* (DHCP) request and was assigned one from the *Network Address Translation* (NAT) pool.

That's when I stepped back from looking at the "trees" and gazed upon the "forest" and realized, with some embarrassment, that the public side (access side) was using a single IPv4 address with *Port Address Translation* (PAT) so the DDR box was blocking all the outbound PAT addresses attached to the single IPv4 address. I wrote down the details and e-mailed them to my provider, and had revised code pushed to the DDR the next day. Problem fixed.

All of this discussion leads me to ponder about other situations of "hard codes" in the network, either RFC-based or circumstance-based, that will falter with a switch to IPv6. Not in the core but in the customer networks. These unintended consequences could be many. Does HP run a dual stack for IPv4 and IPv6? I doubt it.

How can we get customers and vendors thinking about possible long-ago workarounds that they may have hard coded using IPv4? Any other RFCs out there like 3927? (It used to be easy when there were only a few hundred RFCs.) That could be the most expensive portion of the transition, verifying code ...

Keep up the good work; your articles make me think a lot and I really enjoy them. And, yes, I do use them for reference quite often.

Regards,

—Paul Dover
pdover@centeriem.com

[1] S. Cheshire, B. Aboba, and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses," RFC 3927, May 2005.

The author responds:

Thank you Paul for this anecdote and the important lesson behind it. Over some 30 years of intense development we've managed to accumulate a sizeable volume of technical specifications. Indeed, in October 2010 the RFC Editor published RFC 6068, and I'm not sure that any individual could claim a deep familiarity with every one of them, let alone claim to have a good understanding of their potential interaction. So when we look at various transitional technologies to sustain this industry through the next few years of attempting to support a comprehensive dual stack network in the face of the forthcoming hiatus of supply of IPv4 addresses, it should not come as a surprise when some devices or configurations fail in strange and unexpected ways, simply because they adhere to a technical standard that perhaps we've lost sight of in the flurry of generating new transitional technologies.

—Geoff Huston
gih@apnic.net

Fragments

Dr. Jianping Wu Receives Postel Award

The *Internet Society* (ISOC) recently awarded its prestigious *Jonathan B. Postel Service Award* for 2010 to leading Chinese technologist Dr. Jianping Wu for the pioneering role he has played in advancing Internet technology, deployment, and education in China and Asia Pacific over the last twenty years.

Dr. Wu's best-known contribution is the development of the *China Education and Research Network* (CERNET) which he designed and developed to be the first Internet backbone network in China. Created to establish a nation-wide advanced network infrastructure to support education and research among universities, CERNET has since become the world's largest national academic network. Since 1998, Dr. Wu has also devoted his time to the design and development of a large-scale native IPv6 backbone in China that now serves to connect over 200 universities and millions of users.

The Postel Award was established by the ISOC to honour individuals or organisations that, like Jon Postel, have made outstanding contributions in service to the data communications community. Commenting on its presentation to Dr. Wu, Lynn St. Amour, President and CEO of ISOC said: "Jianping Wu has dedicated his career in China to developing a broadly accessible Internet that brings people together. Twenty years ago, Dr. Wu recognized the importance and future impact of the Internet and the pivotal role it would play in terms of its impact on social reform, technology advancement and economic growth for China. He has worked tirelessly to bring his vision to life. As a result, the networks that resulted from his determination and hard work have played an important role in driving Internet development in China and have had a significant impact on the Internet worldwide."

ISOC presented the award, including a US$20,000 honorarium and a crystal engraved globe, during the 78th meeting of the *Internet Engineering Task Force* (IETF) in Maastricht, The Netherlands 25–30 July 2010.

DNSSEC Deployed in the Root Zone

On July 16, 2010 the U.S. Department of Commerce's *National Telecommunications and Information Administration* (NTIA) and the *National Institute of Standards and Technology* (NIST) announced the completion of an initiative with the *Internet Corporation for Assigned Names and Numbers* (ICANN) and VeriSign to enhance the security and stability of the Internet.

The announcement marks full deployment of a security technology—*Domain Name System Security Extensions* (DNSSEC)[1]—at the Internet's authoritative root zone, which will help protect Internet users against cache poisoning and other related cyber attacks.

"The Internet plays an increasingly vital role in daily life, from helping businesses expand to improving education and health care," said Assistant Secretary for Communications and Information and NTIA Administrator Lawrence E. Strickling. "The growth of the Internet is due in part to the trust of its users—trust, for example, that when they type a website address, they will be directed to their intended website. Today's action will help preserve that trust. It is an important milestone in the ongoing effort to increase Internet security and build a safer online environment for users."

"Improving the trustworthiness, robustness and scaling of the Internet's core infrastructure is an activity that lines up strongly with NIST's mission, and we have been contributing to design, standardization and deployment of DNSSEC technology for several years," said NIST Director Patrick Gallagher. "The deployment of DNSSEC at the root zone is the linchpin to facilitating its deployment throughout the world and enabling the current domain-name system to evolve into a significant new trust infrastructure for the Internet."

The *Domain Name System* (DNS) is a critical component of the Internet infrastructure. The DNS associates user-friendly domain names (for example, `www.commerce.gov`) with the numeric network addresses (for example, `170.110.225.168`) required to deliver information on the Internet, making the Internet easier for the public to navigate. The authenticity of the DNS data is essential to Internet use. For example, it is vital that users reach their intended destinations on the Internet and are not unknowingly redirected to bogus and malicious websites.

The DNS was not originally designed with strong security mechanisms, and technological advances have made it easier to exploit vulnerabilities in the DNS protocol that put the integrity of DNS data at risk. Many of these vulnerabilities are mitigated by the deployment of DNSSEC, which is a suite of *Internet Engineering Task Force* (IETF) specifications for securing information provided by the DNS.

A main goal of this action—DNSSEC deployment at the root zone—is to facilitate greater DNSSEC deployment throughout the rest of the global DNS hierarchy. While deployment of DNSSEC will protect Internet users from certain DNS-related cyber attacks, users must continue to exercise vigilance in protecting their information online.

ISOC Embraces DNSSEC

The *Internet Society* (ISOC) recently announced that it has deployed DNSSEC, a set of extensions to the DNS that provides a level of assurance, for its `isoc.org` domain. The announcement builds on an announcement by the *Public Interest Registry* (PIR) that they have implemented DNSSEC for the entire `.org` top-level domain.

"We are pleased to be among the first organisations in the `.org` top level domain to deploy DNSSEC, as DNSSEC provides an important building block for increasing user confidence in the Internet," said Lynn St.Amour, President and CEO of the Internet Society. "Implementing DNSSEC for the `.org` top-level domain is an important step in ensuring the global Internet serves as a trusted channel for communication and collaboration and we applaud the PIR's efforts in this area."

"DNSSEC acts like tamper-proof packaging to make sure that when you type in the website name of your bank you actually get the server IP address your bank wants you to use," said Leslie Daigle, Chief Internet Technology Officer of ISOC. "In this way, DNSSEC allows us to have more confidence in the online activities that are increasingly becoming a part of our lives at work, home, and school."

DNSSEC technology used today is the result of careful protocol engineering and standardization within the IETF; implementation by various DNS vendors; and operational trials by DNS operators. In addition to `.org`, DNSSEC is currently implemented by several country-specific top-level domains: Brazil (`.br`), Bulgaria (`.bg`), The Czech Republic (`.cz`), Puerto Rico (`.pr`), and Sweden (`.se`).

ISOC is a non-profit organisation founded in 1992 to provide leadership in Internet related standards, education, and policy. ISOC is the organisational home of the IETF. With offices in Washington, D.C., and Geneva, Switzerland, it is dedicated to ensuring the open development, evolution, and use of the Internet for the benefit of people throughout the world. For more information see: `http://isoc.org`

DNSSEC Fund Announced

In order to speed up the process of introduction a more secure global DNS infrastructure, the Netherlands-based charity *NLnet Foundation* has announced the creation of a global fund where open source projects can apply for grants to work on *Domain Name System Security Extensions* (DNSSEC) in their Internet applications.

DNSSEC is one of the key technologies for a safer Internet, as it allows the Internet user to know for sure that he or she is being sent to the right computer or service on the Internet. "If you type the name of your bank into a browser, you want to be sure that you are actually directed to a computer of that bank," said Michiel Leenaars, Director of Strategy at NLnet foundation. "Domain names are vital to the way we use the Internet, and without DNSSEC users are open to serious abuse."

DNSSEC provides a cryptographic seal of authenticity that gives real proof of the validity of the domain name you use when you visit a website, chat or send an e-mail. With DNSSEC you get a *chain of trust* from the root of the Internet to the service you want to connect to—opening the way for many new exciting opportunities for humans and computers to exchange information safely. DNSSEC is being gradually introduced worldwide.

The new fund will provide grants for reengineering important software to reliably work with DNSSEC. "The signing of the root through DNSSEC is a historical moment, but in a way it is only the beginning," said Leslie Daigle, Chief Internet Technology Office at the Internet Society. "Actual users will not fully benefit from protection in the more challenging situations as long as DNSSEC does not reach them." A great deal of work has already been done at the infrastructure level—most DNS servers such as *BIND, NSD* and *Unbound* now support the new technology. However, it will take a lot of work at the user level as well: operating systems, web browsers, e-mail servers, VoIP clients, and many other pieces of software need to be able to reliably work with DNSSEC.

"Every Internet user deserves to be protected by DNSSEC, yet currently almost no end user software is ready to take full advantage of the availability of DNSSEC," said Leenaars. "The IT community has a big responsibility in making sure that DNSSEC gets deployed across the board swiftly. We aim to accelerate the process significantly by putting some money on the table, and we invite other stakeholders to join us."

Since there are many applications and platforms that will require work, the NLnet Foundation is very open to cooperation with others as well as to targeted donations from interested stakeholders such as governments, registries and corporations.

The NLnet Foundation is a registered Netherlands charity with a long history of supporting Internet standardization. The foundation gained its capital from selling the first Dutch Internet Service Provider.

Potential applicants and collaborators can find more information at: **http://nlnet.nl/dnssec**

See also:

[1] Miek Gieben, "DNSSEC: The Protocol, Deployment, and a Bit of Development," *The Internet Protocol Journal,* Volume 7, No. 2, June 2004.

[2] Torbjörn Eklöv, and Stephan Lagerholm, "Operational Challenges when Implementing DNSSEC," *The Internet Protocol Journal,* Volume 13, No. 2, June 2010.

[3] **http://www.dnssec.net/**

Call for Papers: Internet Privacy Workshop

The *Internet Architecture Board* (IAB), *World Wide Web Consortium* (W3C), *Internet Society* (ISOC) and the *Massachusetts Institute of Technology* (MIT) will hold a joint *Internet Privacy Workshop* on December 8 and 9, 2010 at MIT, Cambridge, Massachusetts on the question:

"How Can Technology Help to Improve Privacy on the Internet?"

Information about who we are, what we own, what we have experienced, how we behave, where we are located, and how we can be reached are among the most personal pieces of information about us. This information is increasingly being made more easily available electronically via the Internet, often without the consent of the subject. The question for the workshop therefore is: How can we ensure that architectures and technologies for the Internet, including the World Wide Web, are developed in ways that respects users' intentions about their privacy?

This workshop aims to explore the experience and approaches taken by developers of Internet including Web technology, when designing privacy into these protocols and architectures. Engineers know that many design considerations need to be taken into account when developing solutions. Balancing between the conflicting goals of openness, privacy, economics, and security is often difficult, as illustrated by Clark, et al. in "Tussle in Cyberspace: Defining Tomorrow's Internet," see:

http://groups.csail.mit.edu/ana/Publications/PubPDFs/Tussle2002.pdf

As a member of the technical community, we invite you to share your experiences by participating in this important workshop. Workshop participants will focus on the core privacy challenges, the approaches taken to deal with them, and the status of the work in the field. The objective is to draw a relationship with other application areas and other privacy work in an effort to discuss how specific approaches can be generalized.

Interested parties must submit a brief contribution describing their work or approach as it relates to the workshop theme. We welcome visionary ideas for how to tackle Internet privacy problems, as well as write-ups of existing concepts, deployed technologies, and lessons-learned from successful or failed attempts at deploying privacy technologies. Contributions are not required to be original in content.

Submitters of accepted position papers will be invited to the workshop. The workshop will be structured as a series of working sessions, punctuated by invited speakers, who will present relevant background information or controversial ideas that will motivate participants to reach a deeper understanding of the subject.

The organizing committee may ask submitters of particularly topical papers to present their ideas and experiences to the workshop. We will publish submitted position papers and slides together with a summary report of the workshop. There are no plans for any remote participation in this workshop.

To be invited to the workshop, please submit position papers to **privacy@iab.org** by November 5, 2010. More detailed information about the workshop, including further details about the position paper requirements, is available at:

http://www.iab.org/about/workshops/privacy/

We look forward to your input,

Bernard Aboba (IAB)
Daniel Appelquist (W3C)
Jon Peterson (IAB)
Karen Sollins (MIT)
Trent Adams (ISOC)
Karen O'Donoghue (ISOC)
Thomas Roessler (W3C)
Hannes Tschofenig (IAB)

Organizations Urged to Stop Delaying IPv6 Deployment

The *Number Resource Organization* (NRO), the official representative of the five *Regional Internet Registries* (RIRs) that oversee the allocation of all Internet number resources, recently unveiled the findings of a global, independent survey into organizations' IPv6 readiness. Funded by the European Commission and conducted by GNKS Consult and TNO, the study reveals that the majority of organizations are taking steps toward IPv6 deployment, as the IPv4 address pool continues to deplete rapidly.

IP addresses are critical for the operation of the Internet. Every Internet-enabled device needs an IP address to connect to the rest of the network. The biggest threat facing the Internet today is that less than 6% of the current form of IP addresses, IPv4, remains and the pool is likely to be completely depleted next year. This means that organizations need to adopt IPv6, the next-generation addressing protocol. There is a far larger pool of IPv6 addresses, allowing for more devices to connect to the Internet and helping to safeguard the sustainable growth of the Internet.

The survey, which polled over 1,500 organizations from 140 countries, highlights that organizations are increasingly aware of the need to deploy IPv6: approximately 84% already have IPv6 addresses or have considered requesting them from the RIRs. Only 16% of respondents have no plans to deploy IPv6 addresses.

The study also demonstrates that there are some misconceptions around the cost of adopting IPv6. Over half of all respondents noted that the cost of deployment was a major barrier for IPv6 adoption. While organizations might delay investing in IPv6, this may ultimately result in greater costs, with last-minute deployment and poor planning likely to increase the investment required.

Of the 84% of respondents that have requested IPv6 addresses or have considered doing so, three-quarters reported the need to stay ahead of competition as the main reason for IPv6 adoption. Half of these respondents also noted that a lack of available IPv4 space was a major driver for deployment. When asked about issues they had encountered when deploying IPv6:

- 60% cited the lack of vendor support as a major barrier for deployment. However, most of the latest hardware and software support IPv6. The RIRs are strongly urging organizations to check with their suppliers to ensure that the technologies they use are IPv6 compatible.
- 45% reported a struggle to find knowledgeable technical staff to support deployment. However, all five RIRs arrange technical training to facilitate an efficient IPv6 deployment, details of which can be accessed via the NRO website.

Fifty-eight percent of all organizations polled were ISPs. It is likely that respondents to this survey are further ahead in IPv6 deployment than ISPs overall, but all organizations should ensure that their ISP offers or plans to offer services over IPv6. Out of the polled ISPs:

- Approximately 60% already offer, or plan to offer within the next year, IPv6 to consumers.
- 70% already offer, or plan to offer within the next year, IPv6 to businesses.
- Only about 10% of polled ISPs have no plans to offer IPv6 to consumers or businesses.

Axel Pawlik, Chairman of the NRO, commented: "It's great to see that as we move toward complete IPv4 exhaustion, more organizations worldwide are waking up to the need to adopt IPv6 and are sourcing IPv6 addresses from the RIRs."

"Yet there is still a distinct lack of Internet traffic over the next addressing protocol, with not enough ISPs offering IPv6 services and 30% of ISPs saying the proportion of this traffic is less than 0.5%. It's critical that ISPs now take the next step in the global adoption effort by offering IPv6 services to their customers to help boost traffic over IPv6."

Per Blixt, Head of Unit in the Information Society and Medias at the European Commission, said:

"It's encouraging to see that so many organizations have made IPv6 adoption their priority. Still, as the Internet becomes increasingly important for global socio-economic development, it's critical that those who are still sitting on the fence act now on IPv6. Only by ensuring that all organizations adopt IPv6 can we ensure the sustainable growth of the digital economy worldwide."

This survey is a follow-up to a study conducted in 2009 amongst organizations in Europe, Middle East and parts of Central Asia, as well as Asia Pacific; however this year's survey polled organizations worldwide. The full research report is available at:

http://www.nro.net/documents/GlobalIPv6SurveySummaryv2.pdf

The NRO exists to protect the pool of unallocated Internet numbers (IP addresses and AS numbers) and serves as a coordinating mechanism for the five RIRs to act collectively on matters relating to the interests of RIRs. For further information, visit **http://www.nro.net**

The RIRs are independent, not-for-profit membership organizations that support the infrastructure of the Internet through technical coordination. There are five RIRs in the world today. Currently, the *Internet Assigned Numbers Association* (IANA) allocates blocks of IP addresses and ASNs, known collectively as *Internet Number Resources,* to the RIRs, who then distribute them to their members within their own specific service regions. RIR members include *Internet Service Providers* (ISPs), telecommunications organizations, large corporations, governments, academic institutions, and industry stakeholders, including end users

The RIR model of open, transparent participation has proven successful at responding to the rapidly changing Internet environment. Each RIR holds one to two open meetings per year, as well as facilitating online discussion by the community, to allow the open exchange of ideas from the technical community, the business sector, civil society, and government regulators. Each RIR performs a range of critical functions including: The reliable and stable allocation of Internet number resources (IPv4, IPv6 and *Autonymous System Number* resources); The responsible storage and maintenance of this registration data; The provision of an open, publicly accessible database where this data can be accessed. RIRs also provide a range of technical and coordination services for the Internet community. The five RIRs are:

AfriNIC: **http://www.afrinic.net**

APNIC: **http://www.apnic.net**

ARIN: **http://www.arin.net**

LACNIC: **http://www.lacnic.net**

RIPE NCC: **http://www.ripe.net**

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

December 2010 Volume 13, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

ISSN 1944-1134

From The Editor

I have recently started using both a smartphone and a tablet device for Internet access. Like millions of other Internet users, I have discovered the wonders of mobile applications that provide everything from the traditional Internet services (e-mail and web browsing) to specialized software that can pinpoint my location on a map, provide live currency-exchange calculations, give weather forecasts, and my favorite: play radio stations from all over the world. I am old enough to remember the orange glow from pre-transistor vacuum-tube radios, so having a customizable "world radio" in the form of an "app" on a smartphone seems almost like science fiction.

But radio is not the only traditional service that is now available over the Internet. Another prominent example is telephony or *Voice over IP* (VoIP). Not only is VoIP replacing traditional land lines in many places, the original circuit-switched telephone network is itself increasingly using VoIP technology in place of an infrastructure of land lines and dedicated switching equipment. An important aspect of traditional phone service is the notion of special numbers for *emergency services*. Such systems rely on a database of phone numbers and addresses that allow emergency personnel to dispatch responders to the correct location. This location identification becomes a lot more complicated if the caller is using an Internet-based calling service rather than a hard-wired telephone. The IETF has been tackling this problem in the *Emergency Context Resolution with Internet Technology* (ECRIT) working group. Our first article, by Hannes Tschofenig and Henning Schulzrinne, is an overview of the architecture this working group is developing.

According to the ITU-T, a *Next Generation Network* (NGN) is "...a packet-based network which can provide services including Telecommunication Services and is able to make use of multiple broadband, Quality of Service-enabled transport technologies in which service-related functions are independent from underlying transport-related technologies." Paul Veitch, Paul Hitchen, and Martin Mitchell describe the integration of a standalone core BGP/MPLS VPN network into an NGN architecture.

Please check your subscription expiration date and renew online if you wish to continue receiving this journal. Click the "Subscriber Services" link at **www.cisco.com/ipj** to get to the login page. If you need any assitance just send e-mail to **ipj@cisco.com** and we will make the necessary changes for you.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Emergency Services for Internet Multimedia

by Hannes Tschofenig, Nokia Siemens Networks and Henning Schulzrinne, Columbia University

Summoning the police, the fire department, or an ambulance in emergencies is one of the most important functions the telephone enables. As telephone functions move from circuit-switched to Internet telephony, telephone users rightfully expect that this core feature will continue to be available and work as well as it has in the past. Users also expect to be able to reach emergency assistance using new communication devices and applications, such as instant messaging or *Short Message Service* (SMS), and new media, such as video. In all cases, the basic objective is the same: The person seeking help needs to be connected with the most appropriate *Public Safety Answering Point* (PSAP), where call takers dispatch assistance to the caller's location. PSAPs are responsible for a particular geographic region, which can be as small as a single university campus or as large as a country.

The transition to Internet-based emergency services introduces two major structural challenges. First, whereas traditional emergency calling imposed no requirements on end systems and was regulated at the national level, Internet-based emergency calling needs global standards, particularly for end systems. In the old *Public Switched Telephone Network* (PSTN), each caller used a single entity, the landline or mobile carrier, to obtain services. For Internet multimedia services, network-level transport and applications can be separated, with the *Internet Service Provider* (ISP) providing IP connectivity service, and a *Voice Service Provider* (VSP) adding call routing and PSTN termination services. We ignore the potential separation between the Internet access provider, that is, a carrier that provides physical and data link layer network connectivity to its customers, and the ISP that provides network layer services. We use the term VSP for simplicity, instead of the more generic term *Application Server Provider* (ASP).

The documents that the IETF *Emergency Context Resolution with Internet Technology* (ECRIT) working group is developing support multimedia-based emergency services, and not just voice. As is explained in more detail later in this article, emergency calls need to be identified for special call routing and handling services, and they need to carry the location of the caller for routing and dispatch. Only the calling device can reliably recognize emergency calls, while only the ISP typically has access to the current geographical location of the calling device based on its point of attachment to the network. The reliable handling of emergency calls is further complicated by the wide variety of access technologies in use, such as *Virtual Private Networks* (VPNs), other forms of tunneling, firewalls, and *Network Address Translators* (NATs).

This article describes the architecture of emergency services as defined by the IETF and some of the intermediate steps as end systems and the call-handling infrastructure transition from the current circuit-switched and emergency-calling-unaware *Voice-over-IP* (VoIP) systems to a true any-media, any-device emergency calling system.

IETF Emergency Services Architecture

The emergency services architecture developed by the IETF ECRIT working group is described in [1] and can be summarized as follows: *Emergency calls are generally handled like regular multimedia calls, except for call routing*. The ECRIT architecture assumes that PSAPs are connected to an IP network and support the *Session Initiation Protocol* (SIP)[2] for call setup and messaging. However, the calling user agent may use any call signaling or instant messaging protocol, which the VSP then translates into SIP.

Nonemergency calls are routed by a VSP, either to another subscriber of the VSP, typically through some SIP session border controller or proxy, or to a PSTN gateway. For emergency calls, the VSP keeps its call routing role, routing calls to the emergency service system to reach a PSAP instead. However, we also want to allow callers that do not subscribe to a VSP to reach a PSAP, using nothing but a standard SIP[2] user agent (see [3] and [4] for a discussion about this topic); the same mechanisms described here apply. Because the Internet is global, it is possible that a caller's VSP resides in a regulatory jurisdiction other than where the caller and the PSAP are located. In such circumstances it may be desirable to exclude the VSP and provide a direct signaling path between the caller and the emergency network. This setup has the advantage of ensuring that all parties included in the call delivery process reside in the same regulatory jurisdiction.

As noted in the introduction, the architecture neither forces nor assumes any type of trust or business relationship between the ISP and the VSP carrying the emergency call. In particular, this design assumption affects how location is derived and transported.

Providing emergency services requires three crucial steps, which we describe in the following sections: recognizing an emergency call, determining the caller's location, and routing the call and location information to the appropriate emergency service system operating a PSAP.

Recognizing an Emergency Call

In the early days of PSTN-based emergency calling, callers would dial a local number for the fire or police department. It was recognized in the 1960s that trying to find this number in an emergency caused unacceptable delays; thus, most countries have been introducing single nationwide emergency numbers, such as 911 in North America, 999 in The United Kingdom, and 112 in all European Union countries.

This standardization became even more important as mobile devices started to supplant landline phones. In some countries, different types of emergency services, such as police or mountain rescue, are identified by separate numbers. Unfortunately, more than 60 different emergency numbers are used worldwide, many of which also have nonemergency uses in other countries, so simply storing the list of numbers in all devices is not feasible. In addition, hotels and university campuses often use dial prefixes, so an emergency caller in some European universities may actually have to dial 0112 to reach the fire department.

Because of this diversity, the ECRIT architecture decided to separate the concept of an emergency dial string, which remains the familiar and regionally defined emergency number, and a protocol identifier that is used for identifying emergency calls within the signaling system. The calling end system has to recognize the emergency (service) dial string and translate it into an emergency service identifier, which is an extensible set of *Uniform Resource Names* (URNs) defined in RFC 5031[5]. A common example for such a URN, defined to reach the generic emergency service, is `urn:service.sos`. The emergency service URN is included in the signaling request as the destination and is used to identify the call as an emergency call. If the end system fails to recognize the emergency dial string, the VSP may also perform this service.

Because mobile devices may be sold and used worldwide, we want to avoid manually configuring emergency dial strings. In general, a device should recognize the emergency dial string familiar to the user and the dial strings customarily used in the currently visited country. The *Location-to-Service Translation Protocol* (LoST)[6], described in more detail later, also delivers this information.

Some devices, such as smartphones, can define dedicated user interface elements that dial emergency services. However, such mechanisms must be carefully designed so that they are not accidentally triggered, for example, when the device is in a pocket.

Emergency Call Routing

When an emergency call is recognized, the call needs to be routed to the appropriate PSAP. Each PSAP is responsible for only a limited geographic region, its service region, and some set of emergency services. For example, even in countries with a single general emergency number such as the United States, poison-control services maintain their own set of call centers. Because VSPs and end devices cannot keep a complete up-to-date mapping of all the service regions, a mapping protocol, LoST[6], maps a location and service URN to a specific PSAP *Uniform Resource Identifier* (URI) and a service region.

LoST, illustrated in Figure 1, is a *Hypertext Transfer Protocol* (HTTP)-based query/response protocol where a client sends a request containing the location information and service URN to a server and receives a response containing the service URL, typically a SIP URL, the service region where the same information would be returned, and an indication of how long the information is valid. Both request and response are formatted as *Extensible Markup Language* (XML). For efficiency, responses are cached, because otherwise every small movement would trigger a new LoST request. As long as the client remains in the same service region, it does not need to consult the server again until the response returned reaches its expiration date. The response may also indicate that only a more generic emergency service is offered for this region. For example, a request for `urn:service:sos.marine` in Austria may be replaced by `urn:service:sos`. Finally, the response also indicates the emergency number and dial string for the respective service.

The number of PSAPs serving a country varies significantly. Sweden, for example, has 18 PSAPs, and the United States has approximately 6,200. Therefore, there is roughly one PSAP per 500,000 inhabitants in Sweden and one per 50,000 in the United States. As all-IP infrastructure is rolled out, smaller PSAPs may be consolidated into regional PSAPs. Routing may also take place in multiple stages, with the call being directed to an *Emergency Services Routing Proxy* (ESRP), which in turn routes the call to a PSAP, accounting for factors such as the number of available call takers or the language capabilities of the call takers.

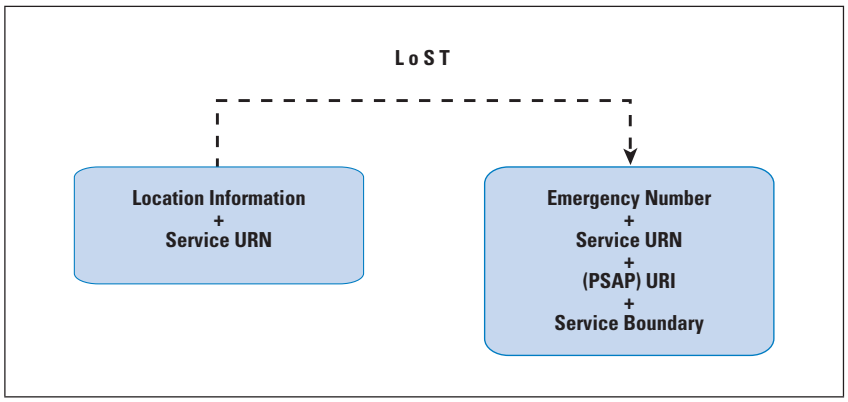


Figure 1: High-Level Functions of Location-to-Service Translation (LoST) Protocol

Location Information

Emergency services need location information for three reasons: routing the call to the right PSAP, dispatching first responders (for example, policemen), and determining the right emergency service dial strings. It is clear that the location must be automatic for the first and third applications, but experience has shown that automated, highly accurate location information is vital to dispatching as well, rather than relying on callers to report their locations to the call taker.

Such information increases accuracy and avoids dispatch delays when callers are unable to provide location information because of language barriers, lack of familiarity with their surroundings, stress, or physical or mental impairment.

Location information for emergency purposes comes in two representations: geo(detic), that is, longitude and latitude, and civic, that is, street addresses similar to postal addresses. Particularly for indoor location, vertical information (floors) is very useful. Civic locations are most useful for fixed Internet access, including wireless hotspots, and are often preferable for specifying indoor locations, whereas geodetic location is frequently used for cell phones. However, with the advent of femto and pico cells, civic location is both possible and probably preferable because accurate geodetic information can be very hard to acquire indoors.

In almost all cases, location values are represented as *Presence Information Data Format Location Object* (PIDF-LO), an XML-based document to encapsulate civic and geodetic location information. The format of PIDF-LO is described in [7], with the civic location format updated in [8] and the geodetic location format profiled in [9]. The latter document uses the *Geography Markup Language* (GML) developed by the *Open Geospatial Consortium* (OGC) for describing commonly used location shapes.

Location can be conveyed either by value ("LbyV") or by reference ("LbyR"). For the former, the XML location object is added as a message body in the SIP message. Location by value is particularly appropriate if the end system has access to the location information; for example, if it contains a *Global Positioning System* (GPS) receiver or uses one of the location configuration mechanisms described later in this section. In environments where the end host location changes frequently, the LbyR mechanism might be more appropriate. In this case, the LbyR is an HTTP/*Secure HTTP* (HTTPS) or SIP/*Secure SIP* (SIPS) URI, which the recipient needs to resolve to obtain the current location. Terminology and requirements for the LbyR mechanism are available in [10].

An LbyV and an LbyR can be obtained through location configuration protocols, such as the *HTTP Enabled Location Delivery* (HELD) protocol[11] or *Dynamic Host Configuration Protocol* (DHCP)[12, 13]. When obtained, location information is required for LoST queries, and that information is added to SIP messages[14].

The requirements for location accuracy differ between routing and dispatch. For call routing, city or even county-level accuracy is often sufficient, depending on how large the PSAP service areas are, whereas first responders benefit greatly when they can pinpoint the caller to a particular building or, better yet, apartment or office for indoor locations, and an outdoor area of at most a few hundred meters. This detailed location information avoids having to search multiple buildings, for example, for medical emergencies.

As mentioned previously, the ISP is the source of the most accurate and dependable location information, except for cases where the calling device has built-in location capabilities, such as GPS, when it may have more accurate location information. For landline Internet connections such as DSL, cable, or fiber-to-the-home, the ISP knows the provisioned location for the network termination, for example. The IETF GEOPRIV working group has developed protocol mechanisms, called *Location Configuration Protocols,* so that the end host can request and receive location information from the ISP. The Best Current Practice document for emergency calling[15] enumerates three options that clients should universally support: DHCP civic[16] and geo[12] (with a revision of RFC 3825 in progress[17]), and HELD[11]. HELD uses XML query and response objects carried in HTTP exchanges. DHCP does not use the PIDF-LO format, but rather more compact binary representations of locations that require the endpoint to construct the PIDF-LO.

Particularly for cases where end systems are not location-capable, a VSP may need to obtain location information on behalf of the end host[18].

Obtaining at least approximate location information at the time of the call is time-critical, because the LoST query can be initiated only after the calling device or VSP has obtained location information. Also, to accelerate response, it is desirable to transmit this location information with the initial call signaling message. In some cases, however, location information at call setup time is imprecise. For example, a mobile device typically needs 15 to 20 seconds to get an accurate GPS location "fix," and the initial location report is based on the cell tower and sector. For such calls, the PSAP should be able to request more accurate location information either from the mobile device directly or the *Location Information Server* (LIS) operated by the ISP. The SIP event notification extension, defined in RFC 3265[19], is one such mechanism that allows a PSAP to obtain the location from an LIS. To ensure that the PSAP is informed only of pertinent location changes and that the number of notifications is kept to a minimum, event filters[20] can be used.

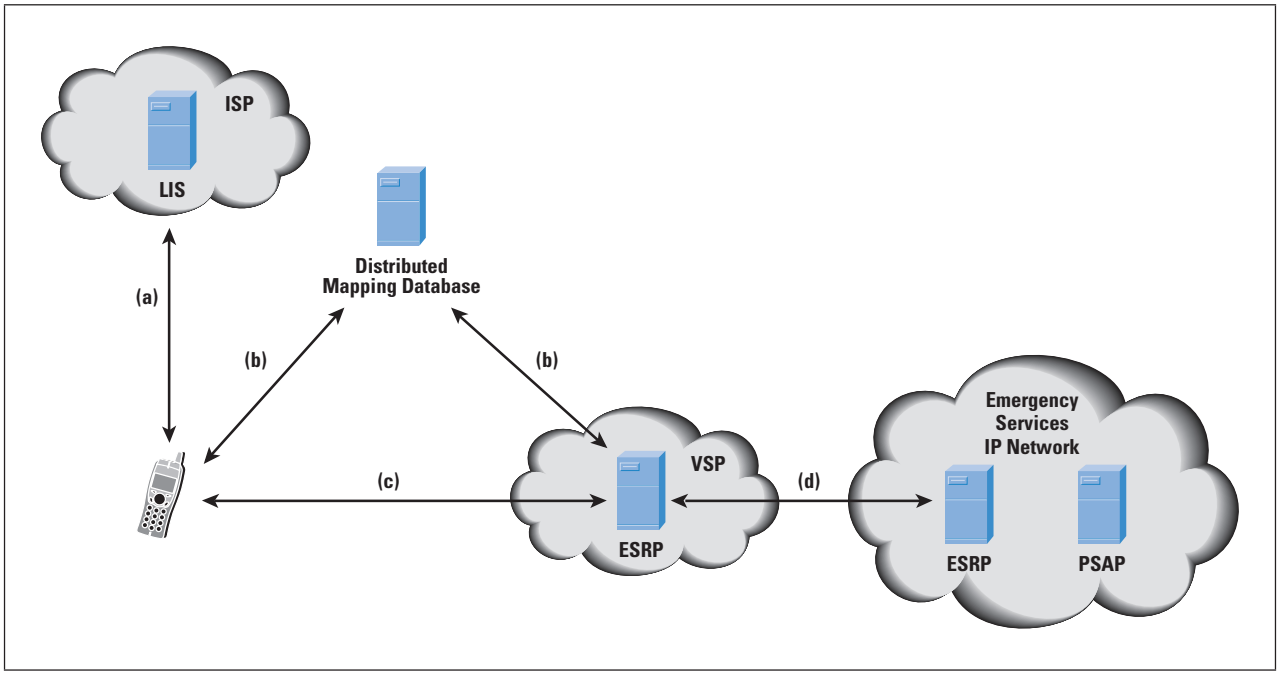
The two-stage location refinement mechanism described previously works best when location is provided by reference (LbyR) in the SIP INVITE call setup request. The PSAP subscribes to the LbyR provided in the SIP exchange and the LbyR refers to the LIS in the ISP's network. In addition to a SIP URI, the LbyR message can also contain an HTTP/HTTPS URI. When such a URI is provided, an HTTP-based protocol can be used to retrieve the current location[21].

Obligations

This section discusses the requirements the different entities need to satisfy, based on Figure 2. A more detailed description can be found in [15].

Note that this narration focuses on the final stage of deployment and does not discuss the transition architecture, in which some implementation responsibilities can be rearranged, with an effect on the overall functions offered by the emergency services architecture. A few variations were introduced to handle the transition from the current system to a fully developed ECRIT architecture.

Figure 2: Main Components Involved in an Emergency Call



With the work on the IETF emergency architecture, we have tried to balance the responsibilities among the participants, as described in the following sections.

End Hosts

An end host, through its VoIP application, has three main responsibilities: it has to attempt to obtain its own location, determine the URI of the appropriate PSAP for that location, and recognize when the user places an emergency call by examining the dial string. The end host operating system may assist in determining the device location.

The protocol interaction for location configuration is indicated as interface (a) in Figure 2; numerous location configuration protocols have been developed to provide this capability.

A VoIP application needs to support the LoST protocol[6] in order to determine the emergency service dial strings and the PSAP URI. Additionally, the device needs to understand the service identifiers, defined in [5].

As currently defined, it is assumed that SIP can reach PSAPs, but PSAPs may support other signaling protocols, either directly or through a protocol translation gateway. The LoST retrieval results indicate whether other signaling protocols are supported. To provide support for multimedia, use of different types of codecs may be required; details are available in [15].

ISP

The ISP has to make location information available to the endpoint through one or more of the location configuration protocols.

In order to route an emergency call correctly to a PSAP, an ISP may initially disclose the approximate location for routing to the endpoint and give more precise location information later, when the PSAP operator dispatches emergency personnel. The functions required by the IETF emergency services architecture are restricted to the disclosure of a relatively small amount of location information, as discussed in [22] and in [23].

The ISP may also operate a (caching) LoST server to improve the robustness and reliability of the architecture. This server lowers the round-trip time for contacting a LoST server, and the caches are most likely to hold the mappings of the area where the emergency caller is currently located.

When ISPs allow Internet traffic to traverse their network, the signaling and media protocols used for emergency calls function without problems. Today, there are no legal requirements to offer prioritization of emergency calls over IP-based networks. Although the standardization community has developed a range of *Quality of Service* (QoS) signaling protocols, they have not experienced widespread deployment.

VSP

SIP does not mandate that call setup requests traverse SIP proxies; that is, SIP messages can be sent directly to the user agent. Thus, even for emergency services it is possible to use SIP without the involvement of a VSP. However, in terms of deployment, it is highly likely that a VSP will be used. If a caller uses a VSP, this VSP often forces all calls, emergency or not, to traverse an outbound proxy or *Session Border Controller* (SBC) operated by the VSP. If some end devices are unable to perform a LoST lookup, VSP can provide the necessary functions as a backup solution.

If the VSP uses a signaling or media protocol that the PSAP does not support, it needs to translate the signaling or media flows.

VSPs can assist the PSAP by providing identity assurance for emergency calls; for example, using [30], thus helping to prosecute prank callers. However, the link between the subscriber information and the real-world person making the call is weak.

In many cases, VSPs have, at best, only the credit card data for their customers, and some of these customers may use gift cards or other anonymous means of payment.

PSAP

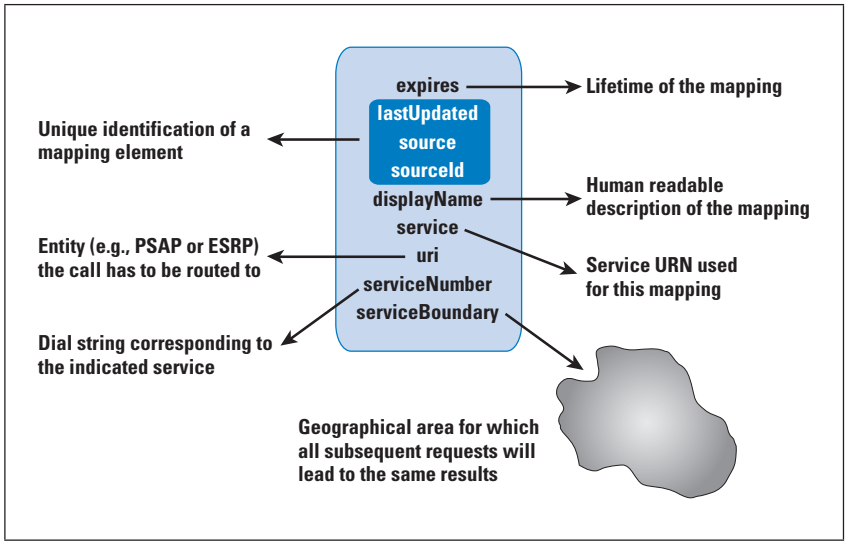
The emergency services Best Current Practice document [15] discusses only the standardization of the interfaces from the VSP and ISP toward PSAPs and some parts of the PSAP-to-PSAP call transfer mechanisms that are necessary for emergency calls to be processed by the PSAP. Many aspects related to the internal communication within a PSAP, between PSAPs as well as between a PSAP and first responders, are beyond the scope of the IETF specification.

When emergency calling has been fully converted to Internet protocols, PSAPs must accept calls from any VSP, as shown in interface (d) of Figure 2. Because calls may come from all sources, PSAPs must develop mechanisms to reduce the number of malicious calls, particularly calls containing intentionally false location information. Assuring the reliability of location information remains challenging, particularly as more and more devices are equipped with *Global Navigation Satellite Systems* (GNSS) receivers, including GPS and Galileo, allowing them to determine their own location[24]. However, it may be possible in some cases to check the veracity of the location information an endpoint provides by comparing it against infrastructure-provided location information; for example, a LIS-determined location.

Mapping Architecture

So far we have described LoST as a client-server protocol. Similar to the *Domain Name System* (DNS), a single LoST server does not store the mapping elements for all PSAPs worldwide, for both technical and administrative reasons. Thus, there is a need to let LoST servers interact with other LoST servers, each covering a specific geographical region. Working together, LoST servers form a distributed mapping database, with each server carrying mapping elements, as shown in Figure 3. LoST servers may be operated by different entities, including the ISP, the VSP, or another independent entity, such as a governmental agency. Typically, individual LoST servers offer the necessary mapping elements for their geographic regions to others. However, LoST servers may also cache mapping elements of other LoST servers either through data synchronization mechanisms (for example, FTP or exports from a *Geographical Information System* [GIS] or through a specialized protocol[25]) or by regular usage of LoST. This caching improves performance and increases the robustness of the system.

Figure 3: Mapping Element



A detailed description of the mapping architecture with examples is available in [29].

Steps Toward an IETF Emergency Services Architecture

The architecture described so far requires changes both in already-deployed VoIP end systems and in the existing PSAPs. The speed of transition and the path taken vary between different countries, depending on funding and business incentives. Therefore, it is generally difficult to argue whether upgrading endpoints or replacing the emergency service infrastructure will be easier. In any case, the transition approaches being investigated consider both directions. We can distinguish roughly four stages of transition (Note: The following descriptions omit many of the details because of space constraints):

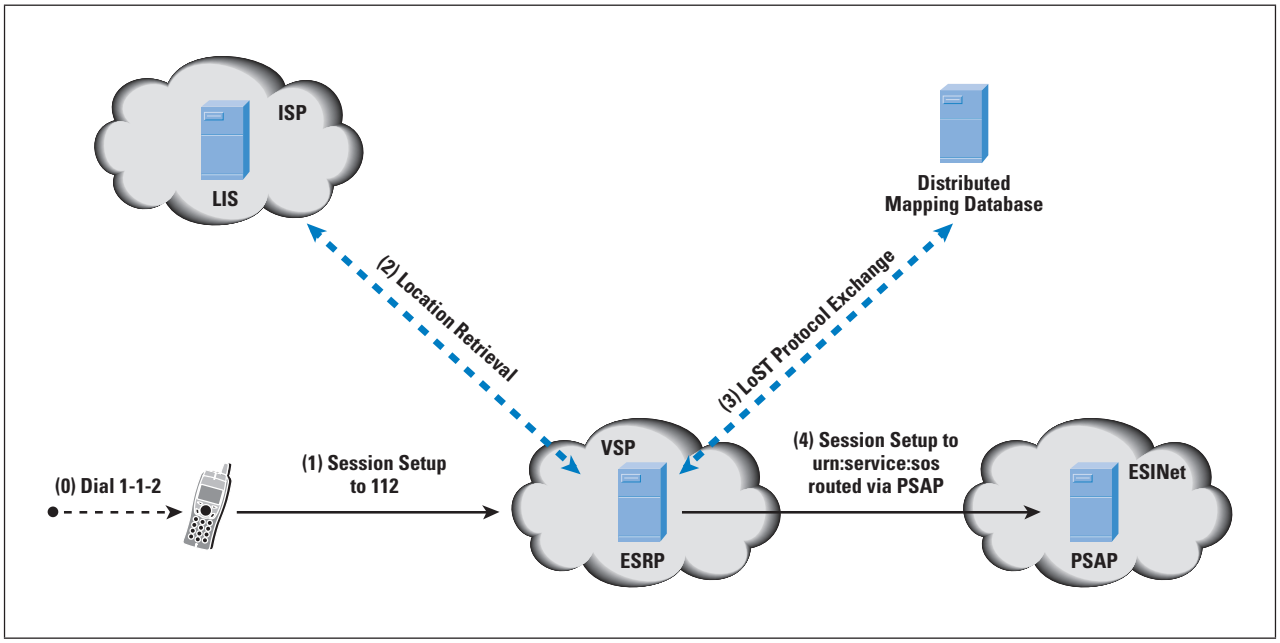
1. Initially, VoIP end systems cannot place emergency calls at all; for example, many software clients, such as *GoogleTalk*, cannot place emergency calls.
2. In a second stage, VoIP callers manually configure their location, and emergency calls are routed to the appropriate PSAP as circuit-switched calls through PSTN gateways using technologies similar to mobile calls. This level of service is now offered in some countries for PSTN-replacement VoIP services; that is, VoIP services that are offered as replacement for the home phone. In the United States, this service is known as the "NENA I2" service.
3. In a third stage, PSAPs maintain two separate infrastructures, one for calls arriving through an IP network and the traditional infrastructure.
4. In the final stage, all calls, including those from traditional cell phones and analog landline phones, reach the PSAP through IP networks, with the traditional calls converted to the ECRIT requirements by the carriers or the emergency service infrastructure.

If devices are used in environments without location services, the VSP's SIP proxy may need to insert location information based on estimates or subscriber data. These cases are described briefly in the following sections.

Traditional Endpoints

Figure 4 shows an emergency services architecture with traditional endpoints. When the emergency caller dials the Europeanwide emergency number 112 (step 0), the device treats it as any other call without recognizing it as an emergency call; that is, the dial string provided by the endpoint that may conform to RFC 4967[26] or RFC 3966[27] is signaled to the VSP (step 1). Recognition of the dial string is then left to the VSP for processing or sorting; the same is true for location retrieval (step 2) and routing to the nearest (or appropriate) PSAP (step 3). Dial-string recognition, location determination, and call routing are simpler to carry out using a fixed device and the voice and application service provided through the ISP than they are when the VSP and the ISP are two separate entities.

Figure 4: Emergency Services Architecture with Traditional Endpoints



There are two main challenges to overcome when dealing with traditional devices: First, the VSP must discover the LIS that knows the location of the IP-based end host. The VSP is likely to know only the IP address of that device, visible in the call signaling that arrives at the VSP. When a LIS is discovered and contacted and some amount of location information is available, then the second challenge arises, namely, how to route the emergency call to the appropriate PSAP. To accomplish the latter task it is necessary to have some information about the PSAP boundaries available.

Reference [15] does not describe a complete and detailed solution but uses building blocks specified in ECRIT. Still, this deployment scenario shows many constraints:

- Only the emergency numbers configured at the VSP are understood. This situation may lead to cases where a dialed emergency number is not recognized.
- Using the IP address to find the ISP is challenging and may, in case of mobility protocols and VPNs, lead to wrong results.
- Security concerns might arise when a potentially large number of VSPs or ASPs are able to retrieve location information from an ISP. It is likely that only authorized VSP and ASPs will be granted access. Hence, it is unlikely that such a solution would work smoothly across national boundaries.
- When the user agent does not recognize the emergency call, functions such as call waiting, call transfer, three-way call, flash hold, and outbound call blocking cannot be disabled.
- The user-agent software may block callbacks from the PSAP.
- Privacy settings may not get considered and identity may get disclosed to unauthorized parties. These identity privacy features exist in some jurisdictions even in emergency situations.
- Certain VoIP call features may not be supported, such as REFER (for conference call and transfer to secondary PSAP) and *Globally Routable UA URI* (GRUU).
- User agents will not convey location information to the VSP (even if available).

Partially Upgraded End Hosts

A giant step forward in simplifying the handling of IP-based emergency calls is to provide the end host with some information about the ISP so that LIS discovery is possible. The end host may, for example, learn the ISP's domain name by using LIS discovery[28], or might even obtain a *Location by Reference* (LbyR) through the DHCP-URI option[13] or through HELD[11]. The VSP can then either resolve the LbyR in order to route the call or use the domain to discover a LIS using DNS.

Additional software upgrades at the end device may allow for recognition of emergency calls based on some preconfigured emergency numbers (for example, 112 and 911) and allow for the implementation of other emergency service-related features, such as disabling silence suppression during emergency calls.

Outlook

In most countries, national and sometimes regional telecommunications regulators, such as the *Federal Communications Commission* (FCC) and individual states, or the European Union, strongly influence how emergency services are provided, who pays for them, and the obligations that the various parties have. Regulation is, however, still at an early stage: in most countries current requirements demand only manual update of location information by the VoIP user. The ability to obtain location information automatically is, however, crucial for reliable emergency service operation, and it is required for nomadic and mobile devices. (Nomadic devices remain in one place during a communication session, but are moved frequently from place to place. Laptops with Wi-Fi interfaces are currently the most common nomadic devices.)

Regulators have traditionally focused on the national or, at most, the European level, and the international nature of the Internet poses new challenges. For example, mobile devices are now routinely used beyond their country of purchase and, unlike traditional cellular phones, need to support emergency calling functions. It appears likely that different countries will deploy IP-based emergency services over different time horizons, so travelers may be surprised to find that they cannot call for emergency assistance outside their home country.

The separation between Internet access and application providers on the Internet is one of the most important differences to existing circuit-switched telephony networks. A side effect of this separation is the increased speed of innovation at the application layer, and the number of new communication mechanisms is steadily increasing. Many emergency service organizations have recognized this trend and advocated for the use of new communication mechanisms, including video, real-time text, and instant messaging, to offer improved emergency calling support for citizens. Again, this situation requires regulators to rethink the distribution of responsibilities, funding, and liability.

Many communication systems used today lack accountability; that is, it is difficult or impossible to trace malicious activities back to the persons who caused them. This problem is not new, because pay phones and prepaid cell phones have long offered mischief makers the opportunity to place hoax calls, but the weak user registration procedures, the lack of deployed end-to-end identity mechanisms, and the ease of providing fake location information increases the attack surface at PSAPs. Attackers also have become more sophisticated over time, and Botnets that generate a large volume of automated emergency calls to exhaust PSAP resources, including call takers and first responders, are not science fiction.

References

[1] Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling Using Internet Multimedia," Internet Draft, work in progress, **draft-ietf-ecrit-framework-11**, July 2010.

[2] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, June 2002.

[3] Winterbottom, J., Thomson, M., Tschofenig, H., and H. Schulzrinne, "ECRIT Direct Emergency Calling," Internet Draft, work in progress, **draft-winterbottom-ecrit-direct-02.txt**, March 2010.

[4] Schulzrinne, H., McCann, S., Bajko, G., Tschofenig, H., and D. Kroeselberg, "Extensions to the Emergency Services Architecture for Dealing with Unauthenticated and Unauthorized Devices," Internet Draft, work in progress, **draft-ietf-ecrit-unauthenticated-access-00.txt**, September 2010.

[5] Schulzrinne, H., "A Uniform Resource Name (URN) for Emergency and Other Well-Known Services," RFC 5031, January 2008.

[6] Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol," RFC 5222, August 2008.

[7] Peterson, J., "A Presence-based GEOPRIV Location Object Format," RFC 4119, December 2005.

[8] Thomson, M. and J. Winterbottom, "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)," RFC 5139, February 2008.

[9] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations," RFC 5491, March 2009.

[10] R. Marshall, "Requirements for a Location-by-Reference Mechanism," RFC 5808, May 2010.

[11] M. Barnes, "HTTP Enabled Location Delivery (HELD)," RFC 5985, September 2010.

[12] Polk, J., Schnizlein, J., and M. Linsner, "Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information," RFC 3825, July 2004.

[13] Polk, J., "Dynamic Host Configuration Protocol (DHCP) IPv4 and IPv6 Option for a Location Uniform Resource Identifier (URI)," Internet Draft, work in progress, **draft-ietf-geopriv-dhcp-lbyr-uri-option-08**, July 2010.

[14] Polk, J., Rosen, B., and J. Peterson, "Location Conveyance for the Session Initiation Protocol," Internet Draft, work in progress, **draft-ietf-sipcore-location-conveyance-03**, July 2010.

[15] Rosen, B. and J. Polk, "Best Current Practice for Communications Services in Support of Emergency Calling," Internet Draft, work in progress, **draft-ietf-ecrit-phonebcp-15**, July 2010.

[16] Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information," RFC 4776, November 2006.

[17] Polk, J., Schnizlein, J., Linsner, M., and B. Aboba, "Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information," Internet Draft, work in progress, **draft-ietf-geopriv-rfc3825bis-11**, July 2010.

[18] Winterbottom, J., Thomson, M., Tschofenig, H., and R. Barnes, "Use of Device Identity in HTTP-Enabled Location Delivery (HELD)," Internet Draft, work in progress, **draft-ietf-geopriv-held-identity-extensions-04**, June 2010.

[19] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification," RFC 3265, June 2002.

[20] Mahy, R., Rosen, B., and H. Tschofenig, "Filtering Location Notifications in the Session Initiation Protocol," Internet Draft, work in progress, **draft-ietf-geopriv-loc-filters-11**, March 2010.

[21] Winterbottom, J., Tschofenig, H., Schulzrinne, H., Thomson, M., and M. Dawson, "A Location Dereferencing Protocol Using HELD," Internet Draft, work in progress, **draft-ietf-geopriv-deref-protocol-01**, September 2010.

[22] Schulzrinne, H., Liess, L., Tschofenig, H., Stark, B., and A. Kuett, "Location Hiding: Problem Statement and Requirements," Internet Draft, work in progress, **draft-ietf-ecrit-location-hiding-req-04**, Feb 2010.

[23] Barnes, R., and M. Lepinski, "Using Imprecise Location for Emergency Context Resolution," Internet Draft, work in progress, **draft-ietf-ecrit-rough-loc-03**, August 2010.

[24] Tschofenig, H., Schulzrinne, H., and B. Aboba, "Trustworthy Location Information," Internet Draft, work in progress, **draft-tschofenig-ecrit-trustworthy-location-00**, September 2010.

[25] Schulzrinne, H., and H. Tschofenig, "Synchronizing Location-to-Service Translation (LoST) Servers," Internet Draft, work in progress, **draft-ietf-ecrit-lost-sync-10**, March 2010.

[26] B. Rosen, "Dial String Parameter for the Session Initiation Protocol Uniform Resource Identifier," RFC 4967, July 2007.

[27] H. Schulzrinne "The tel URI for Telephone Numbers," RFC 3966, December 2004.

[28] Thomson, M. and J. Winterbottom, "Discovering the Local Location Information Server (LIS)," RFC 5986, September 2010.

[29] H. Schulzrinne, "Location-to-URL Mapping Architecture and Framework," RFC 5582, September 2009.

[30] C. Jennings, J. Peterson, and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks," RFC 3325, November 2002.

HENNING SCHULZRINNE, Levi Professor of Computer Science at Columbia University, received his Ph.D. from the University of Massachusetts in Amherst, Massachusetts. He was an MTS at AT&T Bell Laboratories and an associate department head at GMD-Fokus (Berlin) before joining the Computer Science and Electrical Engineering departments at Columbia University. He served as chair of Computer Science from 2004 to 2009. Protocols that he co-developed, such as RTP, RTSP, and SIP, are now Internet standards, used by almost all Internet telephony and multimedia applications. His research interests include Internet multimedia systems, ubiquitous computing, and mobile systems. He is a Fellow of the IEEE. E-mail: **hgs@cs.columbia.edu**

HANNES TSCHOFENIG received a Diploma degree from the University of Klagenfurt, Austria. He joined Siemens Corporate Technology, Munich, in 2001 and joined Nokia Siemens Networks in April 2007 to move to Finland in December 2007, where he focuses on standards development. Most of his time is dedicated to the participation in the Internet Engineering Task Force (IETF) where he, among other responsibilities, co-chaired the ECRIT working group from 2005 to early 2010. Additionally, he co-chairs the Next Generation 112 Technical Committee of the European Emergency Number Association (EENA) and contributes to the technical specifications developed within the National Emergency Number Association (NENA), and he co-organized the SDO emergency services workshop series. In March 2010 he joined the Internet Architecture Board (IAB).
E-mail: **hannes.tschofenig@nsn.com**

Integration of Core BGP/MPLS VPN Networks

by Paul Veitch, Paul Hitchen, and Martin Mitchell, BT Innovate & Design

This article explores the architectural and operational challenges involved in integrating an existing standalone core *Border Gateway Protocol* (BGP)/*Multiprotocol Label Switching* (MPLS) VPN network onto a target *Next-Generation Network* (NGN). The rationale for consolidating and transforming multiple networks is explained, mainly in terms of potential cost savings and operational simplification achieved by the network operator. The article specifically focuses on the MPLS *Carrier-supporting-Carrier* (CsC) architectural framework, which allows the serving nodes of one MPLS VPN network to be interconnected through the serving nodes of another MPLS VPN network. The required architectural building blocks to implement CsC, the manner in which routing protocols must interact, as well as end-to-end packet flow and label encapsulation are all explained. The main design and operational challenges, including maintaining performance levels for customers, network resiliency, fault-handling, and capacity management, are also addressed in this article.

Network operators are under increasing pressure to deliver exceptional levels of customer experience and service while decreasing the capital and operational cost base of their networks. Many operators have traditionally built multiple network platforms, each of which has been uniquely designed to meet the requirements of specific services targeted at specific customer markets, such as voice, broadband IP, *Virtual Private Networks* (VPNs), etc.

In a bid to remain competitive and achieve cost reductions and operational simplifications, many operators have built all IP-based NGNs. The principal transformational benefits of an NGN with a single protocol such as IP at its heart include versatility in catering for multiple traffic requirements (for example, by employing IP *Quality-of-Service* [QoS] techniques), the ability to introduce novel and reusable services and features in a flexible manner, and the potential to maximise vendor interworking due to standards-based technology.

When a network operator builds an NGN, the challenge remains as to how to migrate *existing* networks and customers onto the new platform. The full commercial benefits of an NGN can be properly realised only after legacy networks are either consolidated or phased out completely. Many important factors must be considered, including the cost benefits, the potential effect on end customers, and the operational approach to carrying out migrations. These concerns must be weighed against the commercial and business risks associated with the alternative approach of sustaining and running multiple standalone platforms indefinitely.

This article focuses on a specific scenario: how to integrate an existing BGP/MPLS VPN network that provides VPN services to a corporate customer base with a "target" NGN. Following a brief overview of MPLS VPN services and networks, the rationale for consolidating multiple MPLS VPN networks is explained, mainly in terms of potential cost savings and operational simplification achieved by the network operator. The article then details the MPLS CsC architectural framework that allows the serving nodes or *Points of Presence* (POPs) of one MPLS VPN network to be interconnected to the serving nodes of another MPLS VPN network. The way in which routing protocols must interact and the subsequent effect on end-to-end packet forwarding across a CsC-enabled core network are explained. The principal design and operational challenges introduced by integrating core MPLS networks are then outlined, including maintaining performance levels, network resiliency, fault management, and capacity management.

The Business Case for MPLS VPN Network Consolidation

VPNs are an attractive solution to serve the enterprise networking requirements of a wide range of businesses from *Small-to-Medium Enterprises* (SMEs) to multinational "blue-chip" corporate organisations. Essentially, VPNs provide a transparent network infrastructure that allows multiple customer sites to communicate over a shared backbone network, as though they are using their own private network, regardless of geographical location. Typical applications that run across an organisation's VPN include corporate Intranet, mail services, and *Voice-over-IP* (VoIP) telephony.

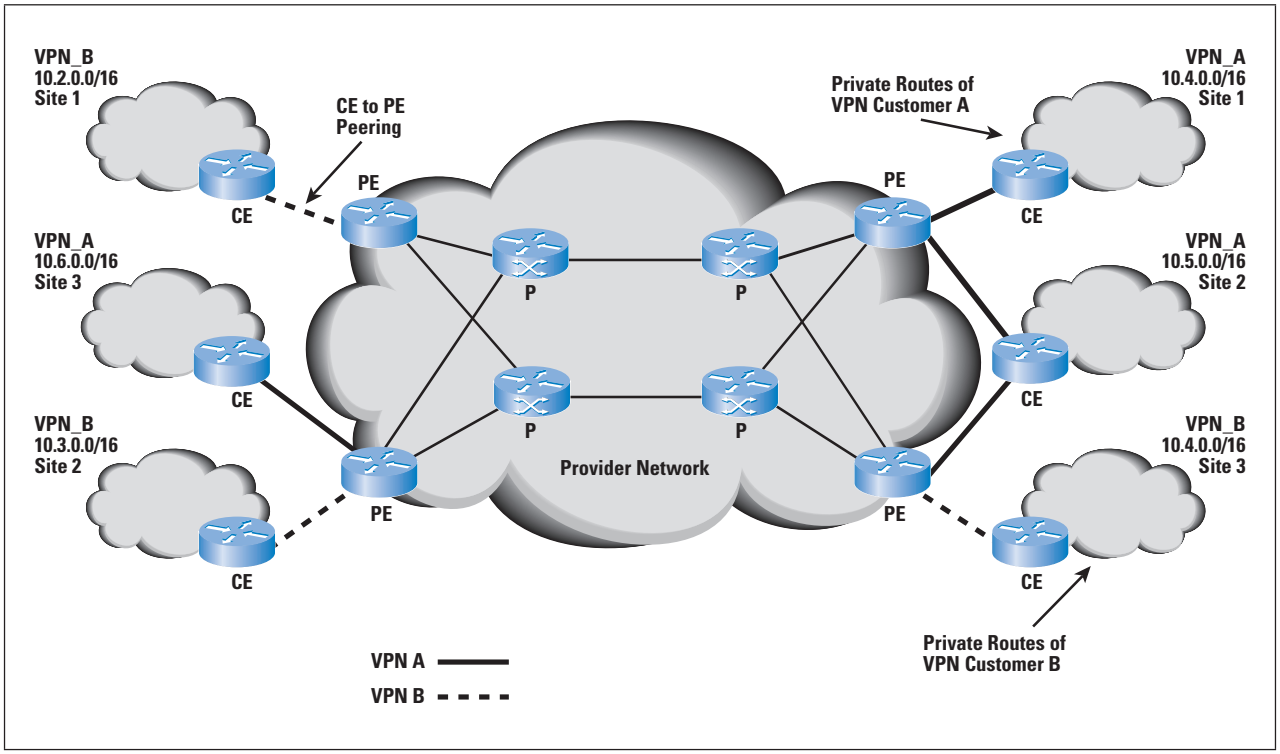
Although distinct categories of VPN networking technology exist[1], this article focuses exclusively on "Layer 3" BGP/MPLS VPNs, as defined in RFC 4364[2] and other related Internet Drafts. Such networks have been deployed for more than 10 years and have seen significant growth during that period.

The critical core network elements of a provider-provisioned BGP/MPLS VPN network are *Provider Edge* (PE) and *Provider Core* (P) routers, as shown in Figure 1.

PE routers terminate customer access circuits, whereas P routers perform packet forwarding and typically do not have directly connected customer access circuits. PE routers perform label encapsulation and de-encapsulation, P routers run label switching, and both operate control-plane protocols that build MPLS *Label Switched Paths* (LSPs) from each PE to each other PE. Many protocols can be used to establish these LSPs; a commonly deployed approach uses the *Label Distribution Protocol* (LDP) in conjunction with an *Interior Gateway Protocol* (IGP), such as *Open Shortest Path First* (OSPF).

When a PE forwards a VPN-addressed packet across the core, it adds an inner MPLS label to identify the VPN of which the packet is a member and then an outer MPLS label to identify the egress PE router. Any intermediate P routers switch the packet to the egress PE using the outer label only. The egress PE uses the inner label to determine which VPN or port to forward the packet to.

Figure 1: Overview of BGP/MPLS VPN Network



The *Customer Edge* (CE) router is not considered part of the provider's core network. It acts as a peer of the PE router, but not a peer of other CE routers. Each PE router supports multiple routing and forwarding tables, called *Virtual Route Forwarding* (VRF) tables. VRF routes are logically separate, and they may contain IP prefixes received from the CE router that overlap with addresses in other VRFs. (For example, in Figure 1, VPN_A, site 1 has the same private routes as VPN_B, site 3.) VPNs are formed by defining individual customer accesses to be members of a specific VRF table, with several sites formed on one PE by defining all sites to use the same VRF table or allocating each site a VRF table and controlling connectivity through selective import and export of the IP routes of each VRF table.

The PE routers use an extended variant of BGP for signaling between themselves and propagating information about the actual routes of each VPN, as well as the inner MPLS label. The extended BGP, referred to as *Multiprotocol BGP*, carries each VPN route together with two new fields, the *Route Distinguisher* (RD) and the *Route Target* (RT), a form of extended BGP Community.

The RD is added to each VPN route to ensure that routes from different customers are unique; BGP treats VPN routes as equal only if both the RD and the IP prefix mask are equal. BGP uses RTs to indicate a group of routes, thus defining VPN membership information for exchange between PEs.

Maintenance Costs of BGP/MPLS VPN Networks

As detailed in the previous section, the main core components of a VPN network based on BGP/MPLS technology are the PE and P routers. Although not shown in detail in Figure 1, another critical element of a core VPN network is the *Wide-Area Network* (WAN) topology that interconnects the P (core) routers residing in specific service nodes, also called POPs. The WAN topology is essentially the way in which transmission links—typically *Synchronous Optical Network* (SONET)/*Packet over SONET/SDH* (PoS), Gigabit Ethernet, or 10 Gigabit Ethernet—are used to interconnect the POPs together.

It follows that maintenance costs associated with a self-contained MPLS VPN network will be incurred for PE and P routers, as well as the interconnecting WAN transmission links. These maintenance costs will split into capital and operational elements.

Capital expenditures are required on an ongoing basis for all IP router infrastructure (PE and P routers), for example, to upgrade hardware to meet increasing capacity demands, replace faulty line cards and processors, or replace end-of-life hardware with newer equipment. Capital expenditures are also needed on WAN links, for example, to replace faulty line cards and optics, as well as to deploy increased capacity transmission links to cater for traffic growth across the core network. Further capital costs accrue from accommodation-related aspects such as power, racking, and air conditioning.

Additional maintenance costs reside in the operational space. For example, if an MPLS VPN network has 40 POP locations, each with a pair of P (core) routers, the 80 core routers will consume a certain amount of operational team resources for critical maintenance, scheduled maintenance activities, and ongoing monitoring and reporting of router status (processors and line cards).

Benefits of Core Integration

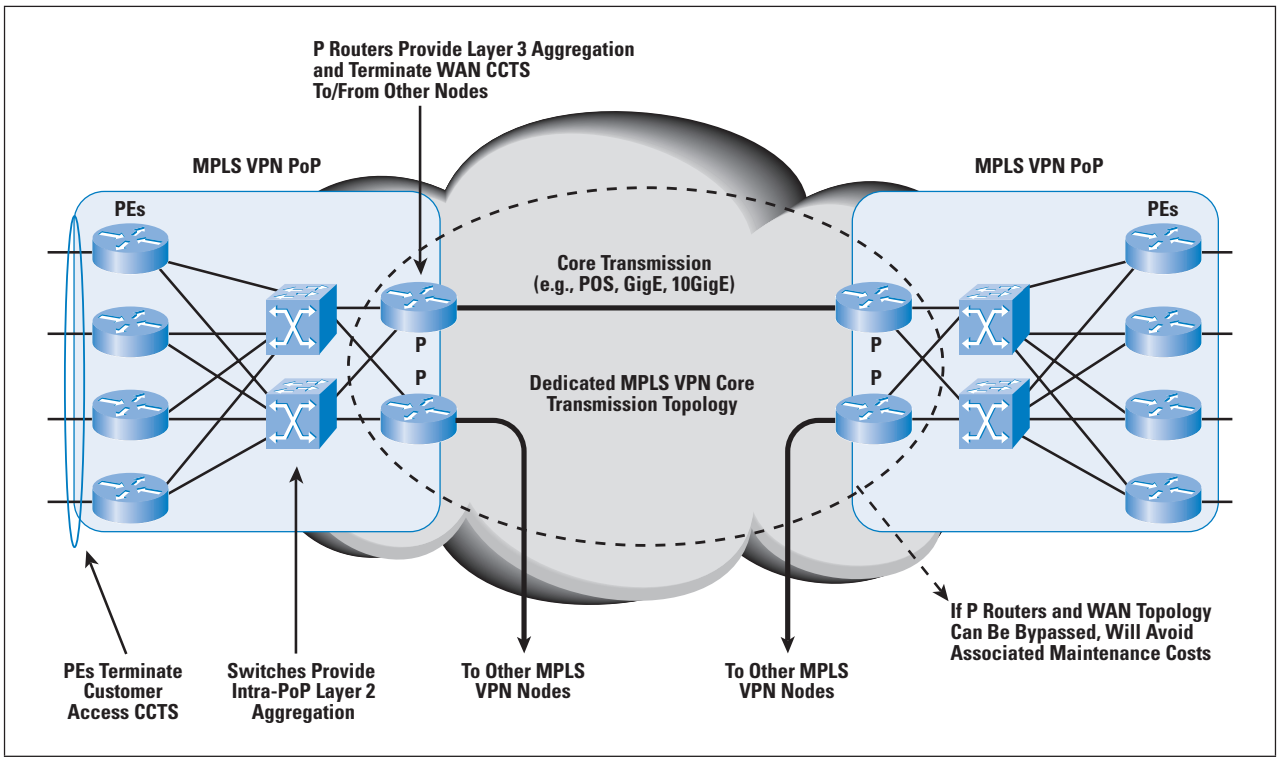
If a network operator has deployed an IP-based NGN alongside an existing MPLS VPN network, the question should be asked: can the existing MPLS VPN network be integrated onto the NGN so as to avoid some or all of the previously stated maintenance costs? One approach would be to target the P (core) routers and WAN transmission links for eventual removal (Figure 2) and replacement by suitable connectivity of the MPLS VPN nodes to the NGN network. The VPN PE routers that often terminate large volumes of customer access circuits and host the rich service-related functions for corporate VPN services can essentially be left *in situ,* minimising the effect on end customers and confining the integration of networks to the inner part of the core infrastructure. The way in which this goal can actually be achieved in practice is detailed in the next section.

The main benefits that can be accrued for the network operator are as follows:

- Substantial cost avoidance for maintaining and upgrading P (core) routers and dedicated WAN links for the existing MPLS VPN network can be achieved (Figure 2). As much as a 35-percent reduction of fixed inner core capital costs is possible.
- If the technical solution for core integration can be made as reusable as possible, then in addition to allowing integration of "same provider" core networks, the network operator could provide the capability on a wholesale basis for other service providers. This capability could be a potentially significant source of new revenue.
- From an operational perspective, integration of core networks should lend itself to a singular and much more streamlined approach to capacity planning, fault management, and network monitoring.

The combination of all these benefits can produce a compelling business case for network operators to consolidate core MPLS-based network platforms.

Figure 2: MPLS VPN Network Showing Inner Core Components Targeted for Replacement



Carrier-supporting-Carrier Framework

Carrier-supporting-Carrier (CsC) is a term used to describe a situation where one network, designated the *customer carrier,* is permitted to use a segment of another network, designated the *backbone carrier*[3]. Although the term "Carrier of Carriers" is also used to describe the same architectural framework, this article uses Carrier-supporting-Carrier for consistency. In principle, the two "carrier" networks could belong to the same organisation, or could belong to two different organisations. Whatever the case, there is no reason why the backbone carrier cannot support multiple customer carrier networks. Furthermore, the customer carrier network itself can be either a BGP/MPLS VPN network providing Layer 3 VPN services or an *Internet Service Provider* (ISP) network[3].

A network operator with an existing BGP/MPLS VPN network infrastructure that has also built an IP-based NGN based on BGP/MPLS technology as per RFC 4364[2] could choose to exploit the CsC architectural framework to merge the two core networks. In such a scenario, the existing BGP/MPLS VPN network that serves the needs of VPN business customers would be viewed as the "customer carrier," whereas the NGN network would be positioned as the "backbone carrier."

Physical Connectivity and CsC VRF Creation

In order to integrate an existing BGP/MPLS VPN network such as that shown in Figure 2, with an NGN core belonging to the same or different organisation, the NGN network must be enabled to act as a backbone carrier. Assuming the NGN network is configured to support BGP/MPLS VPNs as per RFC 4364[2], it comprises PE and P router core infrastructure. The PE routers of the NGN acting as the backbone carrier are denoted "CsC-PEs." The PE routers of the existing BGP/MPLS VPN network, that is, the customer carrier network that is being itself integrated with the NGN core, are denoted "CsC-CEs."

As shown in Figure 3, the NGN backbone carrier network provides MPLS VPN service to the customer carrier network using its own VRF table enabled on the CsC-PE. One important distinction between normal MPLS VPN service and CsC is the fact that traffic passed between the CsC-CE and CsC-PE is labeled rather than native IP[3, 4].

The CsC architecture is designed such that the backbone carrier network—the network provider's NGN network—needs to know only about internal routes within the customer carrier network. This setup allows formation of full "any-to-any" logical connectivity between the customer carrier routers, which in this scenario are the PE routers of the existing BGP/MPLS VPN network providing VPN services to end customers.

Furthermore, the backbone carrier routers themselves do not need to retain route prefix information for the end-customer VPNs connected to the customer carrier network because the end-customer traffic is transported over a second level of VRF tables that bear relevance only to the customer carrier itself, that is, the endpoint CsC-CEs. This *nesting* of MPLS VPN networks emphasises the inherent scalability of the CsC architecture. The CsC backbone carrier is effectively behaving like "proxy" P routers for the customer carrier network.

Figure 3: MPLS VPN "Customer Carrier" Network Connected Across NGN "Backbone Carrier"

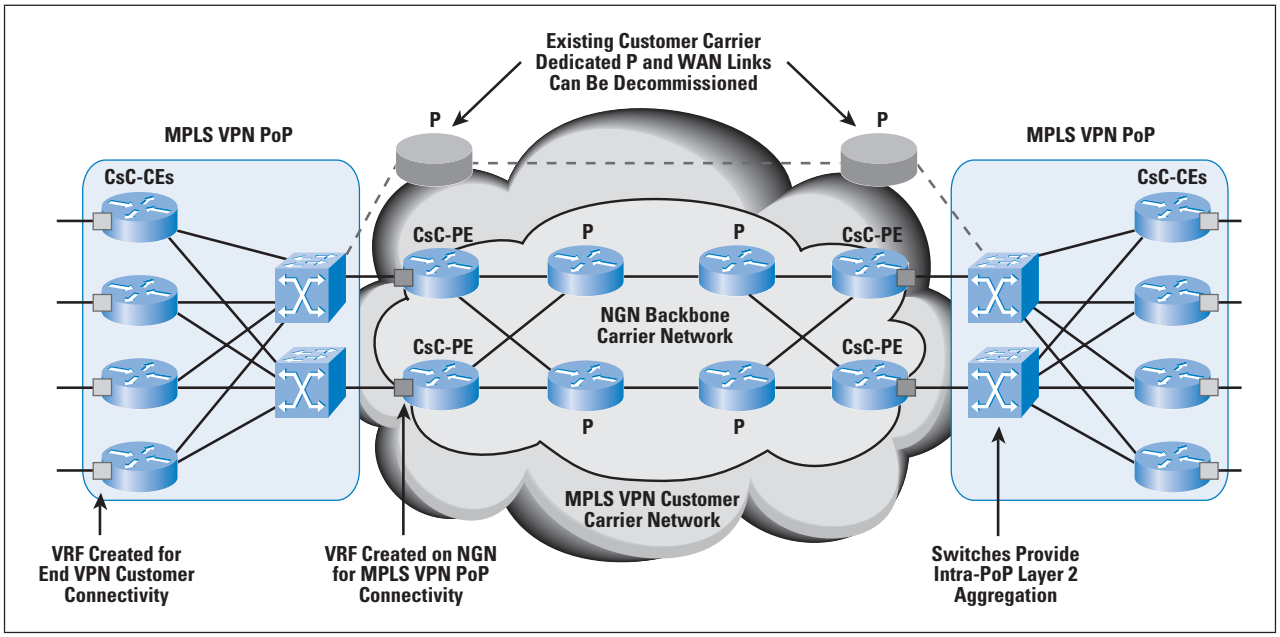


Figure 3 also shows the physical connectivity between the customer carrier network and backbone carrier NGN. Because many large-scale BGP/MPLS network deployments comprise large numbers of PE devices in the same service node or POP, there is often a Layer 2 Ethernet switch acting as an "intra-POP" aggregator. It is convenient to allow physical connectivity between the BGP/MPLS VPN service node and the CsC-PE in the NGN network using this aggregation switch. One or more *Virtual LANs* (VLANs) can be configured across this physical trunk to provide logical Layer 2 connectivity into the CsC-PE on the NGN, and be associated with the CsC VRF on that device. The Layer 2 switch also provides direct intra-POP connectivity between CsC-CEs present on the same VLANs.

Control-Plane Routing Protocols

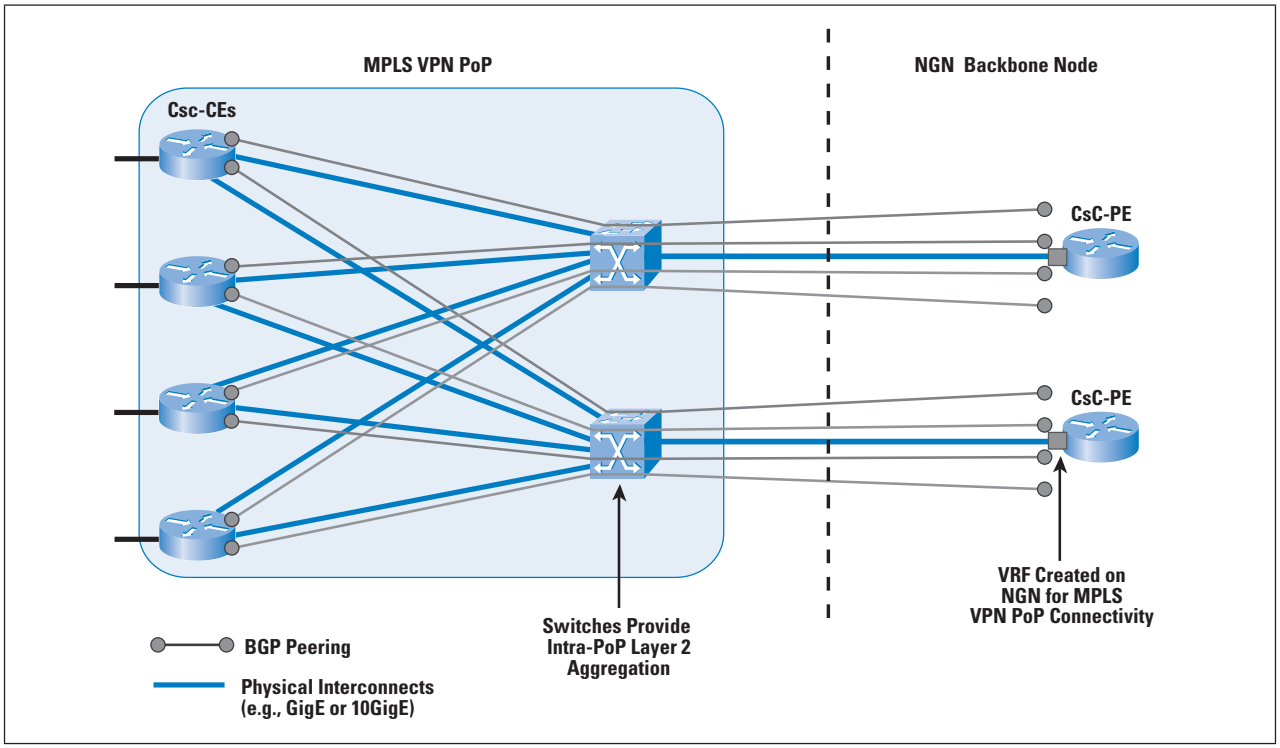
The previous section described the physical connectivity between BGP/MPLS VPN service nodes and the target NGN, with creation of a specific VRF route on the CsC-PEs. This section addresses the way in which the internal routes of the CsC-CEs (that is, the PE routers belonging to the customer carrier BGP/MPLS VPN network) are advertised into this VRF table.

Optional routing protocols include the use of an IGP such as OSPF, or *Exterior Gateway Protocols* (EGPs) such as BGP. With an IGP like OSPF[5], the routing protocol itself is used for route exchange between the CsC-CEs and CsC-PEs, and must be used in conjunction with an LDP[6] for MPLS label exchange between the CsC-CEs and CsC-PEs.

Separating the IP prefix and label allocation protocols between an IGP and LDP can introduce complexities with potential divergence between the two control planes. Such divergence in the extreme case can lead to partial or complete loss in forwarding. Use of an EGP like BGP, however, can be used to implement CsC as a single IP prefix and Label Allocation control-plane protocol between CsC-CE and CsC-PE. Piggybacking MPLS label-mapping information in the BGP update messages helps ensure that an IP prefix and its associated MPLS label are always synchronised in their delivery. The way in which this synchronisation is achieved is documented in RFC 3107[7]. BGP has the benefit of being a mature protocol for use either within the same network organisation or between networks belonging to different operators. Furthermore, BGP employs mechanisms for loop avoidance and control over the number and type of routes advertised and accepted.

Figure 4 shows an example scenario whereby two BGP peerings are established (for resiliency) between each of the four CsC-CEs (which are actually PE routers of the BGP/MPLS VPN customer carrier network) and a pair of target CsC-PE routers (which are the PE routers of the NGN backbone carrier network).

Figure 4: BGP Plus Labels as the Routing Protocol Between CsC-CEs and CsC-PEs

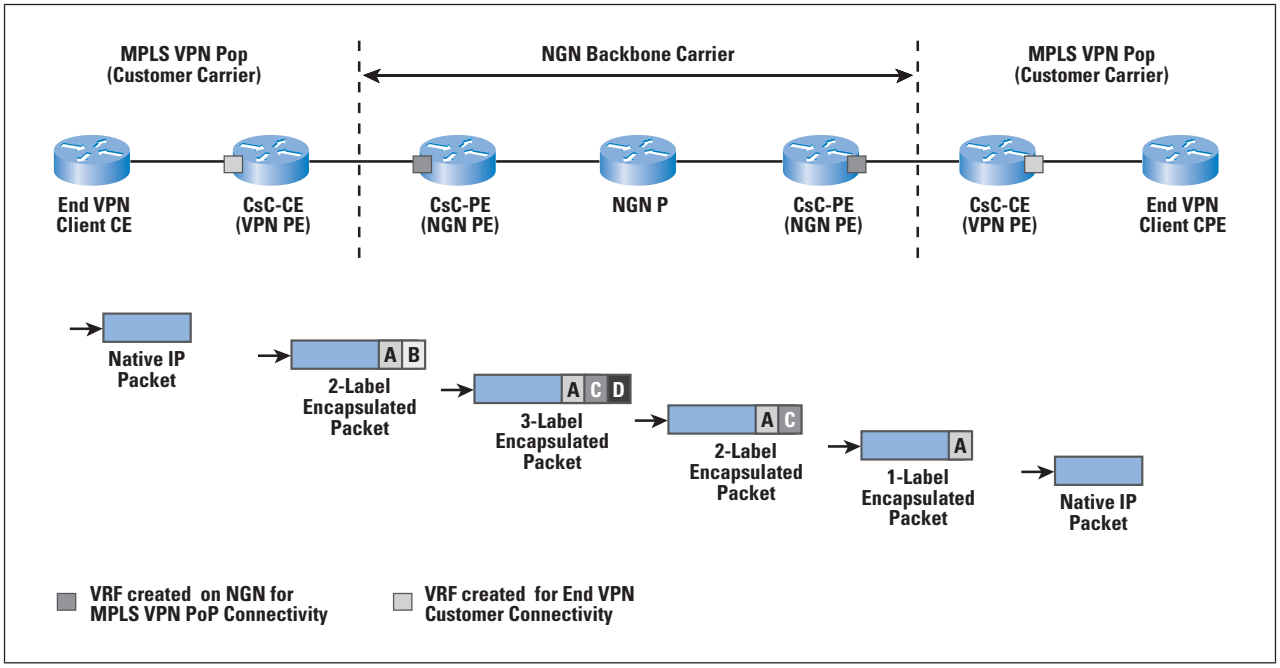


Label Switching of Customer Packets

As shown in Figure 5, viewing packet flow from left to right, a unicast packet originates as a native IP packet when presented from the end client CE router to the MPLS VPN PE router, which is behaving as a CsC-CE in this context. Upon traversal between CsC-CEs in different MPLS VPN POP locations connected by an NGN backbone carrier using CsC, the packet ultimately undergoes three levels of label encapsulation:

- The innermost label corresponds to the *End Customer VRF.* This label is transparent to the NGN backbone carrier (that is, it is not operated upon in lookup and forwarding tables with the NGN). It is label "A" in Figure 5.
- The middle label is the "outer label" as far as the CsC-CE is concerned, swapped at the CsC-PE, and becomes the "inner label" as far as the NGN backbone carrier is concerned. In Figure 5, this label is assigned as label "B" by the CsC-CE as instructed by the CsC-PE through the BGP plus labels (RFC 3107-compliant) peering. At the CsC-PE itself, the label is swapped (to become label "C" in Figure 5) and is used to associate the packet with the CsC VRF. The packet is then identifiable at the destination CsC-PE at the far end of the backbone carrier network; it allows forwarding to the correct interface.
- The outermost label (shown as label "D" in Figure 5) is assigned by the backbone carrier LDP process at the CsC-PE router, and is present only to allow transport across the backbone carrier CsC core. Thus when a packet leaves the CsC-PE for transport across the backbone carrier core it has three levels of labels on each packet.

Figure 5: Label Encapsulation and End-to-End Packet Flow Across a CsC Core Network



As shown in Figure 5, the last P router in the backbone carrier path has "popped" the outermost label (label "D") using penultimate-hop label forwarding. The destination CsC-PE uses and removes the middle label (label "C") to indicate the correct outgoing interface, leaving only the innermost label on presentation to the CsC-CE (label "A"). This CsC-CE, which is the PE router in relation to the end VPN services, uses the last remaining label to determine the VRF route and interface on which to send the native IP packet so that it reaches the required client CE router.

Design and Operational Challenges

The previous section outlined the architectural framework of using CsC to integrate one BGP/MPLS core network with another. This section addresses the important design and operational challenges that such a network transformation brings about.

Maintaining Performance Levels

Many existing operators of "carrier-class" BGP/MPLS networks exploit IP QoS mechanisms to allow different IP-based traffic types to be treated in different ways in terms of how the packets are conveyed across the core network. This treatment relates chiefly to prioritisation of delay, jitter, and/or loss-sensitive traffic, against traffic types that are less sensitive to loss or delay. Customers of VPN services supported on such networks generally demand support of a range of traffic types, including corporate intranet, transactional applications, mail services, data backup, video, and VoIP telephony.

To deal with the range of traffic types, BGP/MPLS VPN service providers have developed the means of supporting IP QoS defining different transport classes with associated service levels. One such example may map, for instance, six service classes based on IETF "Per-Hop Behaviours" as defined by the *Differentiated Services* (DiffServ) working group[8, 9] and the recommended *DiffServ Code Point* (DSCP) values for them. The classes in this example could be broadly described as follows:

- *Expedited Forwarding* (EF), designed and optimised for the delivery of jitter and delay-sensitive applications such as VoIP
- *Assured Forwarding* (AF), intended to support priority data applications; the AF class is split into four equivalent sub-classes (AF1–AF4) used to segregate data or video traffic applications, with priority being maintained over the Default class
- *Default* (DE), to support "best-effort" (that is, unprioritized) data traffic

The DSCP markings dictate the way in which such traffic is placed into queues and conveyed across the core network. At the edge of the MPLS core, the PE maps the incoming DSCP value into the MPLS *Class-of-Service* (CoS) bits (formerly known as EXP bits).

The details of the mapping relate to the specific implementation and policy of the service provider. Under heavy traffic load and congestion situations, such policies dictate how packets are treated in terms of scheduling, queuing, and discard eligibility.

Both the existing BGP/MPLS "customer carrier" and the target NGN "backbone carrier" networks already have their own implementation of QoS classes to allow management and prioritisation of multiple traffic types carried across their respective core infrastructures. A significant design challenge that arises with integrating the networks is that a suitable mapping of the QoS schema present on the PE routers of the customer carrier network (the CsC-CEs in earlier diagrams) to the QoS schema supported on the PE routers of the NGN (the CsC-PEs in earlier diagrams) is necessary.

It is imperative that such a mapping not compromise the existing customer experience for VPN services in terms of packet loss, packet delay, and packet jitter (that is, delay variance). Careful design, mapping of the required service levels, and ultimately end-to-end testing of the QoS mappings is therefore necessary to assure the maintenance of performance levels after the networks are integrated with CsC.

Network Resiliency

As described earlier in the article and shown in Figure 2, an existing standalone BGP/MPLS network platform has interconnected POP locations using underlying core transmission infrastructures such as SONET/SDH/*Dense Wavelength-Division Multiplexing* (DWDM). The actual number of WAN circuits deployed, the use of transmission-layer protection mechanisms, and the overall topological connectivity between POPs determine overall levels of network resiliency. In turn, this aspect of the network architecture significantly affects the overall level of service availability to end customers of VPN services.

When the standalone BGP/MPLS network has its existing core topology replaced with that of the NGN backbone carrier, it is very important to consider the levels of resiliency delivered with the new integrated core architecture, compared with the existing standalone arrangement. Critical considerations include:

- The physical connectivity between the serving nodes of the customer carrier and the backbone carrier should avoid single points of failure where possible.
- If the physical connectivity between the customer carrier and backbone carrier requires the use of WAN transmission links because locations are geographically separate, then suitable levels of circuit protection should be employed
- Because the backbone carrier effectively replaces the existing core topology of the customer carrier, the actual way in which backbone carrier nodes are interconnected and levels of WAN transmission protection etc., should be analysed.

All these aspects should be assessed and incorporated into the actual design process such that there is no detrimental effect on overall levels of service availability to the end customer. Service levels can be verified by reliability modeling of the new network topology, and by comparing the results with the reliability data for the existing topology.

Fault Management

There are many facets of monitoring and managing a core BGP/MPLS network in terms of assurance of service, alarm detection and filtering, customer notification of faults, and so on. In a standalone network environment, it is generally the responsibility of a particular operational team to manage faults on the network and provide service continuity during various types of failure scenarios. As shown in Figure 6, this operational function usually covers all core network elements, including PE and P (core) routers, as well as the WAN topology interconnecting the service nodes or "POPs."

In an integrated core network scenario, however, part of the customer carrier network—the P (core) routers and WAN transmission links, for example—are replaced by the NGN backbone carrier. The NGN backbone carrier has its own operational team with specific processes and systems for carrying out monitoring and management of fault events. A crucial challenge arises in terms of how to realise end-to-end fault management holistically and transparently between customer carrier and backbone carrier networks (Figure 6). Important considerations include:

- The requirement for a clear and unambiguous demarcation between customer carrier and backbone carrier core platforms must be addressed in terms of operational responsibility for specific faults and the hand-over procedures between operational domains.
- The use of existing monitoring tools and systems in both the customer carrier and backbone carrier domains must be assessed to determine whether new interfaces between such systems need to be developed to facilitate the hand-over procedures.

These topics must be factored in to determine the optimal solution for realising smooth and transparent fault-management procedures in an integrated core BGP/MPLS network environment.

Capacity Planning

As shown in Figure 6, in a standalone BGP/MPLS VPN network environment, a particular operational function exists for ongoing core capacity planning to ensure P router and WAN link capacity are suitably dimensioned to cope with current and future traffic demands. When an existing BGP/MPLS VPN network becomes a customer carrier network that is integrated with a target NGN backbone using CsC, there will be a corresponding shift in responsibility for certain aspects of core capacity planning.

VPN service traffic that would have been confined to its own dedicated core network will now be offered onto the NGN backbone carrier core network. As such, the capacity-management function for the NGN backbone carrier must use traffic planning information pertaining to the VPN services in addition to all the other service types supported on the NGN. This aggregated view of traffic demands will accelerate the core capacity dimensioning on the NGN backbone carrier network.

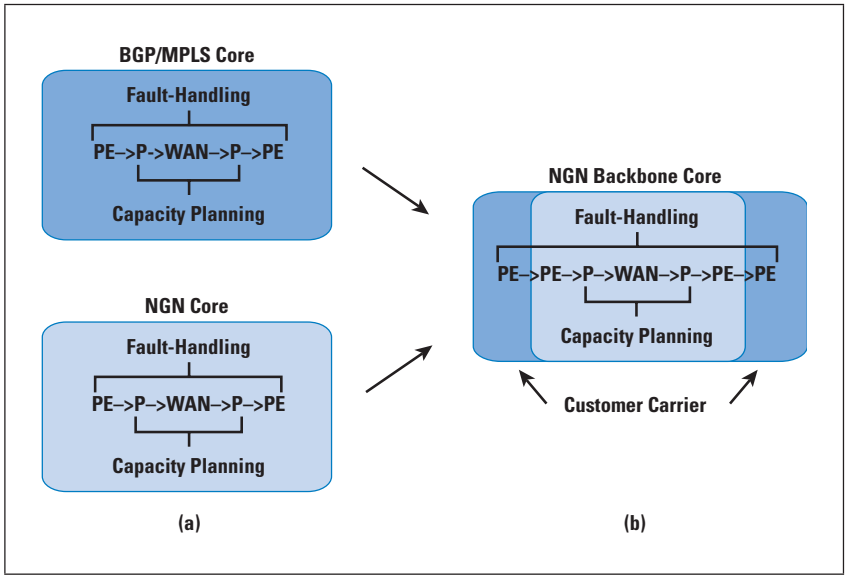


Figure 6: Fault-Management and Capacity-Planning Functions (a) Before Core Integration (b) After Core Integration with CsC

Conclusions

The MPLS-based Carrier-supporting-Carrier (CsC) framework provides network operators with a potential solution for integrating an existing BGP/MPLS VPN network, with a target all-IP based NGN. This solution should enable both capital and operational cost reduction by collapsing multiple core networks into a single NGN core domain. The article emphasised that as well as understanding the critical network architectural building blocks required to implement CsC, there are numerous critical design and operational challenges that an integrated core network presents. These challenges include how to maintain service levels and performance metrics for existing VPN customers, resiliency, fault management, and capacity planning. It is important to note, however, that in addition to the broad topic areas covered in this article, many specific additional challenges will present themselves to network operators who have implemented BGP/MPLS VPN networks, and/or NGN networks in their own specific way.

References

[1] P. Knight and C. Lewis, "Layer 2 and 3 Virtual Private Networks: Taxonomy, Technology, and Standardization Efforts," *IEEE Communications Magazine,* June 2004, pp. 124–131.

[2] E. Rosen and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)," RFC 4364, February 2006.

[3] M. Mahmoud, "Carrier-supporting-Carrier: The Whole Story (1)," *Networkers Online,* December 2008. **http://networkers-online.com/blog/2008/12/carrier-supporting-carrier-the-whole-story-1/**

[4] M. Mahmoud, "Carrier-supporting-Carrier: The Whole Story (2)," *Networkers Online,* December 2008. **http://networkers-online.com/blog/2008/12/carrier-supporting-carrier-the-whole-story-2/**

[5] J. Moy, "OSPF Version 2," RFC 2328, April 1998.

[6] L. Andersson et al., "LDP Specification," RFC 5036, October 2007.

[7] Y. Rekhter and E. Rosen, "Carrying Label Information in BGP-4," RFC 3107, January 2001.

[8] B. Davy et al., "An Expedited Forwarding PHB," RFC 3246, March 2002.

[9] J. Heinanen et al., "Assured Forwarding PHB Group," RFC 2597, June 1999.

PAUL VEITCH holds an M.Eng. and a Ph.D. from the University of Strathclyde, Glasgow. He joined BT at Martlesham Heath, Ipswich, UK, in September 1996, and worked on various aspects of broadband transmission architectures, multi-service platforms, and 3G network design. In 2000, he joined MCI-WorldCom in Cambridge, UK, and led a number of projects on IP backbone network design. In 2003, he returned to BT to work on IP VPN infrastructure design. He is currently the design authority for BT Retail's Internet networks. He can be reached at: **paul.veitch@bt.com**

PAUL HITCHEN holds a B.Eng. in Electrical and Electronic Engineering from the University of Salford. He joined BT at Martlesham Heath in September 1990 and has worked on numerous aspects of BT's data services. From 1990 to 1997 he led the development of BT's multiprotocol router portfolio, developing routing and QoS functions with BT's equipment suppliers and provided consulting to BT's customers on IP and Ethernet networks. During the same period he worked on the introduction of Frame Relay, ATM, and SMDS WAN services for BT. In 1997 he developed BT's first IP VPN service offering, working on the development and standardisation of MPLS and VPN technology. From 1997 to the end of 2006 he led the design of BT's Global MPLS Network and service, expanding the network to provide service to more than 150 countries across the world. He is currently a principal consultant working on BT's 21CN IP/MPLS network, focusing on the integration of BT's networks onto 21CN and introducing content delivery and IPTV into the network. He can be reached at: **paul.hitchen@bt.com**

MARTIN MITCHELL holds an M.Sci. from the University of Bristol and has worked for BT since 2007. He is currently an IP network designer specializing in service provider core design, Ethernet access, and network migrations. He can be reached at: **martin.3.mitchell@bt.com**

Letter to the Editor

Hi Ole,

I enjoyed the article entitled "PMIPv6: A Network-Based Localized Mobility Management Solution" in the last issue of *The Internet Protocol Journal* (Volume 13, No. 3, September 2010).

I believe that in the "Security Considerations" section it should be mentioned that the CSI (Cga & Send maIntenance) working group in the IETF is also working on updating the *Secure Neighbor Discovery* (SEND) specification (RFC 3971) to include the possibility of authenticating the proxied *Neighbor Discovery* (ND) messages sent between the terminal, the *Mobile Access Gateway* (MAG), and the *Local Mobility Anchor* (LMA). This configuration should work in addition to the proposed *IP Security* (IPsec) tunnel between the MAG and the LMA.

The reference material is available at:

https://datatracker.ietf.org/doc/draft-ietf-csi-proxy-send/

https://datatracker.ietf.org/doc/draft-ietf-csi-send-cert/

Regards,

—*Roque Gagliano, Cisco Systems*
rogaglia@cisco.com

One of the authors responds:

Dear Ole and Roque,

Thanks for reading our article and providing these valuable comments. We agree with your point. We just considered the basic security mechanisms in our article, limiting the scope to the protocols already standardized, which cover only the protection of the MAG-LMA signaling. We agree that the efforts being carried out within the CSI working group are worth mentioning with regard to the security aspects of PMIPv6.

Thanks,

—*Carlos J. Bernardos, Universidad Carlos III de Madrid*
cjbc@it.uc3m.es

Book Review

A History of the Internet

A History of the Internet and the Digital Future, by Johnny Ryan, Reaktion Books, ISBN 978 1 86189 777 0, September 2010.

Any attempt to document a 50-year history of people and activities that had such a profound and global effect as the Internet faces some challenges. Sequences are complex; written source materials are sketchy; and the many different memories conflict. Added to this reality, of course, are legitimate disagreements about intents and effects. To evaluate such writing effort means first looking for useful criteria. Here are mine: In terms of basic research, was the effort extensive, looking for multiple, appropriate sources and exploring a wide range of probing and constructive questions? Were the sources and questions interesting? This line of thinking leads to a query about the way the author integrates the resulting massive body of data. Is there an effort to develop critical analyses? Are alternative explanations explored?

Johnny Ryan's ambitious *A History of the Internet and the Digital Future* is a rather modest 246 pages, including 28 pages of references. Overall my feeling is that he does quite an interesting job of satisfying the first half of his title, but a somewhat disappointing job with the second half. His research was extensive throughout, but he takes a more critical view of the history than he does of the social aspects of our digital future. In the first half, he integrates information and reports discrepancies and curiosities. In the second half, he indulges in the common, wide-eyed wonderment that technology futurist efforts inherently risk. (Full disclosure: By way of demonstrating the thoroughness of his research, Ryan even included me as one of his many sources.)

Organization

The book is divided into three parts. Broadly, they cover origins, growth, and social effects. Ryan's use of "centrifugal" is contrasted with "centripetal" and is meant to distinguish paradigmatic tensions between approaches that centralize control versus approaches that distribute it. (Oddly, neither of these pivotal terms is in the index.) On page 8 he sets the stage:

"Three characteristics have asserted themselves throughout the Internet's history and will define the digital age to which we must all adjust: The Internet is a centrifugal force, user-driven and open."

By "centrifugal" he means moving outward, away from centralized control. For me, the terminology proved distracting, because I kept hearing my 8th-grade science teacher condescendingly explaining that there is no physics force called centrifugal. Rather it is a perception of the interaction between inertia and centripetal force.

For those with less compulsive (or effective) science teachers, the analogy might prove more helpful, because the design choice really is central to the history of networking. The tension between centralized versus distributed has marked—and continues to mark—much of the development of networking. In fact, I wish Ryan had explored its continuation as much as he explored its effect on origins.

Early History

In general, Ryan presents a narrative with fine-grained detail of the different players who played a critical role in the creation and pursuit of packet switching and then its evolution to link independent networks and technologies[1]. Efforts to take credit for the former have often become quite public and unseemly; Ryan dissects the play of actors, the essence of their technical ideas, and the details of their activities with documentation and diligence, and even uncovers some discrepancies. He develops a narrative that I found intriguing, enlightening, and credible. What I especially liked was that he explored the organizational milieu in which the activities took place. So we hear of the origins of groups such as the *Advanced Research Projects Agency* (ARPA), Lincoln Labs, and The Rand Corporation; the social and political forces that created them; and the roles they played.

Narrative Arcs

The following is really the strength of this book: It develops narrative arcs about social, political, and organizational environments and the steps taken within them that moved along the path of the Internet. It explores who, when, how, and what, both overall and in detail. At its best, the book provides comparative perspective to help the reader understand what was risky and truly innovative and thereby understand what was really challenging to develop and get adopted. As a minor example, Ryan deserves credit for his exploration and debunking of the media distortions surrounding Al Gore's role and statements concerning the Internet. Strictly speaking, debunking media excesses would not normally seem relevant to a review of the history of a technology, but Ryan uses this example for some consideration of the role of politics in the development of the Internet. The U.S. government could have chosen to assume more control over the Internet; it might have quickly turned it into a telecommunications monopoly, rather than letting it develop through independent market forces.

As would be expected for a story this sweeping, Ryan is sometimes redundant and sometimes inconsistent. Overall, the book would have benefited from more careful editing. So it has a quick reference to the "invention" of e-mail messaging at Bolt Beranek and Newman, but later has a more accurate, detailed account of Ray Tomlinson's 1971 effort, there, to add networking to the *existing* e-mail mechanism. (E-mail messaging was present on the first time-sharing systems of the 1960s, but these systems were standalone services. Tomlinson got them to talk each other.)

Another touchstone I use for discussions of Internet history is the role of the *Computer Science Network* (CSNet), because I worked on that. CSNet served as the forerunner of the larger and more obviously pivotal *National Science Foundation Network* (NSFNet). With NSFNet the Internet developed the ability to support multiple backbones—essential for a truly competitive Internet—and the market-priming creation of regional operational services, from which the seeds of the commercial Internet were sown. Ryan notes the role of CSNet as a kind of market research that led to NSFnet, and in this observation his discussion is notable. But his account of CSNet details is somewhat skewed, because CSNet is cast as having full packet-level connectivity, with e-mail-only telephone-based linkages as a secondary service. In reality full connectivity came later; the original years of CSNet were e-mail-only. Why this fact is important to note—besides overly personal fault-finding—is as a reminder that the accounting efforts for this sort of history are always noisy; the story signal is never pure, even with a diligent effort.

A further touchstone topic is the *Domain Name System* (DNS) and the development of the *Internet Corporation for Assigned Names and Numbers* (ICANN). The interesting part of this saga is later-stage Internet history, and Ryan is relatively sloppy with the details. For example, he muddles what *generic Top-Level Domains* (gTLD) already existed and what new ones were proposed, such as `.com` versus `.biz`; he also muddles the distinction between gTLDs and national domains, such as `.uk`. On the other hand, he certainly captures the continuing tone of controversy that surrounded the development and operation of ICANN, the organization now managing assignment of IP addresses and domain names.

But the most obvious, later-stage touchstone for a history like this one must be the development of the World Wide Web. Ryan gets mixed marks here. He misses the long history of open document publishing that existed even in the earlier *Advanced Research Projects Agency Network* (ARPANET), with "anonymous" FTP, and he misses that the use of *Gopher* predated the web by several years. He also misses just how complete and useful a "dynamically linked document" system Doug Englebart's NLS (computer) system provided 20 years before the invention of the web[2]. Hence, he misses the long, historical arc for publishing on the Internet. On the other hand, he does discuss *Gopher* and explores some of the reasons it lost the competition to the web. He focuses on management and intellectual property issues, whereas I tend to consider *Gopher* as having a much poorer cost/benefit mix. *Gopher* was text-only and required going down a potential long lookup tree—quite a few "clicks"—before getting any content. The web is mixed-media and can provide utility to the reader—that is, content—at each step down a lookup path. So the web is more complex to develop than *Gopher,* but it provides enough additional power and better human factors to be worth it.

Ryan's discussion of the commercial explosive growth of the Internet is a good read, including the Dutch tulip market reference and his introduction to some relevant tidbits of economics theory. However, as the book moves into "Web 2.0" and beyond, it provides reasonable descriptions of who did what to create popular new services, but his critical eye largely stops providing serious analysis. Explanations sound more like exuberance than examination. On the other hand, he certainly provides substance to the view that the Internet enables "long-tail" market opportunities to discover and satisfy specialized segments. His discussion of politicians' inventive use of the Internet is nicely concise and integrated. Again, it provides a narrative arc with substance. But his predictions for the future of users as news consumers or as citizens in political processes have too much tone of certitude and positive outcome than is justifiable in my opinion.

Worth Reading

In sum, the book is certainly worth reading. You will likely learn quite a bit, but make sure you read with glasses that have no hint of rose coloring!

References

[1] Debating which milestone marks "the beginning of the Internet" is a favorite pastime, including among those around during the period in question. Various definitions are legitimate, as long as one is clear about the choice. For me, the operational demonstration of packet switching was when the world changed, so I choose 1969 and the first four nodes of the ARPANET; or its public demonstration in 1972. TCP/IP built on this, by refining and minimizing the work to be done within the infrastructure and by linking independent networks.

[2] In the early 1970s, my job at UCLA included technical documentation and supporting online use by the Computer Science Department's secretaries. We did all our editing remotely, on the Engelbart system, because it was so powerful.

—Dave Crocker, Brandenburg InternetWorking
dcrocker@bbiw.net

Read Any Good Books Lately?

Then why not share your thoughts with the readers of IPJ? We accept reviews of new titles, as well as some of the "networking classics." In some cases, we may be able to get a publisher to send you a book for review if you don't have access to it. Contact us at **ipj@cisco.com** for more information.

Fragments

Photo: Matsuzaki Yoshinobu

Bjoern A. Zeeb Receives Second Itojun Service Award

The second Itojun Service Award was presented at the 79th meeting of the *Internet Engineering Task Force* (IETF) in Beijing, China. Bjoern A. Zeeb received the award for his dedicated work to make significant improvements in open source implementations of IPv6.

First awarded last year, the *Itojun Service Award* honours the memory of Dr. Jun-ichiro "Itojun" Hagino, who passed away in 2007, aged just 37. The award, established by the friends of Itojun and administered by the *Internet Society* (ISOC), recognises and commemorates the extraordinary dedication exercised by itojun over the course of IPv6 development.

"For many years, Bjoern has been a committed champion of, and contributor to, implementing IPv6 in open source operating systems used in servers, desktops, and embedded computer platforms, including those used by some of the busiest websites in the world," said Jun Murai of the Itojun Service Award Committee and Founder of the WIDE Project. "On behalf of the Itojun Service Award Committee, I am extremely pleased to present this award to Bjoern for his outstanding work in support of IPv6 development and deployment."

The Itojun Service Award is focused on pragmatic contributions to developing and deploying IPv6 in the spirit of serving the Internet. The award, expected to be presented annually, includes a presentation crystal, a US$3,000 honorarium, and a travel grant.

"This is a great honour, and I would like to thank the people who recommended me for the award and the committee for believing my work was valuable. I never met Itojun but he was one of the people helping me, and I have the highest respect for his massive foundational work," said Bjoern A. Zeeb. "As the Internet community works to roll out IPv6 to more and more people all around the globe, we also need to help others—developers, businesses, and users—understand and use the new Internet protocols so that the vision Itojun was working so hard for comes true."

Each Internet-connected device uses an IP address and, with the number of Internet-connected devices growing rapidly, the supply of unallocated IPv4 addresses is expected to be exhausted within the next year. To help ensure the continued rapid growth of the Internet, IPv6 provides a huge increase in the number of available addresses. And, while the technical foundations of IPv6 are well established, significant work remains to expand the deployment and use of IPv6.

For more information about the Itojun Service Award see:
`http://www.isoc.org/itojun/`

Remaining IPv4 Address Space Drops Below 5 percent

The *Number Resource Organization* (NRO) recently announced that less than five percent of the world's IPv4 addresses remain unallocated. APNIC, the Regional Internet Registry for the Asia Pacific region, has been assigned two blocks of IPv4 addresses by the *Internet Assigned Numbers Authority* (IANA). This latest allocation means that the IPv4 free pool dipped below 10% in January 2010. Since then, over 200 million IPv4 addresses have been allocated from IANA to the *Regional Internet Registries* (RIRs).

"This is a major milestone in the life of the Internet, and means that allocation of the last blocks of IPv4 to the RIRs is imminent," stated Axel Pawlik, Chairman of the NRO, the official representative of the five RIRs. "It is critical that all Internet stakeholders take definitive action now to ensure the timely adoption of IPv6."

IPv6 is the "next generation" of the Internet Protocol, providing a hugely expanded address space, which will allow the Internet to grow into the future. In 2010, the five RIRs are expected to allocate over 2,000 IPv6 address blocks, representing an increase of over 70% on the number of IPv6 allocations in 2009. In contrast, the number of IPv4 allocations is expected to grow by only 8% in 2010. These statistics indicate an absence of any last minute "rush" on IPv4 addresses, and a strong momentum behind the adoption of IPv6.

"The allocation of Internet number resources by the five RIRs enables every region in the world to benefit from fair and equitable distribution of IPv4 and IPv6 addresses. We are also actively collaborating with stakeholders at the local, regional, and global level to offer training and advice to public and private sector organisations on IPv6 adoption to ensure that everyone is prepared for IPv4 depletion and IPv6 deployment," added Pawlik.

The IANA assigns IPv4 addresses to the RIRs in blocks that equate to 1/256th of the entire IPv4 address space (each block is referred to as a "/8" or "slash-8"). The most recent assignment means that there are now only 12 of these blocks available, which is less than five percent of the entire IPv4 address pool.

The final five blocks of IPv4 addresses will be distributed simultaneously to the five RIRs, leaving only seven blocks to be handed out under the normal distribution method.

According to current depletion rates, the last five IPv4 address blocks will be allocated to the RIRs in early 2011. The pressure to adopt IPv6 is mounting. Many worry that without adequate preparation and action, there will be a chaotic scramble for IPv6, which could increase Internet costs and threaten the stability and security of the global network.

The NRO exists to protect the pool of unallocated Internet numbers (IP addresses and AS numbers) and serves as a coordinating mechanism for the five RIRs to act collectively on matters relating to the interests of RIRs. For further information, visit **http://www.nro.net**

The RIRs are independent, not-for-profit membership organizations that support the infrastructure of the Internet through technical coordination. There are five RIRs in the world today. Currently, the IANA allocates blocks of IP addresses and ASNs, known collectively as *Internet Number Resources,* to the RIRs, who then distribute them to their members within their own specific service regions. RIR members include *Internet Service Providers* (ISPs), telecommunications organizations, large corporations, governments, academic institutions, and industry stakeholders, including end users.

The RIR model of open, transparent participation has proven successful at responding to the rapidly changing Internet environment. Each RIR holds one to two open meetings per year, as well as facilitating online discussion by the community, to allow the open exchange of ideas from the technical community, the business sector, civil society, and government regulators. Each RIR performs a range of critical functions including: The reliable and stable allocation of Internet number resources (IPv4, IPv6 and *Autonymous System Number* resources); The responsible storage and maintenance of this registration data; The provision of an open, publicly accessible database where this data can be accessed. RIRs also provide a range of technical and coordination services for the Internet community. The five RIRs are:

AfriNIC:	**http://www.afrinic.net**
APNIC:	**http://www.apnic.net**
ARIN:	**http://www.arin.net**
LACNIC:	**http://www.lacnic.net**
RIPE NCC:	**http://www.ripe.net**

Find us on Facebook

In addition to *The Internet Protocol Forum,* available at **http://www.ipjforum.org**, IPJ now has its own Facebook page. Join the discussion and get the latest news and updates:

http://www.facebook.com/#!/pages/Internet-Protocol-Journal/163288673690055

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

March 2011 Volume 14, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: www.cisco.com/ipj

ISSN 1944-1134

From The Editors

In 2011 we have already seen some important Internet anniversaries and milestones. We have celebrated 25 years of IETF meetings and 40 years of the FTP protocol, but the most significant milestone took place in February when IANA handed out its final blocks of IPv4 addresses to the RIRs (see page 21). It seems like a good time to publish an edition of IPJ devoted entirely to IPv4/IPv6 transition, and to help me with this task I have invited Geoff Huston as co-editor and author for this issue, so let me hand it over to him:

There is a Chinese proverb that states: 寧為太平犬，不做亂世人 "It's better to be a dog in a peaceful time than be a man in a chaotic period." For the Internet, this year is shaping up to be a time that looks more like developing chaos than serenity and peace. The IANA has given out the last /8's, and demand has already depleted the IPv4 address stocks in the Asia Pacific. Meanwhile, the industry has discovered the mass marketing potential of mobile devices, and expects to sell and connect more than 250 million of them in 2011 alone.

The IETF designed IPv6 in the 1990s for this very reason. Its 128-bit address field is easily capable of accommodating the output of a prolific silicon manufacturing industry for many decades to come. But when we look at today's Internet, very little IPv6 can be seen. Estimates of the number of clients with functional IPv6 services hover at around 0.2 to 0.4 percent of the total.

The story about IPv6 transition technologies is complex, and there are many ways to undertake this effort. In this issue we will examine the various approaches and their relative strengths and weaknesses.

In order to send out a broad message about the need to shift online content from exclusively using IPv4 into a dual-stack world of both IPv4 and IPv6, ISOC is supporting *World IPv6 Day* on June 8. Phil Roberts explains this initiative and its role in helping the overall transition effort.

This transition is going to be difficult. It involves all parts of this diverse industry, and means combining some well-understood and widely-deployed technologies in some surprising and challenging ways. There is much to do, and we hope that this issue of IPJ provides an insight into just what the transition to IPv6 will entail.

—*Geoff Huston,* gih@apnic.net
Chief Scientist, APNIC

—*Ole J. Jacobsen,* ole@cisco.com
Editor and Publisher, IPJ

A Rough Guide to Address Exhaustion

by Geoff Huston, APNIC

The level of interest in IPv4 address exhaustion seems to be increasing, so I thought I would share some answers to the most common questions I have been asked on this topic in recent times.

What is the most significant challenge to the Internet today?

What a wonderfully open-ended question! There are so many challenges that I could identify: improving the level of security on the network, eradicating spam and viruses, improving capacity of the network infrastructure, improving the efficiency of high-speed data transfer, improving the accuracy of search engines, building more efficient and high-capacity data centers, and reducing the unit cost of Internet services, to name but a few.

If there is a common factor in many of these challenges, it is *scaling* the network to meet an ever-expanding agenda of more users, more devices, more traffic, more services, and more policies. And with more users and more forms of use come higher levels of diversity of use and greater need to replace implicit mechanisms of trust with explicit forms of trust negotiation and greater levels of demonstrable integrity of operation.

But these topics are all tactical in nature. They reflect the "how" of making the network work tomorrow by studying how to undertake marginal improvements on the network of today. However, it is not clear that the networks not just of tomorrow or next year, but a decade or more hence should reflect the usage patterns and user population of today. Perhaps a more fundamental challenge is to understand what is missing in today's network that we will need in the future.

This discussion leads to a pretty obvious challenge, at least for me. The basic currency of any network is *identifiers*. Identifiers allow the network to distinguish between clients and ensure that conversations occur between those parties who intended to communicate. In the world of packet-switched networking, such as IP, these endpoint identifiers are synonymous with the concept of an *address*. What is missing in today's network is an abundant supply of new addresses that will allow the network to scale up in size by a further factor of at least 1 million, and hopefully more than a billion-fold.

In fact, the supply of addresses is not just inadequate for future needs for a decade hence. The stock of addresses is facing imminent depletion, and the question of availability of addresses is best phrased in terms of months rather than years.

Perhaps the term "address" is somewhat of a misnomer in this context, but it may well be too late to change that now. The primary role of an IP address is not to uniquely identify the location of an endpoint of a network in relation to some positional or topographical coordinate set, but to simply uniquely identify an endpoint to distinguish it from all other endpoints. Its location is not an intrinsic property of this so-called *address*. But common convention is to call these endpoint identifiers "addresses," so I will stick with the same convention here.

So my candidate "most significant challenge for the Internet today" is that we are running out of further supply of IP addresses.

What is an IP address, and why is it so important?

One of the revolutionary changes introduced by the so-called *packet-switched* network architecture of the Internet—as compared to its telephone predecessor that used *circuit switching*—was that a massive amount of "intelligence" was ripped out of the network and placed into the devices that connect at the edge.

IP networks are incredibly simple, and at their most basic level they do very little. They are built of routers and interconnecting conduits. The function of a router is quite simple. As a packet arrives at the router from the connected circuitry (or from a wireless interface), it is divided into a common IP header and a payload. The IP header of the packet contains, among other components, two fixed-length fields: the address of the intended *destination* of the packet, and, like a postal envelope, the address of the packet creator, or the *source*. The router uses the destination address of the packet to make a routing decision as to how to dispose of the packet. For each incoming packet, the router inspects the destination address in the packet and either passes it to a connected computer if there is an address match or otherwise passes it down the *default* path to the next router. And that is a working description of the entirety of today's Internet. The important aspect here is that every connected device must have a unique address. As long as this condition is satisfied, everything else can be made to work.

In the current version of the Internet Protocol, an "address" is a 32-bit field, which can encompass some 4.4 billion unique values.

Why are we running out of addresses?

Blame silicon. Over the past 50 years, the silicon chip industry has graduated from the humble transistor of the 1950s to an astonishing industry in its own right, and the key to this silicon industry is volume.

Individual processor chips may take hundreds of millions of dollars to design, but if fabricated in sufficient volume, each processor chip may take as little as a few dollars to manufacture and distribute. The larger the production run of the silicon die, the lower the unit price of the resultant chip. We currently produce a huge volume of computers every year. In 2008 alone around 10 billion computer processors were manufactured. Although most of these microprocessors are simple 8-bit processors that are used to open doors or run elevators, a sizable proportion are used in devices that support communications, whether it is in laptop computers, smartphones, or even more basic communication applications. Typically we do not invent a new communications protocol for each new application. We recycle. And these days if we want a communications protocol for a particular application, it is easiest to simply embed the IP protocol engine onto the chip. The protocol is cheap, well tested, and it works across almost any scale we can imagine from a couple of bits per second to a couple of billion bits per second.

So it is not just the entire human population of the planet who may well have a desire to access the Internet in the future, but equally important is the emerging world of "things" that communicate. Whether it is the latest fashion in mobile phones or more mundane consumer electronics devices such as televisions or games consoles, all these devices want to communicate, and to communicate they need to have a unique identification code to present to the network, or, an "address."

We are presently turning on more than 200 million new Internet services every year, and today we have used up most of the 4.4 billion addresses that are encompassed by the IP protocol.

When will we run out?

As of September 2010, some 151 million addresses were left in the general-use pool of unallocated addresses that are managed by the central pool administration, the *Internet Assigned Numbers Authority* (IANA). The world's IP address consumption rate peaked earlier this year at a new all-time high of an equivalent rate of 243 million addresses per year.

By early February 2011 IANA handed out its last address blocks to the RIRs.

The five *Regional Internet Registries* (RIRs)[1] still had pools of addresses available for general use at that time, but from that point, as they further run down their local pools, the IANA is now unable to provide any more addresses to replenish them. The Asia Pacific Regional Registry, APNIC, has been experiencing the highest level of demand in the world, accounting for some two-thirds of all addresses consumed in early 2011. APNIC exhausted its general use IPv4 address pool in April 2011.

Although the current models of address consumption show that the other regions will be able to manage available address pools for a few more months, this prediction does not account for the multi-national nature of many of the largest of the service providers, and at this stage it is not known how much address-consumption pressure will shift outward from APNIC to the other RIRs now that APNIC's available address pool is effectively drained. So it may well be that 2011 will see IPv4 addresses cease to be generally available in many parts of the world, and by early 2012 there will be no further generally available IPv4 addresses in Europe, North America and Asia.

What is the plan?

This news of imminent exhaustion of the supply of addresses is not a surprise. Although the exact date of predicted address exhaustion has varied over time, the prospect of address exhaustion was first raised in technical circles in August 1990, and work has been undertaken since that time to understand what might be possible and how that could be achieved.

The 1990s saw an intense burst of engineering activity that was intended to provide a solution for this forthcoming address problem. The most significant outcome of this effort was the specification of a successor IP protocol to that of IPv4, called IP Version 6 or *IPv6*.

Why IPv6 and not IPv5?

It would be reasonable to expect the successor protocol of IP Version 4 to be called IP Version 5, but as it turned out Version 5 of the Internet Protocol Family was already taken. In the late 1980s the Internet Protocol itself was the topic of a considerable level of research, as researchers experimented with different forms of network behavior. Version 5 of the Internet Protocol was reserved for use with an experimental IP protocol, the *Internet Stream Protocol, Version 2* (ST-II), written up as RFC 1190 in 1990. When it came time to assign a protocol number of the "next generation" of IPv4, the next available version number was 6, hence IPv6.

The outcome of this process was a relatively conservative change to the IP protocol. The major shift was to enlarge the address fields from 32 bits to 128 bits in length. Other changes were made that were thought to be minor improvements at the time, although hindsight has managed to raise some doubts about that!

The design intent of IPv6 is a usable lifetime of more than 50 years, as compared with a "mainstream" deployment lifetime of IPv4 of 15 years, assuming that you are prepared to draw a line at around 1995 and claim that at that time the protocol moved from an interesting academic and research project to a mainstream pillar of the global communications industry.

That 50 years of usable life for IPv6 is admittedly very ambitious, because it is intended to encompass a growth of the ubiquity of silicon from the current industry volumes of hundreds of millions of new connected devices every year to a future level of activity that may encompass in the order of hundreds of billions to possibly some trillions of new connected devices every year.

So the technical plan to address the address-exhaustion problem was to perform an upgrade of the Internet and convert the Internet from IP Version 4 to IP Version 6.

Nothing else needs to be changed. This change is not intended to be radical or revolutionary. The change from circuit switching to packet switching was a revolutionary change for both the communications industry itself and for you and me as enthusiastic communicators. The change from IPv4 to IPv6 is intended to be a polar opposite, and at best it is intended to be a transparent and largely invisible transition. E-mail will still be e-mail. The web should still look just as it always did, and anything that works on IPv4 is expected to work on IPv6. IPv6 is not inherently any faster, nor any cheaper, nor is it even all that much better. The major change in IPv6 is that it supports a much larger address field.

How many addresses are in IPv6?

In theory, there are 2 to the power 128 unique addresses in IPv6—a very large number. If each IPv6 address were a single grain of sand, the entire IPv6 address space would construct 300 million planets, each the size of the earth!

But theory and practice align only in theory. In practice the IPv6 address plan creates a usable span of addresses that encompasses between 2 to the power 50 and 2 to the power 60 devices. Although this number is nowhere near 2 to the power 128, it is still a range of numbers that are between 1 million to 1 billion times the size of the IPv4 address space.

How do we transition to IPv6?

Unfortunately IPv6 is not "backward-compatible" with IPv4. Backward compatibility would allow for a piecemeal transition, where IPv6 could be regarded as a fully functional substitute for IPv4, so that the existing network base would keep using IPv4 forever, while the most recent devices would use IPv6 and all devices could communicate with each other. The lack of such backward compatibility implies that this communication is simply not possible. IPv4 and IPv6 are distinct and different communications protocols, in the same way that English and, say, German are distinct and different languages.

Attempts have been made to design various forms of automated protocol translator units that can take an incoming IPv4 packet and emit a corresponding IPv6 packet in the same manner as a language interpreter. However, this approach also has some major limitations, so it is usable only in very carefully constrained contexts.

The implication of this lack of backward compatibility and inability to perform automated translation within the network is that if we want to preserve comprehensive any-to-any connectivity during the transition, we have to equip each device that is performing a transition with both protocol stacks, or, in effect, allow the device to become "bilingual," and conduct a conversation in either IPv4 or IPv6, as required. This transition has been termed a *dual-stack* transition.

When my computer supports IPv6, can I return my IPv4 address?

Each device needs to maintain its capability to converse using IPv4 while there are still other devices out there that remain IPv4-only. So a device that becomes IPv6-capable cannot immediately give up its IPv4 address. It will need to keep this IPv4 capability and operate in dual-stack mode for as long as there are other devices and services out there that are reachable only using IPv4.

The implication of this constraint is that we will need to add dual-stack devices to the Internet and consume both IPv4 and IPv6 addresses during this transition.

So, no, you will need to keep your IPv4 address for as long as there are folk out there with whom you want to communicate who have not also migrated to be a dual-stack IPv4- and IPv6-capable entity.

What needs to be done to transition the network to IPv6?

What is encompassed in "transition?" Do all *Internet Service Providers* (ISPs) have to decide when and how to reprogram their systems and reconfigure their routers, switches, and middleware? Will they need to replace all their customers' modems with ones that support IPv6? What is the agenda?

This level of uncertainty about the transition to IPv6 is evidently widespread in today's Internet. Most of the actors in the Internet are unsure about what needs to be done, from the largest of the service providers down to individual end users. Yes, it appears to be a simple matter of reprogramming devices from being just IPv4-capable to being capable of supporting both IPv4 and IPv6, but it is not quite so simple. Dual-stack operation is not easy, nor will it just happen without any form of applied impetus. Imagine that this transition is from everyone on the planet speaking Latin to each other to everyone speaking Esperanto. If this situation were a simple matter of everyone stopping using one language and being rebooted to use the other language one by one, then imagine the plight of the first people to undertake this transition—from being connected and being able to communicate with everyone else using Latin, these first users would find themselves speaking exclusively Esperanto to ... nobody! They would in effect have been disconnected from the network.

So the transition is a little trickier than just turning a big switch from IPv4 to IPv6. Because this transition is a piecemeal and fragmentary one, each device, each router, each firewall, each load server, and all those other components of the network service platform need to be programmed with an additional protocol, and become, in effect, bilingual. And in this case there are no magic interpreters that can "translate" between IPv4 and IPv6. So it is only when the entire network is bilingual in a dual-stack mode that we can turn off IPv4 and consider the transition to be complete.

For an extended period of time the Internet is going to have to operate as two Internets. We have never tried that type of operation before, at least not on a grand scale as this one; in fact, it has often been likened to replacing the jet engines of an airplane while the plane is in flight. Somehow we now have to not only sustain a growth rate of at least some 250 million new connections per year, but at the same time retrofit IPv6 to the existing installed base while continuing to support IPv4. The complexity of this operation is significant, and there is considerable confusion about what to do, when to do it, how much it will all cost, and who will pay. So yes, we are all unsure about what needs to be done.

How long do we expect this dual-stack transition to take?

If only we knew! The Internet today encompasses some 1.7 billion users, and hundreds of millions of devices out there are configured to "talk" only IPv4. Some of these devices will surely die in the coming years, and others may be upgraded or reprogrammed, but others will persist in operation for many years to come while continuing to speak only IPv4. Even looking at what is being sold today, although many general-purpose computers (or at least their operating systems) are now configured to operate in dual-stack mode, when you look at embedded devices such as *Digital Subscriber Line* (DSL) or cable modems, or firewalls, or a myriad of other devices that are integral to the operation of today's Internet, many of these devices are still configured in firmware to operate exclusively using IPv4.

Some modeling of the transition process has projected an 80-year transition process. That projection is heading into the realms of the absurd, given that our expectations for the operational lifespan of IPv6 have a lower bound of just 50 years or so. However, given the sheer scope of the conversion task and the current level of penetration of IPv6 to levels of between 2 and 5 percent of today's Internet, and given that a deadline of 2 years from now implies a conversion rate of in excess of 1 million devices every day in that 2-year span. It seems that an expectation that this transition could be substantially completed in as little as 2 years also strikes an unrealistic note.

So a more realistic assumption is that we will probably take around 5 years to complete this transition, and we will need to operate the Internet in dual-stack mode with both IPv4 and IPv6 across this entire period.

But at the current level of Internet growth, the IPv4 address pool cannot sustain a further 5 years of growth—at least not with the current amount of unallocated addresses remaining in the allocation pools. The current address-consumption rate is some 250 million addresses per year. The depleted IPv4 address pool simply cannot withstand the pressures of a 5-year transition without a radical change to the model of the IPv4 network. And if we need to rework the model of the IPv4 network simply to sustain a transition to IPv6, then can't we simply get going with IPv6 a little more quickly instead?

However, "fully depleted" or even "run out" is perhaps not the most appropriate way to describe what will happen to IPv4 addresses in the coming months. It is probably more accurate to say "unobtainable at the current prices." When the current orderly process of allocation of IPv4 addresses comes to an end, that does not mean that IPv4 addresses will be completely unobtainable. In this world many things that are scarce are still obtainable—for a price. It is quite reasonable to anticipate that for as long as there is still a demand for IPv4 addresses there will be some form of "aftermarket" where addresses are traded for money. However, as with many markets, what is not possible to predict is the price for addresses that will be established by such a market-based address-trading regime.

What about "address sharing" in IPv4? Why do we need IPv6, given that we could simply share addresses in IPv4?

Yes, of course address sharing[2] is an option, and we have been doing it for many years already in IPv4. But is it a viable substitute for IPv6?

As part of the engineering effort to develop a successor protocol to IPv4 in the mid 1990s, the IETF published a novel approach of *address sharing,* which we call today *Network Address Translation,* or NAT.[3] These days almost every DSL modem, and other forms of customer connection equipment, comes equipped with NAT functions. Today most Internet Service Providers give their subscribers a single IPv4 address. At home I have a single IPv4 address, and you probably do too. But in my home I have about 20 connected devices of various sorts (I am counting TiVo units, game consoles, televisions, printers, and such, because they are all in essence Internet-connected devices, and I believe that my situation is not unusual). All these devices "share" the single external IP connection, so all of them "share" this single IPv4 address.

But address sharing has its limitations. When a single household shares a single address, nothing unusual happens. But If I were to try to do the same address-sharing trick of using a single IP address to share across, say 2,000 customers, I would cross over into a world of pain. Many applications today gain speed through parallelism, and they support parallelism through consuming port addresses.

Each IP address can support the parallel operation of 65,535 sessions, using a 16-bit *port identifier* as the distinguishing identifier. But when address sharing is used, these ports are shared across the number of devices that are all sharing this common address. When 2,000 customers are sharing a single address and each customer has some 20 or so devices, then the average number of port addresses per device is 1.5. Common applications that exploit parallel operation include such favorites as *Gmail, Google Maps,* and *iTunes*. With a sufficiently constrained number of available ports to use, these applications would cease to work. Indeed, many network applications would fail, and at a level of a single address shared across 2,000 households, I would guess that up to half of these 2,000 customers would not have a working Internet at any single point in time.

Our experience suggests that address sharing works only up to a point, and then it breaks everything badly. We are already address sharing at the level of sharing a single address per household, and households are these days buying more connected devices of various sorts, not *fewer*. So attempting to share that single address across more than one household is at best a temporary solution, and is not a sustainable option that is an alternative to IPv6.

So we need to transition to IPv6, and we need to do so within an impossibly short time.

This discussion all sounds like a terrible problem.
Was this global "experiment" with the Internet all one big mistake?
Should we have looked elsewhere for a networking technology back in the 1990s?

The IP address problem is—for me at any rate—a fascinating one. At the time when researchers were working on the specifications for the Internet Protocol in the 1970s, they decided to use fixed-length 32-bit fields of the interface identifier addresses in the protocol. This decision was a radical one at the time. Contemporary network protocols, such as *DECnet Phase III,* used 16-bit address lengths, and 8-bit addresses were also very common at the time. After all, computers were so big and expensive, who could possibly afford more than 256 unique devices in a single network? Eight bits for addresses was surely enough! Using 32 bits in the address field was not an easy decision to make, because there was constant pressure to reduce the packet headers in order to leave more room for the data payload, so to reserve such a massive amount of space in the address fields of the protocol header to allow two 32-bit address fields was a very bold decision.

However, it was a decision that has proved to be very robust. TCP/IP has sustained the Internet from a mere handful of warehouse-sized computers running at mere kilobits per second to today, where probably more than 3 billion devices connect to the Internet in one way or another, at speeds that range from a few hundred bits per second to a massive 100 Gbps—all talking one single protocol that was invented more than 30 years ago.

IP has demonstrated a scale factor 1 billion! In my mind that achievement demonstrates a level of engineering foresight that is truly phenomenal. So in some sense the underlying observation here is not that IPv4 is running out of addresses today, but that it has been able to get to today at all!

Given that IPv4 has been able to scale by a factor of 1 billion, then if we can make IPv6 scale by a further factor of 1 billion from today we will have done well.

Disclaimer

The views expressed in this article do not necessarily represent the views or positions of the *Asia Pacific Network Information Centre* (APNIC).

References

[1] Daniel Karrenberg, Gerard Ross, Paul Wilson, and Leslie Nobile, "Development of the Regional Internet Registry System," *The Internet Protocol Journal,* Volume 4, No. 4, December 2001.

[2] Geoff Huston, "NAT++: Address Sharing in IPv4," *The Internet Protocol Journal,* Volume 13, No. 2, June 2010.

[3] Geoff Huston, "Anatomy: A Look inside Network Address Translators," *The Internet Protocol Journal,* Volume 7, No. 3, September 2004.

GEOFF HUSTON, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of numerous Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005; he served on the Board of Trustees of the Internet Society from 1992 until 2001. E-mail: **gih@apnic.net**

World IPv6 Day

by Phil Roberts, ISOC

On June 8, 2011, websites including *Google, Facebook, Yahoo!,* and *Bing* will make their main webpages reachable over IPv6 for a 24-hour period from 00:00 to 23:59 *Coordinated Universal Time* (UTC). This activity, *World IPv6 Day,* a "test flight" of IPv6, is motivating organizations across the Internet industry to prepare their services for IPv6, the next generation of the Internet Protocol. Internet Service Providers, hardware makers, operation system and application vendors, and other websites are indeed working to make this activity of testing IPv6 on an Internet scale successful.

The Internet is a never-ending exercise in collaboration. Making a successful transition to IPv6 is one of the major challenges facing the Internet today. Although IPv6 is used extensively in many large networks today, the World IPv6 Day activity is acting as a focal point to bring together all parts of the Internet industry to accelerate deployment of IPv6 in all parts of the Internet.

For some time the deployment of IPv6 has faced a "chicken-and-egg problem." Website owners have been reluctant to deploy IPv6 because there were not many end users to view their webpages over IPv6. Network operators have been hesitant to deploy IPv6 for many end users because there were few places for those users to view content over IPv6. That the most popular websites in the world according to Alexa rankings are deploying IPv6 on their main webpages is a clear indication that the Internet industry is moving beyond this long-standing impasse. Although June 8 is a 24-hour test, it is clear that this is a move toward regular operation of IPv6, and network operators can confidently roll out IPv6 to end users knowing that the Internet industry is making a concerted effort to make IPv6 an operational reality.

Today, IPv6 connectivity concerns provide another disincentive for a major website to enable IPv6 for regular operation. Badly configured or poorly behaving implementations may prevent end users from reaching a major website that enables IPv6 on its main page. It is currently estimated that this problem will affect only a minor percentage of end users—at the time of the announcement of World IPv6 Day, the estimate was that only 0.05 percent of end users would experience difficulties.

Although this percentage is small, it is potentially a very large number of end users for a website that has visitors numbering in the tens of millions (or more). It is simply impossible from a business point of view for a website of this magnitude to deploy IPv6 alone when this many users could be affected. The users who would not be able to get to that website will simply go to another website in search of similar services.

However, because several such websites have agreed to do this testing at the same time, and for the same duration, individual end users who experience disruption of their connectivity by IPv6 may be able to determine that the problem they are experiencing is indeed not a problem with a set of major websites but may, in fact, be a problem in their own host or network, and will provide an incentive for them to take steps to determine the source of the problem and repair it.

Website owners, network operators, and hardware and software vendors are collaborating to minimize these effects leading up to World IPv6 Day. All of these organizations are working to provide tools to detect these problems and offer suggested fixes in advance of June 8. The test site **http://test-ipv6.com/** allows end users today to test their connectivity and determine whether their connectivity to websites will be affected when those websites enable IPv6.

Some websites have already performed a similar 24-hour test. Last year, the German online news site Heise (**http://www.heise.de**) conducted a similar experiment. The site enabled IPv6 on its main page for 24 hours, turned it off, examined the effects of the experiment, and then permanently enabled IPv6 on its main page. Two major websites in Norway did a similar test, and they also have enabled IPv6 permanently. An activity like this for many websites is clearly a step toward regular and normal IPv6 operations. Website owners will, of course, determine when it makes sense for their business to make IPv6 operations available permanently.

Since the announcement of World IPv6 Day, many other websites from around the world have indicated that they are deploying IPv6, and many of those have decided to join in the global IPv6 test on June 8. The list of websites includes major websites such as *Google, Facebook,* and *Yahoo!* and very small websites with small numbers of visitors. It is exciting that websites from every inhabited continent plan to participate. Major websites from the Czech Republic, Portugal, Brazil, and Japan, for example, are joining this test, with more websites joining every day.

For further information about World IPv6 Day, please visit:
http://www.isoc.org/wp/worldipv6day

There you will find details about the websites that will be turning on IPv6 on June 8, how to join, and information for networks and individuals, including an FAQ.

PHIL ROBERTS joined the Internet Society (ISOC) in 2008. Prior to that he spent several years with Motorola in research and product development, all in the area of mobile broadband systems. He has been active in the IETF for more than a decade. He can be reached at: **roberts@isoc.org**

Transitional Myths

by Geoff Huston, APNIC

Last October, I attended the *Réseaux IP Européens* (RIPE)[1] meeting in Rome, and—not unexpectedly for a group that has some interest in IP addresses—the topic of IPv4 address exhaustion, and the related topic of the transition of the network to IPv6, captured a lot of attention throughout the meeting. One session I found particularly interesting was on the transition to IPv6, where people related their experiences and perspectives on the forthcoming transition to IPv6.

I found the session interesting, because it exposed some commonly held beliefs about the transition to IPv6, so I will share them here, and discuss a little about why I find them somewhat fanciful.

Myth 1: "We have many years for this transition."

No, I don't think we do!

The Internet is currently growing at a rate that consumes some 200 million IPv4 addresses every year, or 5 percent of the entire address IPv4 pool. This growth rate reflects an underlying growth of service deployment by the same order of magnitude of some hundreds of millions of new services activated per year. Throughout a dual-stack transition, all existing services will continue to require IPv4 addresses, and all new services will also require access to IPv4 addresses. The pool of unallocated addresses was exhausted in February 2011, and the *Regional Internet Registries* (RIRs)[2] will exhaust their local pools commencing early 2011 and through 2012. When those pools exhaust, then all new Internet services will need access to IPv4 addresses as part of the IPv4 part of the dual-stack environment, but at that point there will be no more freely available addresses from the registries. Service providers have some local stocks of IPv4 addresses, but even those stocks will not last for long.

As the network continues to grow, the pressure to find the equivalent of a further 200 million or more IPv4 addresses each year will become acute—and at some point will be unsustainable. Even with the widespread use of *Network Address Translators* (NATs)[3] and further incentives to recover all unused public address space, the inexorable pressure of growth will cause unsustainable pressures on the supply of addresses.

It is unlikely that we can sustain 10 more years of network growth using dual stack, so transition will need to happen faster than that. How about 5 years? Even then, at the higher level of growth forecasts, we will still need to flush out the equivalent of 1.5 billion IPv4 addresses from the existing user base to sustain a 5-year transition, and this number seems to be a stretch target. A more realistic estimate of transition time, in terms of accessible IPv4 addresses from recovery operations, is in the 3–4 year timeframe, and no longer.

So no, we do not have many years for this transition. If we are careful—and a bit lucky—we will have about 4 years.

Myth 2: "It is just a change of a protocol code. Users will not see any difference in the transition."

If only that were true!

In an open market environment, scarcity is invariably reflected in price. For as long as this transition lasts, this industry is going to have to equip new networks and new services with IPv4 addresses, and the greater the scarcity pressure on IPv4 addresses, the greater the scarcity price of IPv4 addresses. Such a price escalation of an essential good is never a desirable outcome, and although numerous possible measures can be taken to mitigate the problem, to some extent or other, the scarcity pressure and the attendant price escalation suggest a reasonable expectation of some level of price pressure on IPv4 addresses.

In addition, an *Internet Service Provider* (ISP) may not be able to rely solely on customer-owned and-operated NATs to locally mask out some of the incremental costs of IPv4 address scarcity. It is likely—and increasingly so the longer the transition takes—that the ISP will also have to operate NATs. The attendant capital and operational costs of such additional network functions will ultimately be borne by the service provider's customer base during the transition.

But it is not just price that is affected by this transition—network performance may also be affected. Today a connection across the Internet is typically made by using the *Domain Name System* (DNS) to translate a name to an equivalent IP address, and then launching a connection-establishment packet (or the entire query in the case of the U*ser Datagram Protocol* [UDP]) to the address in question. But such an operation assumes a uniform single protocol. In a transition world you can no longer simply assume that everything is contactable with a single protocol, and it is necessary to extend the DNS query to two queries, one for IPv4 and one for IPv6. The client then needs to select which protocol to use if the DNS returns addresses in both protocols. Then there is the tricky problem of failover. If the initial packet fails to elicit a response within some parameter of retries and timeouts, then the client will attempt to connect using the other protocol with the same set of retries and timeouts. In a dual-stack transitional world, not only does failure take more time to recognize, but even partial failure may take time.

So users may see some changes in the Internet. They may be exposed to higher prices that reflect the higher costs of operating the service, and they may see some instances where the network simply starts to appear "sluggish" in response.

Myth 3: "NAT upon NAT upon NAT will work."

Maybe. But maybe not all the time, and maybe not in ways that match what happens today.

The Internet has been operating for more than a decade now with a very prevalent model of a single level of address translation in the path. Application designers now assume its existence, and also make some other rather critical assumptions, notably that the NAT is close to the client in a client-server world, and that there is a single NAT in the path, and that its particular form of address translation behavior can be determined with numerous probe tests. There is even a client-to-NAT protocol to assist certain applications to communicate port-binding preferences to the local NAT. In a multilevel NAT world, such assumptions do not directly translate, but it is not necessarily the case that the application is aware of the added NATs in the end-to-end path.

However, it is not just the added complexity of the multipart NAT that presents challenges to applications. The NAT layering is intended to create an environment where a single IP address is dynamically shared across multiple clients, rather than being assigned to a single client at a time. Applications that use parallelism extensively by undertaking concurrent sessions require access to a large pool of available port addresses. Modern web browsers are a classic example of this form of behavior. The multiple NAT model effectively shares a single address across multiple clients by using the port address, effectively placing the pool of port addresses under contention. The higher the density of port contention, the greater the risk that this multiple layering of NATs will have a visible effect on the operation of the application.

There is also a considerable investment in the area of logging and accountability, where individual users of the network are recorded in the various log functions through their public-side address. Sharing these public addresses across multiple clients at the same time—as is the intended outcome of a multilayer NAT environment—implies that the log function is now forced to record operations at the level of port usage and individual transactions. Not only does this reality have implications in terms of the load and volume of logged information, there is also a tangible increase in the level of potential back tracing of individual users' online activities if full port usage logging were to be instituted, with the attendant concerns that this back tracing represents an inappropriate balance between accountability and traceability and personal privacy. It is also unclear whether there will be opportunity to have any public debate on such a topic, given that the pressure to deploy multilevel NAT is already visible.

Myth 4: "Changing the Customer Premises Equipment (CPE) is easy."

No, not necessarily.

I think we have all seen many transition plans, including multilevel Version 4 NATs, NATs that perform protocol translation between IPv4 and IPv6, NATs plus tunneling, as in *Dual-Stack Lite,* the *IVI Bi-direction Mapping Gateway, 6to4, 6RD,* and *Teredo,* to call up but a few of the various transitional technologies that have been proposed in recent times. (See the article "Transitioning Protocols" starting on page 22.)

All approaches to dual-stack transition necessarily make changes to some part of the network fabric, whether it is changes to the end systems to include an IPv6 protocol stack in addition to an IPv4 stack, or the addition of more NATs, or gateways into the network infrastructure. Of course, within a particular transitional model there is a selective choice as to what elements of the infrastructure are susceptible to change and what elements are resistant to change. Some models of transition, such as 6RD and Dual-Stack Lite, assume that changing the CPE is easy and straightforward, or at least that such a broad set of upgrades to customer equipment is logistically and economically feasible. 6RD contains an implicit assumption that the network operator has no economic motivation to alter the network elements, and wishes to retain a single protocol infrastructure that uses IPv4.

Where the CPE is owned, operated, and remotely maintained by the service provider, upgrading the image on the CPE might present fewer obstacles than upgrading other elements of the network infrastructure, such as broadband remote-access servers that operate in a single protocol mode, but sweeping generalizations in this industry are unreliable. Service providers tend to operate customized cost models, and appear to be operating with specialized mixes of vendor equipment and operational support systems. For this reason operators tend to have differing perspectives on what component of their network is more malleable, and correspondingly have differing perspectives on which particular transition technology suits their particular environment.

This industry is volume-based, where an underlying homogeneity of the deployed technology—and economies of scale and precision of process—are critical components of reliable and cost-efficient rollouts. It is somewhat unexpected to see this transition expose a relative high degree of customization and diversity in network service environments.

Myth 5: "My ISP has enough IPv4 addresses to last for years, so it does not have a problem."

Well, not necessarily.

The assumption behind this statement is that everyone else is also able to persist with IPv4, and everyone you wish to reach, and every service point you wish to access, will maintain some form of connectivity in IPv4 indefinitely.

But this assumption is not necessarily valid. At the point in time when a significant number of clients or services cannot be adequately supported on IPv4, then irrespective of how many IPv4 addresses ISPs have, they will need to provide their clients with IPv6 in order to reach these IPv6-only services On a network, the actions of others directly affect your own local actions. So if you believe that you need do nothing, and you can use an IPv4 service for years into the future, then this position will be inadequate at the point in time when a significant number of others encounter critical levels of scarcity such that they are incapable of sustaining the IPv4 side of a dual-stack deployment, and are forced to deploy an IPv6-only service. The greater the level of address hoarding, the greater the level of pressure to deploy IPv6-only services on the part of those service providers who are badly placed in terms of access to IPv4 addresses.

Myth 6: "We will always have to run IPv4 protocols."

Probably not.

Or at least not in terms and volumes that are significant to the industry over the forthcoming decades. Protocols do die. DECnet and *Systems Network Architecture* (SNA) no longer exist as widely deployed networking protocols. In particular, networking in the public space is all about any-to-any connectivity, and to support this connectivity we need a common protocol foundation. In terms of the dynamics of transition, this situation is more about tipping points of the mass of the market than it is about sustained coexistence of diverse protocols. When a new technology—or in this case, protocol—achieves a critical level of adoption, the momentum switches from resisting the change to embracing it.

The aftermath of such transitions does not leave a legacy of enduring demand for the superseded technology. As difficult as it is to foresee today, when the industry acknowledges that the new technology achieves this critical mass of adoption, the dynamics of the networking effect propels the industry into a tipping point where the remainder of the transition is likely to be both inevitable and comprehensive. The likely outcome of this situation is that there is no residual significant level of demand for IPv4.

Myth 7: "There is a technology that will translate between IPv4 to IPv6."

Yes, but...

Such a technology effectively maps between IPv4 and IPv6 addresses. One approach, the *IVI Bi-direction Mapping Gateway,* provides a 1:1 mapping by embedding fields of one address in the other. Another approach, originally termed *Network Address Translator – Protocol Translator* (NAT-PT), uses a mapping table in a fashion similar to a conventional NAT unit. The common constraint here is that if there are no IPv4 addresses, then such a bidirectional mapping cannot be sustained in each approach. Ultimately, if every packet that traverses the public Internet requires public address values in the source and destination fields, and the ISP must provide a protocol bridge between IPv4 and IPv6, then public IPv4 addresses are required.

But it is not just the requirement for continued access to addresses that is the critical concern here. A reading of RFC 4966[4], "Reasons to Move the Network Address Translator – Protocol Translator (NAT-PT) to Historic Status" should curb any untoward enthusiasm that this approach is capable of sustaining the entire load of this dual-stack transition without any further implications or problems.

Myth 8: "We do not necessarily have to transition to IPv6. There are substitutes."

Nothing is visible from here!

If we want to continue to operate a network at the price, performance, and functional flexibility that is offered by packet-switched networks, then the search for alternatives to IPv6 is necessarily constrained to a set of technologies that offer approaches that are—at a suitably abstract level—isomorphic to IP. But going from abstract observations to a specific protocol design is never a fast or easy process, and the lessons from the genesis of both IPv4 and IPv6 point to a period of many years of design and progressive refinement to develop a viable approach. In our current context any such redesign is not a viable alternative to IPv6, given the timeframe of IPv4 address exhaustion. It is unlikely that such an effort would elicit a substitute to IPv6, and it is more likely that such an effort may lead toward an inevitable successor to IPv6, if we dare to contemplate networking technologies further into the future.

Other approaches exist, based on application-level gateways and similar forms of mapping of services from one network domain. We have been there before in the chaotic jumble of networks and services that defined much of the 1980s, and it is a past that I for one find easier to forget! Such an outcome is of considerably higher complexity, considerably less secure, harder to use, more expensive to operate, and more resistant to scaling.

Like it or not, the pragmatic observation of today's situation is that we do not have a viable choice here. No viable substitutes exist.

Myth 9: "We know what is happening."

I am not sure that is universally true! The comments I have heard about the current situation lead me to the observation that there are many different perspectives on the situation. Individuals perceive the transition in terms that relate to their own circumstances and their own limitations, and a more encompassing perspective of the entire Internet and this transition is harder to assemble. So, from the perspective of the Internet as a whole, no, we are not really aware of what is happening.

Myth 10: "We know what we are doing."

Individually this statement is, hopefully, true. But at the level of the entirety of the Internet, no, we do not really have a clear perspective of this transition.

Myth 11: "We have a plan!"

See the comment for myth 10.

Myth 12: "The Internet will be fine!"

I am unsure about this one.

The worrying observation is that the Internet has so far thrived on diversity and competition. We have seen constant innovation and evolution on the Internet, and the entrance of new services and new service providers.

But if we rely solely on IPv4 for the future Internet, then this level of competition and diversity will be extremely challenging to sustain. If we lose that impetus of competitive pressure from innovation and creativity, then the Internet will likely stagnate under the oppression of brutal volume economics. The risks of monopoly formation under such conditions are relatively high.

I hope one observation I heard at the RIPE session will be a myth as this transition gets underway:

> *"The incumbents will have all the IPv4 space.*
> *Thanks for playing!"*

If that is *not* a myth, then we are going to be in serious trouble!

Disclaimer

The views expressed in this article do not necessarily represent the views or positions of the *Asia Pacific Network Information Centre* (APNIC).

References

[1] **http://www.ripe.net/ripe/meetings/ripe-61/**

[2] Daniel Karrenberg, Gerard Ross, Paul Wilson, and Leslie Nobile, "Development of the Regional Internet Registry System," *The Internet Protocol Journal,* Volume 4, No. 4, December 2001.

[3] Geoff Huston, "Anatomy: A Look inside Network Address Translators," *The Internet Protocol Journal,* Volume 7, No. 3, September 2004.

[4] Cedric Aoun and Elwyn Davies, "Reasons to Move the Network Address Translator – Protocol Translator (NAT-PT) to Historic Status," RFC 4966, July 2007.

Pool of Unallocated IPv4 Addresses Now Completely Emptied

On February 3, 2011 a critical point in the history of the Internet was reached with the allocation of the last remaining IPv4 Internet addresses from a central pool. It means the future expansion of the Internet is now dependant on the successful global deployment of the next generation of Internet protocol, called IPv6.

The announcement was made by four international non-profit groups, which work collaboratively to coordinate the world's Internet addressing system and its technical standards. At a news conference in Miami, Florida, the *Internet Corporation for Assigned Names and Numbers* (ICANN) joined the *Number Resources Organization* (NRO), the *Internet Architecture Board* (IAB) and the *Internet Society* (ISOC) in announcing that the pool of first generation Internet addresses has now been completely emptied. The final allocation of Internet addresses was administered by the *Internet Assigned Numbers Authority* (IANA), which is a function of ICANN.

"This is a major turning point in the on-going development of the Internet," said Rod Beckstrom, ICANN's President and Chief Executive Officer. "No one was caught off guard by this. The Internet technical community has been planning for IPv4 depletion for some time. But it means the adoption of IPv6 is now of paramount importance, since it will allow the Internet to continue its amazing growth and foster the global innovation we've all come to expect."

Two "blocks" of the dwindling number of IPv4 addresses—about 33 million of them—were allocated in late January to APNIC, the *Regional Internet Registry* (RIR) for the Asia Pacific region. When that happened, it meant the pool of IPv4 addresses had been depleted to a point where a global policy was triggered to immediately allocate the remaining small pool of addresses *equally* among the five global RIRs.

"It's only a matter of time before the RIRs and *Internet Service Providers* (ISPs) must start denying requests for IPv4 address space," said Raúl Echeberría, Chairman of the NRO, the umbrella organization of the five RIRs. "Deploying IPv6 is now a requirement, not an option."

Transitioning Protocols

by Geoff Huston, APNIC

In the previous article, I looked at some common myths associated with the transition to IPv6. In this article I would like to look behind the various opinions and perspectives about this transition, and examine in a little more detail the nature of the technologies being proposed to support the transition to IPv6.

After some time of hearing dire warnings about the imminent exhaustion of the stocks of available IPv4 address space, we have now achieved the first milestone of address exhaustion, the depletion of the central pool of *Internet Assigned Numbers Authority* (IANA)-managed address space. The last five /8s were handed out from IANA to the *Regional Internet Registries* (RIRs) on February 3, 2011. After some years of industrywide general inattention and inaction with IPv6, perhaps it is not unexpected to now see a panicked response along the lines of "Maybe we should do something now!"

But what exactly should be done? It is one thing to decide to "support" IPv6 in a network, but quite another to develop a specific plan, complete with specific technologies, timelines, costs, vendors, and a realistic assessment of the incremental risks and opportunities. Although working through some of this detail has the normal levels of uncertainty that you would expect to see in any environment that is undergoing constant change and evolution, an additional level of uncertainty here is a by-product of the technology itself.

There is not just *one* approach to adding support for IPv6 in your network, but *many.* And it is not just one major objective you need to address—incremental deployment of IPv6 as a second protocol into your operational network without causing undue disruption to existing services—but two, because the second challenging objective is how to fuel continued growth in your network service platform when the current supply lines of readily available IPv4 addresses are effectively exhausted.

When?

The most common question I have heard recently is: "How long do we have?"

The remaining pools of IPv4 address space continue to be drawn down. At the start of February 2011, the IANA pool was fully depleted, with the final allocation to the RIRs[1] of IPv4 addresses.

Using a model based on monthly address demands now predicts that the next 18 months or so will see the first three RIRs depleted of IPv4 addresses.

The *Asia Pacific Network Information Centre* (APNIC) was the first RIR to exhaust its available pool of IPv4 addresses in April 2011, with the *RIPE Network Coordination Centre* (RIPE NCC) predicted to follow in late 2011 and the *American Registry for Internet Numbers* (ARIN) in early 2012. The *Latin American and Caribbean Internet Addresses Registry* (LACNIC) is predicted to follow in 2014, and the *African Network Information Centre* (AFRINIC) in 2016.

The good news is that many people have been busy thinking about these intertwined objectives of extending the useful lifetime of IPv4 in the Internet and simultaneously undertaking the IPv6 transition, and there is a wealth of possible measures you can take, and a broad collection of technologies you can use. Fortunately, we are indeed spoiled with choices here!

The not-so-good news is that there is no simple single path to follow. Each individual network needs to carefully consider the transition and select an approach that matches their particular circumstances. For an industry used to playing "follow the leader" for many years, a variety of choice is not always appreciated. And, unfortunately, we are spoiled for choices here.

Let's look at each of the major transitional technologies that are currently in vogue, and examine their respective strengths and weaknesses and their intended area of applicability. We will look at these technologies first from the perspective of the end user and then from the other side, examining options for *Internet Service Providers* (ISPs).

The Dual-Stack ISP Client

If your service provider provides a dual-stack service with both IPv6 and IPv4, then your task should be relatively straightforward. If you configure your modem or router with IPv6 in addition to IPv4, you are finished, assuming of course that your local modem or router unit actually supports IPv6—an assumption that may not be valid in many of the older and, unfortunately, many of the currently available devices.

The conventional approach in this form of environment is to use *IPv6 Prefix Delegation,* where the ISP provides the client with an IPv6 prefix, usually a /48 or a /56 IPv6 address prefix, which is then passed into the client network through an *IPv6 Router Advertisement*. Local hosts should be constructed to configure their IPv6 stack automatically, and your system should be connected as a dual-protocol system.

You probably do, however, need to be aware of some caveats, of which the most important is likely to relate to the probable absence of a *Network Address Translation* (NAT)[2] function in IPv6. Currently most commercial IPv4 Internet services assign a single IP address to each client.

To allow this address to be shared within the client's network, most IPv4 "edge" devices autoconfigure themselves as NAT devices, permitting outgoing connections using the *Transmission Control Protocol* (TCP) or *User Datagram Protocol* (UDP), and allowing some *Internet Control Message Protocol* (ICMP) message types to traverse the NAT, but not much else. For many clients this NAT configuration becomes the default local security framework, because it permits outbound connections through TCP and UDP to be made, but not much else, and permits initiation of no sessions as incoming sessions. With IPv6 the local network is generally configured with an entire subnet, and instead of a NAT, this subnet is directly connected to the Internet.

The local network is then in a mixed situation of being behind a NAT in IPv4, but directly connected to the Internet using IPv6. This asymmetric configuration with respect to IPv4 and IPv6 raises some questions about the effect on the security of your local network. You need to think about adding appropriate filter rules to the gateway IPv6 configuration that performs the same level of access control to your local site that you have already set up with IPv4 and the NAT. The best advice here is to configure some filter rules for IPv6 that limit the extent of exposure of your internal network to the broader Internet to be directly comparable to the configuration you are using with IPv4.

The IPv4-Only ISP Client

Even today, when the IPv4 pools are rapidly depleting, it is really not very common to have an ISP offering dual-stack IPv4 and IPv6 services. Let's look at the more common situation, when your ISP is still offering only IPv4. As an end user, can you still set up some form of IPv6 access?

The answer is "Yes," but you must use tunnels, and the story can get somewhat ugly.

6to4 Tunnels

If you have public IPv4 addresses on your local network, you may elect to configure your local system to use the *6to4 Tunneling Protocol*.

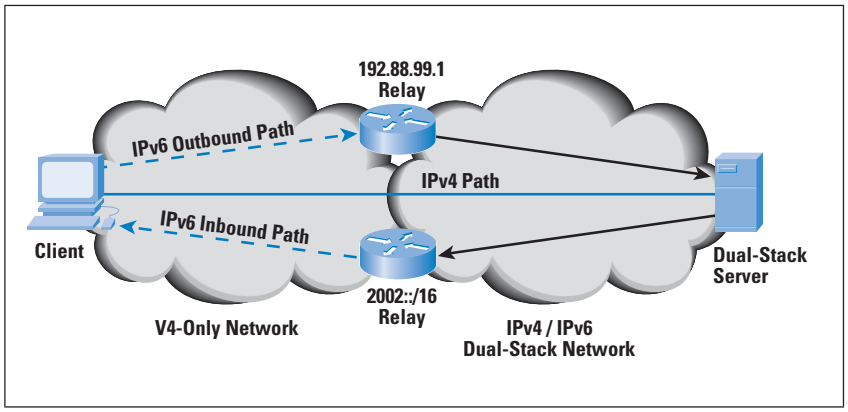
6to4 is an autotunneling protocol coupled with an addressing structure. The IPv6 address of a 6to4-reachable host begins with the IPv6 prefix `2002::/16`. The address architecture embeds a 32-bit IPv4 address of the end host into the next 32 bits. That way the IPv6 address carries the "equivalent" IPv4 address within the IPv6 address.

To send an IPv6 packet, the local host must first tunnel through the local IPv4 network. To perform this tunneling, the local host encapsulates the IPv6 packet in an outer IPv4 packet header. The IP protocol used is neither TCP nor UDP, but protocol 41, an IP protocol number reserved for tunneling IPv6 packets (RFC 2473)[3].

The IPv4 packet is addressed to an IPv4-to-IPv6 relay. To avoid manual configuration of each client, all these relays share the same *anycast* address, `192.88.99.1`. These relays strip the outer IPv4 packet header off the packet and forward the IPv6 packet into the IPv6 network. The IPv6 destination treats the packet normally, and generates a packet in response without any special processing.

The reverse path to a 6to4 host uses an IPv6-to-IPv4 relay. The IPv6 address of the 6to4 local host started with the IPv6 address prefix `2002::/16`, so the IPv6 packet that is being sent back to this host has a destination address that uses the `2002::/16` 6to4 prefix. This prefix is interpreted as an anycast relay address. A route to the IPv6 `2002::/16` prefix is advertised by IPv6-to-IPv4 relays. When a relay receives a packet destined to a `2002::/16` address, it lifts the IPv4 address from inside the IPv6 address. It then wraps the IPv6 packet in an IPv4 packet header, using as a destination address this extracted IPv4 address, and using protocol 41 as the IP protocol. The resultant IPv4 packet is then passed to the 6to4 host in the IPv4 network (Figure 1).

Figure 1: 6to4 Tunneling Architecture



If the local network has public IPv4 addresses on the local network, then individual hosts on the local network may use 6to4 directly. Of course then the local gateway needs to be configured to accept incoming IP packets that use protocol 41.

An alternative is to configure the gateway device of the local network as a 6to4 gateway, and use the IPv4 address on the ISP side of the gateway as a common 6to4 address for the local network. The gateway then advertises this synthetic 48-bit IPv6 prefix to the interior network with a conventional IPv6 Router Advertisement. The gateway can couple this advertisement with a NAT function and provide native IPv6 to interior hosts that are configured on RFC 1918[4] local IPv4 addresses.

In general, 6to4 is a relatively poor approach to provisioning IPv6, and you really should avoid it if at all possible. Indeed, your experience will probably be better overall if you continue running IPv4 and avoid accessing IPv6 with 6to4!

The major concern here is that a successful connection relies on the assistance of both an outbound and an inbound 6to4 third-party relay. On the IPv4 side a 6to4 connection relies on the presence of a usable route to a IPv4-to-IPv6 relay, and preferably one that is as close as possible to the IPv4 endpoint. On the IPv6 side a 6to4 connection relies on a usable relay advertising a route to `2002::/16`. Again, to avoid extended path overheads, this relay should be as close as possible to the IPv6 endpoint. This path asymmetry can cause connection "black holes," where one party can deliver packets to the other but not the reverse.

Also, such configurations have problems if the IPv4 host is configured with stateful filters that insist that the IPv4 source address in incoming packets match the destination address of outgoing packets, not necessarily true in a 6to4 connection.

Finally, it seems that many sites operate with firewall filters that disallow incoming packets other than TCP and UDP (and possibly some forms of ICMP). The 6to4 packets use protocol 41, and there appears to be widespread use of filter rules that block such packets.

Tunneling also adds an additional packet header to a packet, inflating the size of the packet. Such an expansion of the packet on certain path elements of the network may cause path packet size problems, increasing the risk of encountering Path *Maximum Transmission Unit* (MTU) "black holes" due to the increase of the packet size by 20 bytes when the IPv4 packet header is attached to the packet.

Teredo Tunnels

If the local network is behind an IPv4 NAT and the NAT gateway does not support 6to4, then all is not lost, because another form of tunneling could possibly be an answer. *Teredo* is described in RFC 4380[5].

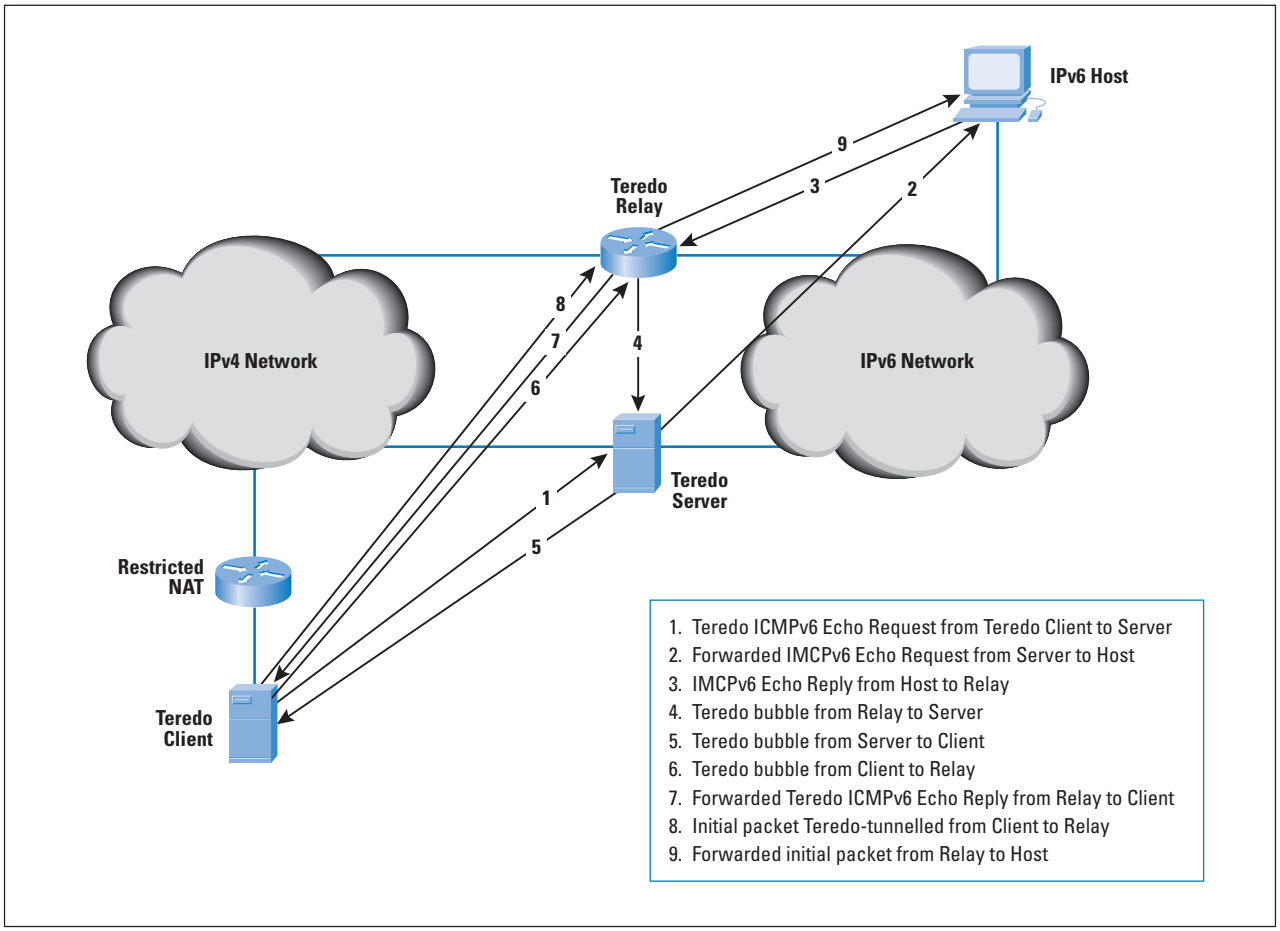
Teredo, like 6to4, is an autotunneling protocol coupled with an addressing structure. Like 6to4, Teredo uses its own address prefix, and all Teredo addresses share a common IPv6 /32 address prefix, namely `2001:0000::/32`. The next 32 bits are the IPv4 address of the Teredo server. The IPv6 interface identifier field is used to support NAT traversal, and it is encoded with the triplet of a field describing the NAT type, the view of the relay of the UDP port number used to reach the client (the external UDP port number used by the NAT binding for the client), and the view of the relay of the IPv4 address used to reach the client (the external IPv4 address used by the NAT binding for the client).

Teredo uses what has become a relatively conventional approach to NAT traversal, using a simplified version of the *Session Traversal Utilities for NAT* (STUN)[6] active probing approach to determine the type of NAT; it uses concepts of "clients," "servers," and "relays."

A Teredo *client* is a dual-stack host that is located in the IPv4 world, assumed to be located behind a NAT. A Teredo *server* is an address and reachability broker that is located in the public IPv4 Internet, and a Teredo *relay* is a Teredo tunnel endpoint that connects Teredo clients to the IPv6 network. The tunneling protocol used by Teredo is not the simple IPv6-in-IPv4 protocol 41 used by 6to4. NAT devices are sensitive to the transport protocol and generally pass only TCP and UDP transport protocols. In the Teredo case the tunneling is UDP, so all IPv6 Teredo packets are composed of an IPv4 packet header and a UDP transport header, followed by the IPv6 packet as the UDP payload. Teredo uses a combination of ICMPv6[7] message exchanges to set up a connection and tunneled packets encapsulated using an outer IPv4 header and a UDP header, and it contains the IPv6 packet as a UDP payload.

It should be noted that this reliance on ICMPv6 to complete an initial protocol exchange and confirm that the appropriate NAT bindings have been set up is not a conventional feature of IPv4 or even IPv6, and IPv6 firewalls that routinely discard ICMP messages will disrupt communications with Teredo clients.

Figure 2: Teredo Tunneling



The exact nature of the packet exchange in setting up a Teredo connection depends on the nature of the NAT device that sits in front of the Teredo client. Figure 2 shows an example packet exchange that Teredo uses when the client is behind a Restricted NAT.

Teredo represents a different set of design trade-offs as compared to 6to4. In its desire to be useful in an environment that includes NAT functions in the IPv4 path, Teredo is a per-host connectivity approach, as compared to the 6to4 approach, which can support both individual hosts and entire end sites within the same technology. Also, Teredo is a host-centric multiparty rendezvous application, and Teredo clients require the existence of dual-stack Teredo servers and relays that exist in both the public IPv4 and IPv6 networks. Teredo is more of a connectivity tool than a service solution, and one that is prone to many forms of operational failure.

On the other hand, if you are an isolated IPv6 host behind an IPv4 NAT and you want to access the IPv6 network, then 6to4 is not an option, and you either have to set up static tunnels across the NAT to make it all work or turn on Teredo in your dual-stack host; if everything goes according to theory, you should be able to establish IPv6 connectivity. It is highly likely that the IPv6 Teredo connection will fail in strange ways, and, like 6to4, this is a technology best avoided!

Tunnel Brokers

In contrast to these autotunnel approaches, the simplest form of tunneling IPv6 packets over an IPv4 network is the manually configured IPv6-in-IPv4 tunnel.

Here an IPv6 packet is simply prefixed by a 20-octet IPv4 packet header. In the outer IPv4 packet header, the source address is the IPv4 address of the tunnel ingress, the destination address is the IPv4 address of the tunnel egress, and the IP protocol field uses value 41, indicating that the payload is an IPv6 packet. The packet is passed across the IPv4 network from tunnel ingress to egress using conventional IPv4 packet forwarding, and at the egress point the IPv4 IP packet header is removed and the inner IPv6 packet is routed in an IPv6 network as before. From the IPv6 perspective the transit across the IPv4 network is a single logical hop.

Alternatively, like *Virtual Private Network* (VPN) tunnels, the tunnel can be configured using UDP or TCP, and with some care, the tunnel can be configured through NAT functions in the same way as VPN tunnels can be configured through NAT functions.

The advantage of this approach is that the need to manually configure the tunnel endpoints ensures that the tunnel relay function is not provided, intentionally or unintentionally, by third parties through some well-intentioned, but ultimately random, act of goodwill. The need to perform a manual configuration also reduces the chances that the tunnel will be broken through local firewall filters.

Of course the need to perform a manual configuration does not lend itself to a "plug-and-play" environment, nor is this approach a viable one for a larger mass market of consumer devices and services.

Client Conclusions

None of these approaches to offer IPv6 connectivity to end hosts behind an IPv4-only service provider offers the same level of robustness and performance as native IPv4 services. All of these approaches require a significant degree of local expertise to set up and maintain, and they often require a solid understanding of other aspects of the local environment, such as firewall and filter conditions and Path MTU behavior to maintain. With the exception of the tunnel broker approach, they also require third-party assistance to support the connection, further adding to the set of potential performance and reliability concerns.

It appears that the most robust and reliable way to provision IPv6 to end hosts is for the service provider to provision IPv6 as an integral part of its service offering, and offer clients a dual-stack service in both IPv4 and IPv6.

IPv6 for Internet Service Providers

Although the "self-help" autotunneling approaches for clients outlined earlier in this article are a possible answer, their utility is appropriately restricted to a very small number of end clients who have the necessary technical expertise and who are willing to debug some rather strange resultant potential problems relating to asymmetric paths, third-party relays, potential MTU mismatches, and interactions with filters. This approach is not a reasonable one for the larger Internet.

From the perspective of the mass market for Internet Services, we cannot assume that clients have the motivation, expertise, and means to bypass their ISP and set up IPv6 access on their own, either through autotunneling or manually configured tunnels. The inference from this observation is that for as long as the mass-market ISPs do not commit to IPv6 services, and for as long as they continue to stall in deploying services supporting dual access for their clients, the entire IPv6 transition story remains effectively stalled.

How can ISPs support IPv6 access for their clients?

The Dual-Stack Service Network

Perhaps it is obvious, but the most direct response here is for the ISP to operate a *Dual-Stack Network*.

And the most direct way to achieve this operation is for the ISP's infrastructure to also support IPv6 wherever there is IPv4, so that the delivery of services to the ISP's clients in IPv6 faithfully replicates the service offered in IPv4.

This solution implies that the network needs to support IPv6 in the ISP's routing infrastructure, in the network data plane, in the load-management systems, in the operational support infrastructure, in access and accounting, and in peering and in transit. In short, wherever there is IPv4 there needs to be IPv6.

The infrastructure elements that require dual-stack service at the next level include the routing and switching elements, including the internal and external routing protocols. The task includes negotiating peering and transit services in IPv6 to complement those in IPv4. Network infrastructure also includes VPN support and other forms of tunnels, as well as data center front-end units, including load balancers, filters and firewalls, and various virtualized forms of service provision. The task also includes integration of IPv6 in the network management subsystem and the related network measurement and reporting system. Even a comprehensive audit of the supported *Management Information Bases* (MIBs) in the active elements of the network to ensure that the relevant IPv6 MIBs are supported is an essential task. A similar task is associated with equipping the server infrastructure with IPv6 support, and at the higher levels of the protocol stack are the various applications, including web services, mail, *Domain Name System* (DNS), authentication and accounting, *Voice over IP* (VoIP) servers, Load Balancers, Cloud Servers, and similar applications.

And those are just the common elements of most ISPs' infrastructures. Every ISP also has more specialized elements in its service portfolio, and each one of these elements also requires a comprehensive audit to ensure that there is an IPv6 solution for each of these elements that leads to a comprehensive dual-stack outcome.

As obvious as this approach might appear, it has two significant problems. First, it requires a comprehensive overhaul of every element in the ISP's service network. Even for small-scale ISPs this overhaul is not trivial, and for larger service provider platforms it is an exercise that may take months if not years and make considerable inroads into the operating budgets of the ISPs. Secondly, it still does not account for the inevitable fact that in the coming months the current supply lines of IPv4 addresses will end and any continued expansion of the service platform will require some different approaches to the way in which IPv4 addresses are deployed in the service platform.

Although the approach of simply provisioning IPv6 alongside IPv4 in a simple dual-protocol service infrastructure may appear to be the most obvious response to the need to transition to IPv6, it may not necessarily be the most appropriate response for many ISPs to the dual factors of IPv6 transition and IPv4 address exhaustion.

Are there alternative approaches for ISPs? Of course.

Hybrid Approaches

Saying that an ISP must deploy IPv6 across all of its infrastructure and actually doing it are often quite different. The cost of converting all parts of an ISP's operation to run in dual-stack mode can be quite high, and the benefit of running every aspect of an ISP's service offering in dual-stack mode is dubious at best.

Are there middle positions here? Is it possible for an ISP to deliver robust IPv6 services to clients while still operating an IPv4-only internal network? One way to look at an ISP's network is as a transit conduit (Figure 3).

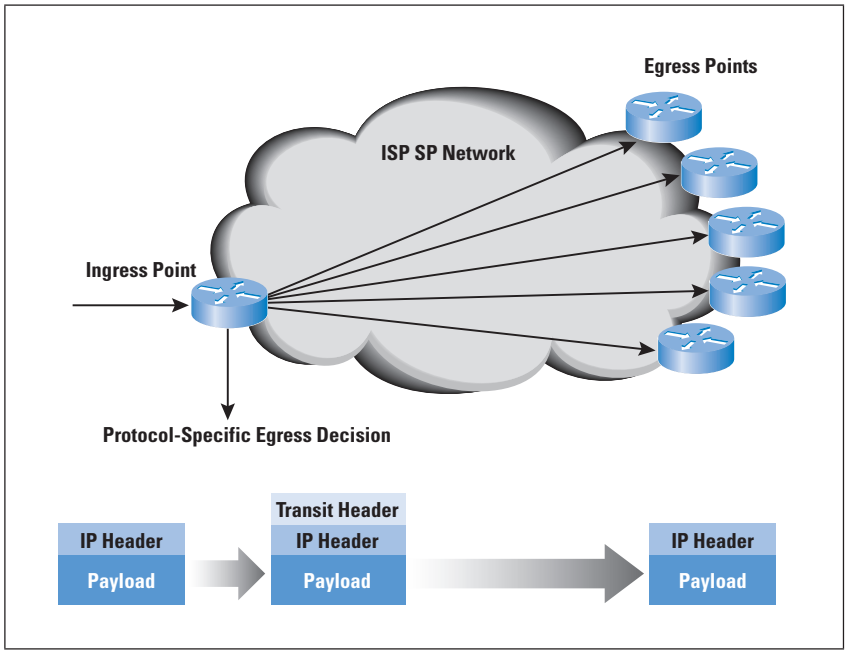


Figure 3: Generic ISP Packet Transit Architecture

The ISP needs to be able to accept packets from an external interface, determine the appropriate egress point for the packet within the context of the local network, and then ensure that the packet is passed out this egress interface. The internal network need not operate in the same protocol context as the protocol of the packets the network is handling. Viewed at a level of the minimal essentials, the network needs to be able to have some protocol-specific capability at its ingress points in order to determine the appropriate egress point of each incoming packet, and thereafter during the transit of the service provider's network, the minimum necessary association to maintain the identity of this preselected egress point with the packet. Now if the network uniformly supports the same protocol as the packet, then the same egress decision can be made at each forwarding point within the network.

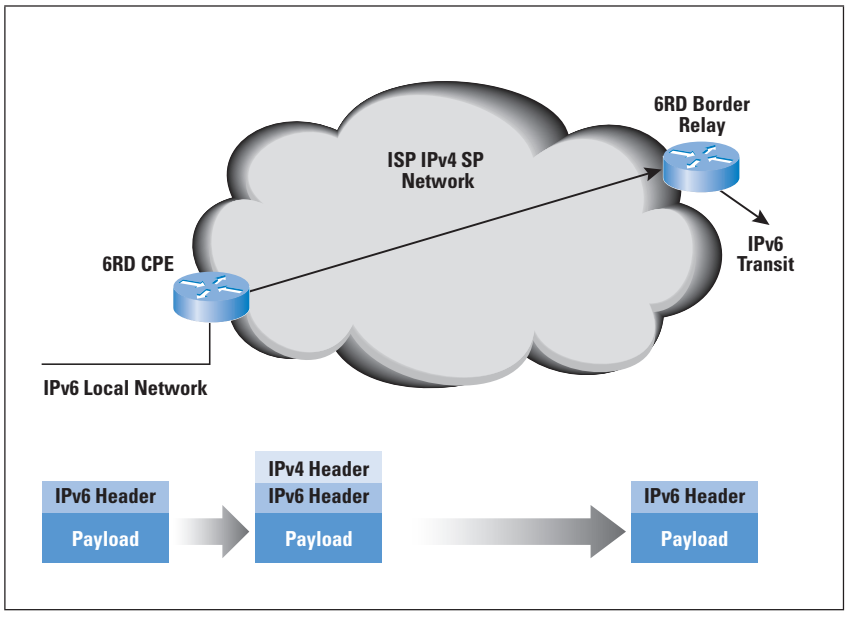
Alternatively, the packet can be encapsulated with an outer wrapper that identifies the egress point using the same protocol context as that used by the service provider's internal switching elements, and the packet can be passed through the service provider's transit network using only this temporary wrapper to determine the sequence of forwarding decisions. *Multiprotocol Label Switching* (MPLS) networks are an excellent example of this form of approach, as are other forms of IP-in-IP encapsulation. The advantage of this approach is that the internal infrastructure of the service provider network need not be altered to support additional carriage protocols: the changes to specifically support IPv6 are required only at the network ingress elements, and a basic encapsulation stripping function is used at all egress points.

With this information in mind, let's look at some of these hybrid approaches to supporting IPv6 in a service provider network.

6RD

6RD, described in RFC 5969[8], is an interesting refinement of the 6to4 approach. It shares the same basic encapsulation protocol and the same address structure of embedding of the IPv4 tunnel endpoint into the IPv6 address. However, it has removed the concept of third-party relays and the use of the common `2002::/16` IPv6 prefix, and instead uses the provider's IPv6 prefix. The effect of these changes is to limit the scope of the tunneling mechanism to that of tunneling across the network infrastructure of a single provider, and the intended function is to tunnel from the *Customer Premises Equipment* (CPE) to IPv6 *Border Relays* operated by the customer's ISP (Figure 4).

Figure 4: 6RD Tunneling



If 6to4 is not recommended for use because of high failure rates of connections and suboptimal performance, then why would 6RD be any better?

The most compelling reason to believe that 6RD will perform more reliably than 6to4 is that 6RD removes the wild-card third-party relay element from the picture. For outbound traffic the CPE provides the tunnel encapsulation, which is, hopefully, under the ISP's operational control. The IPv6-in-IPv4 tunnel is directed to the ISP's own 6RD Border Relay rather than the 6to4 relay anycast address. Because this process is also under the ISP's direct operational control, it eliminates the outbound third-party relay function. For the reverse path, the use of the provider's own IPv6 prefix in 6RD, instead of the generic `2002::/16` prefix, ensures that the inbound packets are sent through IPv6 directly to the ISP, and the IPv6-in-IPv4 tunnel is again limited to a hop across the ISP's own internal infrastructure.

As long as the ISP effectively manages all CPE devices, and as long as the CPE itself is capable of supporting the configuration of additional functional modules that can deliver unicast IPv6 to the client and 6RD tunnels inward to the ISP, then 6RD is a viable option for the ISP. At the cost of upgrading the CPE set to include 6RD support, and the cost of deployment of 6RD Border Relays that terminate these CPE tunnels, together with IPv6 transit from these Border Relays, the ISP is in a position to provide dual-stack support to its client base from an internal network platform that remains an IPv4 service platform, thereby deferring the process of conversion of its entire network infrastructure base to support IPv6.

For ISPs seeking to defray the internal infrastructure IPv6 conversion costs over a number of years, or for ISPs seeking an incremental path to IPv6 support that allows the existing infrastructure to remain in place temporarily, 6RD can be an interesting and cost-effective alternative to a comprehensive dual-stack deployment, as long as the ISP has some mechanism to load the CPE with IPv6 support and 6RD relay functions.

MPLS and 6PE

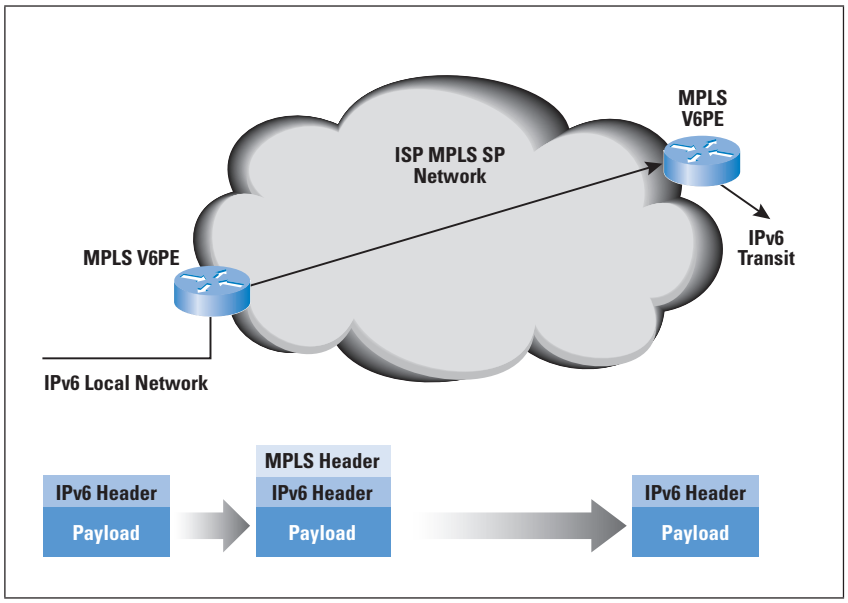
The 6RD approach has many similarities to MPLS, in that an additional header is added to incoming packets at the network boundary, and the encapsulation effectively directs the packet to the appropriate network egress point (as identified by ingress), where the encapsulation is stripped and the original packet is passed out.

Rather than using an IPv4 header to direct a packet from ingress to egress, if the network is already using MPLS, why not simply support IPv6 on an existing MPLS network as a PE-to-PE MPLS path set and bypass the IPv4 step?

Why not, indeed, and RFC 4659[9] describes how this bypass can be achieved.

If you are running an MPLS network, then the role of the interior routing protocol and label distribution function is to maintain viable paths between all network ingress and egress points. The protocol-specific function in such networks is not the interior network topology management function, but the maintenance of the mapping of egress to protocol-specific destination addresses (Figure 5).

Figure 5: MPLS and 6PE



As with 6RD, if the local problem is some form of prohibitive barrier to the immediate deployment of IPv6 in a dual-stack configuration across the network infrastructure, then this approach allows an IPv4 MPLS network to set up paths across the network IPv4 MPLS infrastructure from provider edge to provider edge. These paths may be used to tunnel IPv6 packets across the network by associating the IPv6 destination address of the incoming packet with the IPv4 address of the egress router, using the *interior Border Gateway Protocol* (iBGP) *Next-Hop* address, for example.

The incremental changes to support IPv6 are constrained to adding IPv6 to the service provider's iBGP routing infrastructure, and to the provider-edge devices in the MPLS network, while all other parts of the service provider's service platform can continue to operate as an MPLS IPv4 network for now.

IPv4 Address Compression

It is not just the challenge of adding a new protocol to the existing IPv4 network infrastructure that confronts ISPs. The entire reason for this activity is the prospect of exhaustion of supply of IPv4 addresses. When this prospect was first aired, in 1990, it was assumed that the Internet would be supported by industry players that acted rationally in terms of common interests.

One of the more critical assumptions made in the development of transitional tools was that transition activity would be undertaken well in advance of IPv4 address exhaustion. Competitive interest would see each actor making the necessary investments in new technologies to mitigate the risks of attempting to operate a network in an environment of acute general scarcity of addresses. As much fun as the debate as to whom the "last" IPv4 address should be given might be, it was assumed that this event was, in fact, never going to happen. The assumption was that industry actors would anticipate this situation and take the necessary steps to avoid it. The transition to IPv6 would be effectively complete well before the stocks of IPv4 addresses had been exhausted, and IPv4 addresses would be an historical artefact well before we needed to use the last one!

Obviously, this scenario has not happened.

This industry is going to exhaust the available supplies of IPv4 addresses well before the transition to IPv6 is complete—and in some cases well before the transition process has even commenced! This situation creates an additional challenge for ISPs and the Internet, and raises a further question as well. The challenge is to fold into this dual-stack transition the additional factor of having to work with fewer and fewer IPv4 addresses as the transition process continues. This situation implies that the necessary steps that the ISP must take include ones that increase the intensity of use of each IPv4 address, and wherever possible substitute a private-use IPv4 address for public IPv4 addresses.

The question that this scenario raises is one of guessing how long this hybrid model of an Internet where a significant proportion of network services and network clients remains entrenched in an IPv4-only world will persist. For as long as such IPv4-only network domains persist, and for as long as these IPv4-only network domains encompass significant service and customer populations, all the other parts of the Internet are forced to maintain residual IPv4 capability and cannot transition their customers and services to an IPv6-only environment. Students of economic game theory may see some rich areas of study in this developing situation.

More practically, for an ISP the question becomes one of attempting to understand how long this hybrid period of attempting to operate a dual-stack network with continuing postexhaustion demand for further IPv4 addresses will last. Will an after-market for the redistribution of addresses emerge? How will the increasing scarcity pressure affect pricing in such a market? How long will demand persist for IPv4 addresses in the face of escalating prices? Will the industry turn to IPv6 in a rapid surge in response to cost escalation for additional IPv4 addresses, or will a dual-stack transition lumber on for many years? In such a large, diverse, heterogeneous environment of today's Internet, the one constant factor is that the immediate future of the Internet is clouded with extremely high levels of uncertainty.

The cumulative effect of the individual decisions made by service providers, enterprises, carriers, vendors, policy makers, and consumers has created a somewhat chaotic environment that adds a significant level of uncertainty and associated investment risk into the current planning process for ISPs.

Carrier-Grade NATs

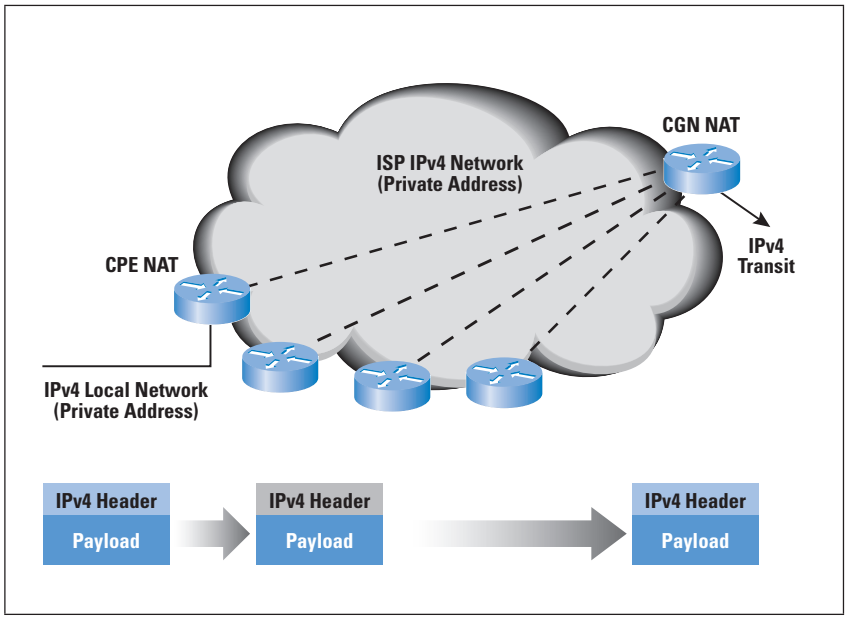
I have often heard it said that address scarcity in IPv4 is nothing new, and it first occurred when the first NAT device that supported port mapping was deployed. At this point the concept of *address sharing* was introduced to the Internet, and, from the perspective of the NAT industry, we have not looked back since.

In today's world NATs are extremely commonplace. Most clients are provisioned with a single address from their ISP, which they then share across their local network using a NAT. Whether it is well advised or not, NATs typically form part of a client's network security framework, and they often are an integral part of a customer's multihoming configuration if the client uses multiple providers.

But in this model of NATs as the CPE, the ISP uses one IPv4 address for each client. If the ISP wants to achieve greater levels of address compression, then it is necessary to share a single IPv4 address across multiple customers.

The most direct way to achieve this scenario is for ISPs to operate their own NAT, variously termed a *Carrier-Grade NAT* (CGN) or a *Large-Scale NAT* (LSN), or *NAT444*. This approach is the simplest, and, in essence, is a case of "more of the same" (Figure 6).

Figure 6: Carrier-Grade NATs



The Carrier-Grade NAT allows a single public address to be shared across multiple clients, who, in turn, further share this address across the end systems in their local networks.

From behind the CPE in the client edge network not much has changed with the addition of the CGN in terms of application behavior. It still requires an outbound packet to trigger a binding that would allow a return packet through to the internal destination, so nothing has changed there. Other aspects of NAT behavior, notably the NAT binding lifetime and the form of NAT "cone behavior" for UDP, take on the more restrictive of the two NAT functions in sequence. The binding times are potentially problematic in that the two NATs are not synchronized in terms of binding behavior. If the CGN has a shorter binding time, it is possible for the CGN to misdirect packets and cause application-level problems. However, this situation is not overly different from a single-level NAT environment where aggressively short NAT binding times also run the risk of causing application-level problems when the NAT drops the binding for an active session that has been quiet for an extended period of time.

However, one major assumption is broken in this structure, namely that an IP address is associated with a single customer. In the CGN model a single public IP address may be used simultaneously by many customers at once, albeit on different port numbers. This scenario has obvious implications in terms of some current practices in filters, firewalls, "black" and "white" lists, and some forms of application-level security and credentials where the application makes an inference about the identity and associated level of trust in the remote party based on the remote party's IP address.

This approach is not without its potential operational problems as well. For the service provider, service resiliency becomes a critical concern in so far as moving traffic from one NAT-connected external service to another will cause all the current sessions to be dropped. Another concern is one of resource management in the face of potentially hostile applications. For example, an end host infected with a virus may generate a large amount of probe packets to a large range of addresses. In the case of a single edge NAT, the large volumes of bindings generated by this behavior become a local resource-management problem because the customer's network is the only affected site. In the case where a CGN is deployed, the same behavior will consume port-binding space on the CGN and, potentially, can starve the CGN of external address port bindings. If this problem is seen to be significant, the CGN would need to have some form of external address rationing per internal client in order to ensure that the entire external address pool is not consumed by a single errant customer application.

The other concern here is one of *scalability*. Whereas the most effective use of the CGN in terms of efficiency of usage of external addresses occurs when the greatest numbers of internal edge NATed clients are connected, there are some real limitations in terms of NAT performance and address availability when a service provider wants to apply this approach to networks where the customer population is in the millions or larger. In this case the service provider must use an IPv4 private address pool to number every client. But if network 10 is already used by each customer as its "internal" network, then what address pool can be used for the service provider's private address space? One of the few answers that come to mind is to deliberately partition the network into numerous discrete networks, each of which can be privately numbered from `172.16.0.0/12`, allowing for some 600,000 or so customers per network partition, and then use a transit network to "glue" together the partitioned elements.

The advantage of the CGN approach is that nothing changes for the customer. There is no need for any customers to upgrade their NAT equipment or change it in any way, and for many service providers this motivation is probably sufficient to choose this path. The disadvantages of this approach lie in the scaling properties when looking at very large deployments, and the concerns of application-level translation, where the NAT attempts to be "helpful" by performing *Deep Packet Inspection* and rewriting what it thinks are IP addresses found in packet payloads. Having one NAT do this process is bad enough, but loading them up in sequence is a recipe for trouble.

Are there alternatives?

The Address-plus-Port Approach

One NAT in the path is certainly worse than none from the perspective of application agility and functions. And two NAT functions do not make it any better! Inevitably, that second NAT device adds some additional levels of complexity and fragility into the process.

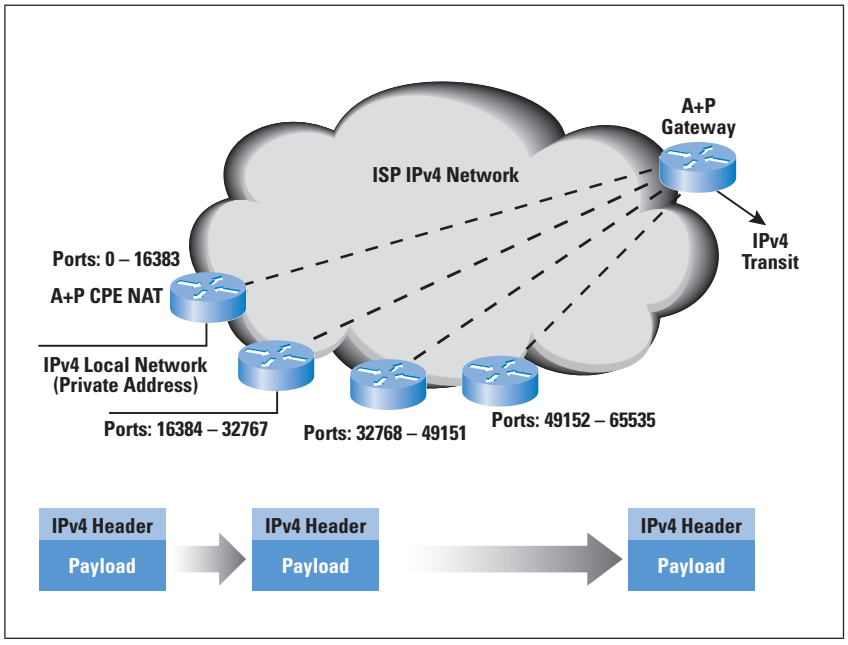
The question is, can these two NAT functions be collapsed back into a single NAT, yet still allow sharing of public IPv4 addresses across multiple end clients? CPE NAT devices currently map connections into the 16-bit *port* field of the single external address. If the CPE NAT could be coerced into performing this mapping into, say, 15 bits of the port field, then the external address could be shared between two edge CPEs, with the leading bit of the port field denoting which CPE. Obviously, moving the bit marker further across the port field will allow more CPE devices to share the one address, but it will reduce the number of available ports for each CPE in the process.

The theory is again quite simple. The CPE NAT is dynamically configured with an external address, as happens today, and a port range, which is the additional constraint. The CPE NAT performs the same function as before, but it is now limited in terms of the range of external port values it can use in its NAT bindings to those that lie within the provided port range. Other CPE devices are concurrently using the same external IP address, but with a different port range.

For outgoing packets this scenario implies only a minor change to the network architecture, in that the RADIUS exchange to configure the CPE now must also provide a port range to the CPE device. The CPE is then constrained such that as it maps private addresses and TCP or UDP port values to the external address and port values, the mapped port value must fall within the configured range.

The handling of incoming packets is more challenging. Here the service provider must forward the packet based not only on the destination IP address, but also on the port value in the TCP or UDP header, because there are now multiple CPE egress points that share the same IP address. A convenient way to perform forwarding is to take the Dual-Stack Lite approach and use an IPv4-in-IPv6 tunnel between the CPE and the external address-plus-port (A+P) gateway. This address-plus-port gateway needs to be able to associate each address and port range with the IPv6 address of a CPE (which it can learn dynamically as it decapsulates outgoing packets that are similarly tunneled from the CPE to the address-plus-port gateway). Incoming packets are encapsulated in IPv6 using the IPv6 destination address that it has learned previously. In this manner the NAT function is performed just once, at the edge, much as it is today, and the interior device is a more conventional form of tunnel server (Figure 7).

Figure 7: Address-plus-Port-Approach



This approach relies on every CPE device being able to operate using a restricted port range, to perform IPv4-in-IPv6 tunnel ingress and egress functions, and act as an IPv6 provisioned endpoint for the service provider network. This set of constraints is perhaps unrealistic for many service provider networks. Further modifications to this model propose the use of an accompanying CGN operated by the service provider to handle those CPE devices that cannot support this address-plus-port function.

This approach has some positive aspects. Pushing the NAT function back to the network edge has some considerable advantage over the approach of moving the NAT to the interior of the network. The packet rates are lower at the edge, allowing for commodity computing to process the NAT functions across the offered packet load without undue stress. The ability to control the NAT behavior with the *Internet Gateway Device* protocol as part of the *Universal Plug and Play* (uPnP) framework will still function in an environment of restricted port ranges. Aside from the initial provisioning process to equip the CPE NAT with a port range, the CPE and the edge environment are largely the same as that of today's CPE NAT model.

That is not to say that this approach is without its negative aspects, and it is unclear as to whether the perceived benefits of a "local" NAT function outweigh the problems in this particular model of address sharing. The concept of port "rationing" is a very suboptimal means of address sharing, given that when a CPE is assigned a port range, those port addresses are unusable by any other CPE. The prudent service provider would assign to each CPE a port address pool equal to some estimate of peak demand, so that, for example, each CPE would be assigned some 1024 ports, allowing a single external IP address to be shared across only some 60 such CPE clients. The Carrier-Grade NAT and Dual-Stack Lite approaches do not attempt this form of rationed allocation, allowing the port address pool to be treated as a common resource, with far higher levels of usage efficiency. The leverage obtained in terms of efficiently using these additional 16 bits of address space is reduced by the imposition of a fixed boundary between customer and service provider use. The central NAT model effectively pools the port address range and would result in more efficient sharing of this common pool across a larger client base.

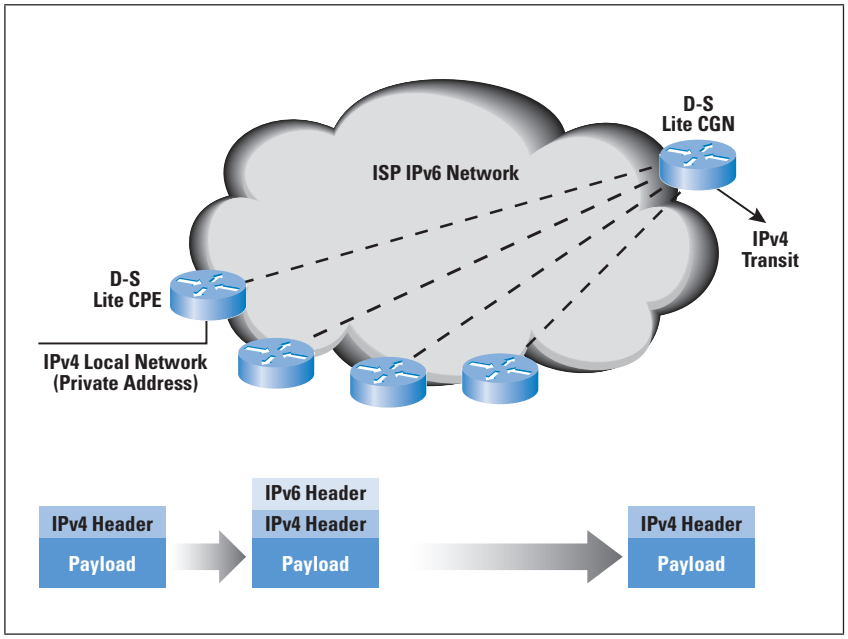
The other consideration here is that this approach means a higher overhead for the service provider, in that the service provider would have to support both "conventional" CPE equipment and address-plus-port equipment. In other words, the service provider will have to deploy a CGN and support customer CPE using a two-level NAT environment in addition to operating the address-plus-port infrastructure. Unless customers would be willing to pay a significant price premium for such address-plus-port service, it is unlikely that this option would be attractive for the service provider as an additional cost above the CGN cost.

Dual-Stack Lite

The concept behind the *Dual-Stack Lite* approach is that the service provider's network infrastructure will need to support IPv6 running in native mode in any case, so is there a way in which the service provider can continue to support IPv4 customers without running IPv4 internally?

Here the customer NAT is effectively replaced by a tunnel ingress-egress function in the Dual-Stack Lite home gateway. Outgoing IPv4 packets are not translated, but are encapsulated in an IPv6 packet header, which contains a source address of the carrier side of the home gateway unit, and a destination address of the ISP's gateway unit. From the service provider's perspective, each customer is no longer uniquely addressed with an IPv4 address, but instead is addressed with a unique IPv6 address, and provided with the IPv6 address of the provider's combined IPv6 tunnel egress point and IPv4 NAT unit (Figure 8).

Figure 8: Dual-Stack Lite



The service provider's Dual-Stack Lite gateway unit will perform the IPv6 tunnel termination and a NAT translation using an extended local binding table. The NAT "interior" address is now a 4-tuple of the IPv4 source address, protocol ID, and port, plus the IPv6 address of the home gateway unit, while the external address remains the triplet of the public IPv4 address, protocol ID, and port. In this way the NAT binding table contains a mapping between interior "addresses" that consist of IPv4 address and port plus a tunnel identifier, and public IPv4 exterior addresses. This way the NAT can handle a multitude of net 10 addresses, because they can be distinguished by different tunnel identifiers.

The resultant output packet following the stripping of the IPv6 encapsulation and the application of the NAT function is an IPv4 packet with public source and destination addresses. Incoming IPv4 packets are similarly transformed, where the IPv4 packet header is used to perform a lookup in the Dual-Stack Lite gateway unit, and the resultant 4-tuple is used to create the NAT-translated IPv4 packet header plus the destination address of the IPv6 encapsulation header.

The advantage of this approach is that there now needs to be only a single NAT in the end-to-end path, because the functions of the customer NAT are now subsumed by the carrier NAT. This scenario has some advantages in terms of those messy "value-added" NAT functions that attempt to perform deep packet inspection and rewrite IP addresses found in data payloads. There is also no need to provide each customer with a unique IPv4 address, public or private, so the scaling limitations of the dual-NAT approach are also eliminated. The disadvantages of this approach lie in the need to use a different CPE device—or at least one that is reprogrammed. The device now requires an external IPv6 interface and at the minimum an IPv4/IPv6 tunnel gateway function. The device can also include a NAT if so desired, but it is not required in terms of the basic Dual-Stack Lite architecture.

This approach pushes the translation into the interior of the network, where the greatest benefit can be derived from port multiplexing, but it also creates a critical hotspot for the service itself. If the Dual-Stack Lite NAT fails in any way, the entire customer base is disrupted. It seems somewhat counterintuitive to create a resilient end-to-end network with stateless switching environments and then place a critical stateful unit right in the middle!

Protocol Translation

So far we have looked at two general forms of approach to hybrid networks that are intended to support both IPv6 transition and greater levels of address usage in IPv4, namely address mapping and tunneling. A third approach lies in the area of protocol translation.

RFC 2765[10] contains the details of a relatively simple protocol-translation mechanism. The approach relies on the basic observation that IPv6 did not make any radical changes to the basic IP architecture of IPv4, and that it was therefore possible to define a stateless mapping algorithm that could translate between certain IPv4 and IPv6 packets. Of course the one major problem here is that there are far more addresses in IPv6 than in IPv4, so the approach used was to map IPv4 addresses into the trailing 32 bits of the IPv6 address prefix `::FFFF:0:0/96`. The approach assumed that to the IPv6-only end host the entire IPv4 network was visible in this mapped IPv6 prefix, and that when the IPv6-only end host wished to communicate with a remote host who was addressed using this IPv4-mapped prefix it would use a source address also drawn from the same IPv4-mapped prefix. In other words, it assumed that all IPv6-only hosts were also assigned a unique IPv4 address.

The *NAT-Protocol Translation* (NAT-PT) approach attempted to relax this constraint, allowing IPv6-only hosts to use a dynamic mapping to a public IPv4 address through the NAT-PT function, in the same way as NAT functions work in an all-IPv4 domain (Figure 9). The proposed approach assumed that the local host was located behind a modified DNS environment where the IPv4 "A" record of an IPv4-only remote service is translated by the DNS gateway into a local IPv6 address where the initial 96 bits of the IPv6 address identify the internal address of the NAT-PT gateway and the trailing 32 bits are the IPv4 address of the remote service. When the local host then uses this address as an IPv6 destination address, the packet is directed by the local routing environment to the NAT-PT device. This device can construct an "equivalent" IPv4 packet by using the local IPv4 address as the source address and the last 32 bits of the IPv6 address as the destination address, and bind the IPv6 source port to a free local port value. These sets of transforms can be locally stored as an active NAT binding. Return IPv4 packets can be mapped back into their "equivalent" IPv6 form by using the values in the binding to perform a reverse set of transforms on the IP address and port fields of the packet.

This approach was published as RFC 2766[11] in February 2000. Some 7 years later in July 2007, the IETF published RFC 4966[12], deprecating NAT-PT to "historic," with an associated list of applications that would not operate correctly through such a device. This negative judgement of NAT-PT seems rather curious to me, given that conventional CPE NAT functions in IPv4 appear to share most, if not all, of the same shortfalls that are listed in RFC 4966. Given the extensive set of compromises that are required in the environment that is partially crippled by IPv4 address exhaustion, it seems rather contradictory to insist upon extremely high levels of functions and robustness from these hybrid translation approaches.

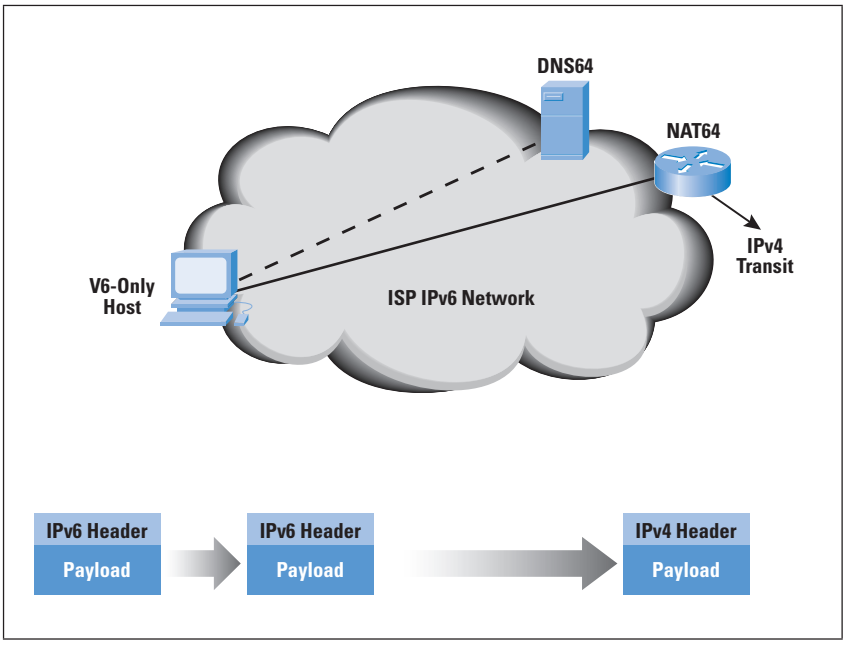


Figure 9: NAT Protocol Translation – NAT64

Not unsurprisingly, NAT-PT is undergoing a revival, this time under the name "NAT64." Not much has changed from the basic approach outlined in NAT-PT. The IPv6-only client performs a DNS lookup through a modified DNS server that is configured with DNS64. If the queried name contains only an IPv4 address, the DNS64 server synthesises an IPv6 response by merging the prefix address of the NAT64 gateway with the IPv4 address. When the client uses this address, the IPv6 packet is directed to the NAT64 gateway, and the same transform as described previously for NAT-PT takes place.

This setup is similar to the CGN model, in so far as the service provider operates a common NAT that shares an IPv4 address pool across a set of end clients.

ISP Conclusions

There really is no single clear path forward from this point. Different ISPs will see some advantages in pursuing different approaches to this dual problem of introducing IPv6 into their service portfolio and at the same time introducing additional measures that allow more efficient use of IPv4 addresses.

However, one common theme is becoming clear. So far ISPs have been able to "externalize" many of these problems by pushing much of the complexity and fragility of NAT functions out to the customer and loading up the CPE with these functions. This approach of externalizing much of the complexity of address compression in NAT functions over to the customer's network cannot be sustained with the IPv6 transition, and no matter which approach is used, whether it is a CGN, NAT64, Dual-Stack Lite, 6RD, or MPLS with 6PE, the ISP now has to actively participate in the delivery of IPv6 and in increasing the efficiency of the use of IPv4.

So for the ISP it is time to start making some technical choices as to how to address the combination of these two rather unique challenges of transition and exhaustion.

References

[1] Daniel Karrenberg, Gerard Ross, Paul Wilson, and Leslie Nobile, "Development of the Regional Internet Registry System," *The Internet Protocol Journal,* Volume 4, No. 4, December 2001.

[2] Geoff Huston, "Anatomy: A Look inside Network Address Translators," *The Internet Protocol Journal,* Volume 7, No. 3, September 2004.

[3] Alex Conta and Stephen Deering, "Generic Packet Tunneling in IPv6 Specification," RFC 2473, December 1998.

[4] Yakov Rekhter, Bob Moskowitz, Daniel Karrenberg, Geert Jan de Groot, and Eliot Lear, "Address Allocation for Private Internets," RFC 1918, February 1996.

[5] Christian Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)," RFC 4380, February 2006.

[6] Jonathan Rosenberg, Rohan Mahy, Philip Matthews, and Dan Wing, "Session Traversal Utilities for NAT (STUN)," RFC 5389, October 2008.

[7] Alex Conta, Stephen Deering, and Mukesh Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," RFC 4443, March 2006.

[8] Mark Townsley and Ole Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) – Protocol Specification," RFC 5969, August 2010.

[9] Jeremy De Clercq, Dirk Ooms, Marco Carugi, and Francois Le Faucheur, "BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN," RFC 4659, September 2006.

[10] Erik Nordmark, "Stateless IP/ICMP Translation Algorithm (SIIT)," RFC 2765, February 2000.

[11] George Tsirtsis and Pyda Srisuresh, "Network Address Translation – Protocol Translation (NAT-PT)," RFC 2766, February 2000.

[12] Cedric Aoun and Elwyn Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status," RFC 4966, July 2007.

Further Reading

The IETF has been working on the issues related to the transition to IPv6 for the past 18 years, and in the intervening period has generated many hundreds of documents. In selecting the following documents as a helpful reading list, I have tried to select only from the more recent documents and those that are overviews of transition technologies rather than reference specifications for individual technologies.

[1] Jari Arkko and Fred Baker, "Guidelines for Using IPv6 Transition Mechanisms during IPv6 Deployment," Internet Draft, Work in Progress, December 2010.

The document discusses the IPv6 deployment models and migration tools, and considers what appears to be effective in networks to date. This Internet Draft, **`draft-arkko-ipv6-transition-guidelines-14.txt`***, is about to be published as an Informational RFC.*

[2] Brian Carpenter and Sheng Jian, "Emerging Service Provider Scenarios for IPv6 Deployment," RFC 6036, October 2010.

This document describes practices and plans that are emerging among Internet Service Providers for the deployment of IPv6 services, using data collected in a survey of numerous ISPs carried out in early 2010.

[3] Reinaldo Penno, Tarun Saxena, Mohamed Boucadair, and Senthil Sivakumar, "Analysis of 64 Translation," Internet Draft, Work in Progress, **draft-ietf-behave-64-analysis-01**, January 2011.

This paper is a working document of the IETF's BEHAVE Working Group. The document notes that because of specific problems, NAT-PT was deprecated by the IETF as a mechanism to perform IPv6-IPv4 translation. Since then, new efforts have been undertaken within IETF to standardize alternative mechanisms to perform IPv6-IPv4 translation. This document evaluates how the new translation mechanisms avoid the problems that caused the IETF to deprecate NAT-PT.

[4] Fred Baker, Xing Li, and Kevin Yin, "Framework for IPv4/IPv6 Translation," Internet Draft, Work in Progress, August 2010.

It is common in the IETF these days to generate a "framework" document as part of the process of developing technical specifications. This draft is a framework document for the general IPv4/IPv6 translation technology. This Internet Draft, **draft-ietf-behave-v6v4-framework-10.txt***, will soon be published as an Informational RFC.*

[5] Elwyn Davies, Suresh Krishnan, and Pekka Savola, "IPv6 Transition/Coexistence Security Considerations," RFC 4942, September 2007.

The transition into a dual-stack environment, while attempting to preserve the integrity of a single service regime, presents numerous security concerns. This document is a good overview of such concerns.

[6] Dan Wing and Andrew Yourtchenko, "Improving User Experience with IPv6 and SCTP," *The Internet Protocol Journal,* Volume 13, No. 3, September 2010.

Building efficient applications in a dual-stack world can be very challenging. It is often the case that poor management of a dual-stack system can make the user experience far slower than just continuing in the IPv4 world. One way to redress this problem is to exchange sequential testing of IPv6 and IPv4 connectivity into a parallel operation—both protocols at once. This article explains the concept.

GEOFF HUSTON, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of numerous Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005; he served on the Board of Trustees of the Internet Society from 1992 until 2001. E-mail: **gih@apnic.net**

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

June 2011 Volume 14, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

ISSN 1944-1134

From The Editor

The process of adding security to various components of Internet architecture reminds me a little bit of the extensive seismic retrofitting that has been going on in California for decades. The process is slow, expensive, and occasionally intensified by a strong earthquake after which new lessons are learned. Over the past 13 years this journal has carried many articles about network security enhancements: *IP Security* (IPSec), *Secure Sockets Layer* (SSL), *Domain Name System Security Extensions* (DNSSEC), Wireless Network Security, and E-mail Security, to name but a few. In this issue we look at routing security again, specifically the efforts underway in the *Secure Inter-Domain Routing* (SIDR) Working Group of the IETF to provide a secure mechanism for route propagation in the *Border Gateway Protocol* (BGP). The article is by Geoff Huston and Randy Bush.

Our second article discusses *Site Multihoming* in IPv6. Multihoming is a fairly common technique in the IPv4 world, but as part of the development and deployment of IPv6, several new and improved solutions have been proposed. Fred Baker gives an overview of these solutions and discusses the implications of each proposal.

By all accounts, *World IPv6 Day* was a successful demonstration and an important step toward deployment of IPv6 in the global Internet. Several major sites left IPv6 connectivity in place after the event, an encouraging sign. Discussions are already underway for another similar event, this time perhaps lasting for as long as a week. Phil Roberts gives an overview of what happened on June 8 and provides pointers to some of the important lessons learned from this experiment.

I want to take a moment to mention the IPJ subscription renewal campaign. As you know, each subscriber is issued a unique subscription ID that, coupled with an e-mail address, gives access to the subscription database by means of a "magic URL." Unfortunately, sometimes the e-mail containing this URL may not arrive in the subscriber's mailbox, perhaps because of spam filtering. Additionally, readers change e-mail addresses as well as postal addresses. If your subscription has expired or you have changed e-mail, postal mail, or delivery preference, send an e-mail to **ipj@cisco.com** with the updated information and we will make sure your subscription is re-instated. The purpose of the renewal campaign is to ensure that we are sending copies of IPJ to the correct addresses and only to those who prefer paper copies. IPJ is always available via our website at **http://cisco.com/ipj**

—Ole J. Jacobsen, Editor and Publisher

ole@cisco.com

Securing BGP with BGPsec

by Geoff Huston, APNIC and Randy Bush, IIJ

For many years the fundamental elements of the Internet: *names* and *addresses*, were the source of basic structural vulnerabilities in the network. With the increasing momentum behind the deployment of *Domain Name System Security Extensions* (DNSSEC)[0], there is some cause for optimism that we have the elements of securing the name space now in hand, but what about addresses and routing? In this article we will look at current efforts within the *Internet Engineering Task Force* (IETF) to secure the use of addresses within the routing infrastructure of the Internet, and the status of current work of the *Secure Inter-Domain Routing* (SIDR) Working Group.

We will look at the approach the SIDR Working Group has taken, and examine the architecture and mechanisms that it has adopted as part of this study. This work was undertaken in three stages: the first concentrated on the mechanisms to support attestations relating to addresses and their use; the second looked at how to secure origination of routing announcements; and the third looked at how to secure the transitive part of *Border Gateway Protocol* (BGP) route propagation.

Supporting Attestations About Addresses Through the RPKI

Prior work in the area of securing the Internet routing system has focused on the operation of BGP in an effort to secure the operation of the protocol and validate, as far as is possible, the contents of *BGP Update* messages. Some notable contributions in more than a decade of study include *Secure-BGP* (S-BGP)[1, 16], *Secure Origin BGP* (soBGP)[2], *Pretty Secure BGP* (psBGP)[3], IRR[4], and the use of an *Autonomous System* (AS) *Resource Record* (RR) in the *Domain Name System* (DNS), signed by DNSSEC[5].

The common factor in this prior work was that they all required, as a primary input, a means of validating basic assertions relating to origination of a route into the interdomain routing system: that the IP address block and the AS numbers being used are valid and that the parties using these IP addresses and AS numbers in the context of routing advertisement are properly authorized to so do.

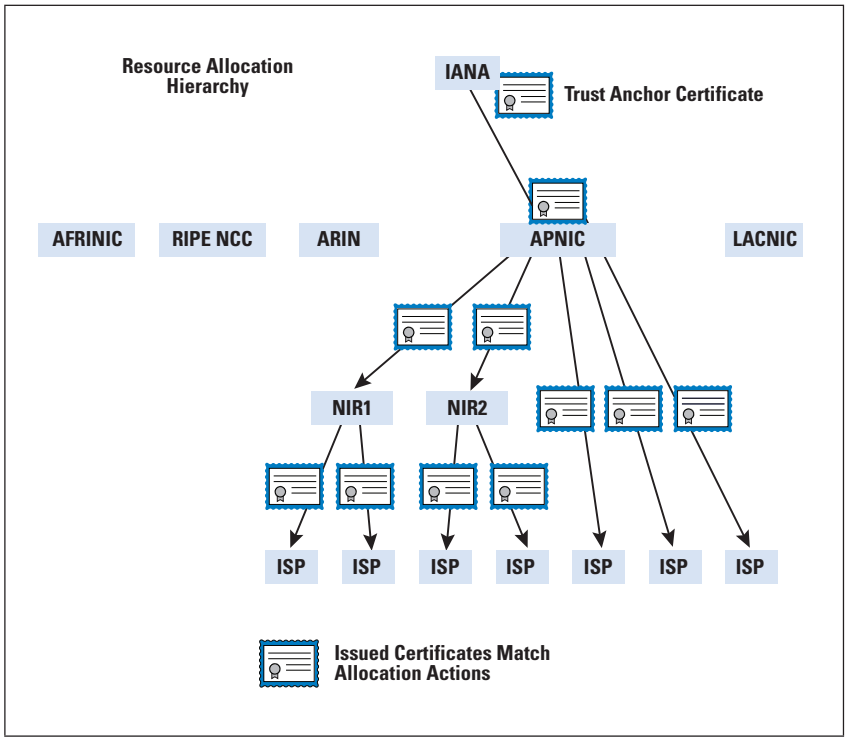
The approach adopted by SIDR for the way in which trust is formalized in the routing environment is through the use of *Resource Certificates*. These certificates are X.509 certificates that conform to the *Public-Key Infrastructure X.509* (PKIX) profile[6]. They also contain an extension field that lists a collection of IP resources (IPv4 addresses, IPv6 addresses, and AS Numbers)[7]. These certificates attest that the certificate issuer has granted to the certificate subject a unique "right-of-use" for the associated set of IP resources, by virtue of a resource allocation action.

This concept mirrors the resource allocation framework of the Internet *Assigned Numbers Authority* (IANA), the *Regional Internet Registries* (RIRs), operators, and others, and the certificate provides a means for a third party (relying party) to formally validate assertions related to resource allocations[8].

The hierarchy of the *Resource Public Key Infrastructure* (RPKI) is based on the administrative resource allocation hierarchy, where resources are distributed from the IANA to the RIRs, *Local Internet Registries* (LIRs), *National Internet Registries* (NIRs), and end users. The RPKI mirrors this allocation hierarchy with certificates that match current resource allocations (Figure 1).

The *Certification Authorities* (CAs) in this RPKI correspond to entities that have been allocated resources. Those entities are able to sign authorities and attestations, and to do so they use specific-purpose *End Entity* (EE) certificates. This additional level of indirection allows the entity to customize each issued authority for specific subsets of number resources that are administered by this entity. Through the use of single-use EE certificates, the issuer can control the validity of the signed authority through the ability to revoke the EE certificate used to sign the authority. As is often the case, a level of indirection comes in handy.

Figure 1: Hierarchy of the RPKI



Signed attestations relating to addresses and their use in routing are generated by selecting a subset of resources that will be the subject of the attestation, by generating an EE certificate that lists these resources, and by specifying validity dates in the EE certificate that correspond to the validity dates of the authority. The authority is published in the RPKI repository publication point of the entity. The RPKI makes conventional use of *Certificate Revocation Lists* (CRLs) to revoke certificates that have not expired but are no longer valid. Every Certification Authority in the RPKI regularly issues a CRL according to the declared CRL update cycle of the Certification Authority. A Certification Authority certificate may be revoked by an issuing authority for numerous reasons, including key rollover, the reduction in the resource set associated with the certificate subject, or termination of the resource allocation. To invalidate an object that can be verified by a given EE certificate, the Certification Authority that issued the EE certificate can revoke the corresponding EE certificate.

The RPKI uses a distributed publication framework, wherein each Certification Authority publishes its products (including EE certificates, CRLs, and signed objects) at a location of its choosing. The set of all such repositories forms a complete information space, and it is fundamental to the model of securing BGP in the public Internet that the entire RPKI information space be available to every *Relying Party* (RP). It is the role of each RP to maintain a local cache of the entire distributed repository collection by regularly synchronizing each element in the local cache against the original repository publication point. To assist RPs in the synchronization task, each RPKI publication point uses a *manifest,* a signed object that lists the names (and hash values) of all the objects published at that publication point. It is used to assist RPs to ensure that they have managed to synchronize against a complete copy of the material published at the Certification Authority publication point.

The utility of the RPKI lies in its ability to validate digitally signed information and, therefore, give relying parties some confidence in the validity of signed attestations about addresses and their use. The particular utility of the RPKI is not as a means of validation of attestations of an individual's identity or that individual's role, but as a means of validating that person's authority to use IP address resources. Although it is possible to digitally sign any digital object, it has been suggested that the RPKI system uses a very small number of standard signed objects that have particular meaning in the context of routing security.

Securing Route Origination

The approach adopted by SIDR to secure origination of routing information is one that uses a particular signed authority, a *Route Origination Authorization* (ROA)[10]. An ROA is an authority created by a prefix holder that authorizes an AS to originate one or more specific route advertisements into the interdomain routing system.

An ROA is a digital object formatted according to the *Cryptographic Message Syntax Specification* (CMS)[11] that contains a list of address prefixes and one AS number. The AS is the specific AS being authorized to originate route advertisements for one or more of the address prefixes in the ROA. The CMS object also includes the EE resource certificate for the key used to verify the ROA. The IP Address extension in this EE certificate must encompass the IP address prefixes listed in the ROA contents.

The ROA conveys a simple authority. It does not convey any further routing policy information, nor does it convey whether or not the AS holder has even consented to actually announce the prefix(es) into the routing system. The associated EE certificate is used to control the validity of the ROA, and the CMS wrapper is used to securely bind the ROA and the EE certificate within a single signed structure.

There is one special ROA, one that authorizes AS 0 to originate a route. Because AS 0 is a reserved AS that should never be used by a BGP speaker, this ROA is a "negative" authority, used to indicate that no AS has authority to originate a route for the address prefix(es) listed in the ROA.

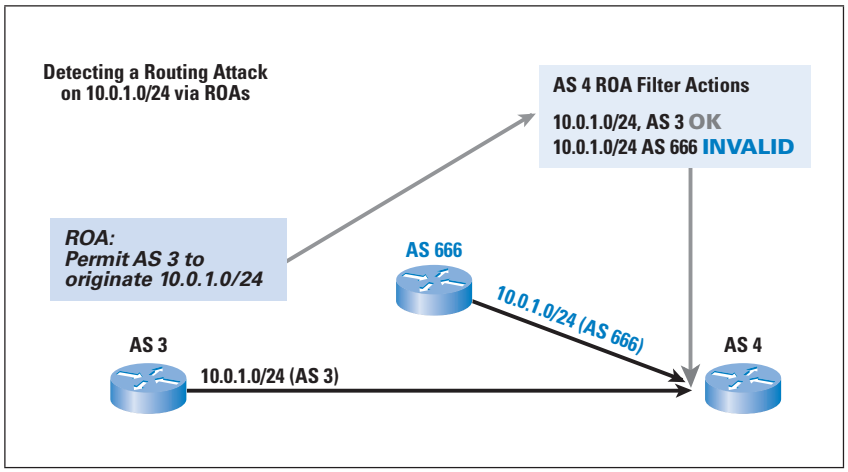
If the entire routing system were to be populated with ROAs, then identification of an invalid route advertisement would be directly related to detection of an invalid ROA or a missing ROA. However, in a more likely scenario of partial use of ROAs (such as when only some legitimate route originations are authorized in a ROA), the absence of an ROA cannot be interpreted simply as an unauthorized use of an address prefix. This scenario leads to the use of a tri-state validation process for routes, as follows.

If a given route matches exactly the information contained in an ROA whose EE certificate can be validated in the RPKI (a "valid" ROA), then the route can be regarded as a "valid" origination. Where the address prefix matches that in a valid ROA but the origination AS does not match the AS number in the ROA, and there are no other valid ROAs that explicitly validate the announcing AS, then the route can be considered to be "invalid." Also, where the address prefix is more specific than that of a valid ROA, and there are no other valid ROAs that match the prefix, then the route can also be considered "invalid." Where the prefix in a route is not described in any ROA and is not a more specific prefix of any ROA, the route has an "unknown" validation outcome.

These three potential outcomes can be considered a set of relative local preferences. Routes whose origins can be considered "valid" are generally proposed to be preferred over routes whose origins are unknown, which, in turn, can generally be preferred over routes whose origins are considered invalid. However, such relative preferences are a matter to be determined by local routing policy. Local policies may choose to adopt a stricter policy and, for example, discard routes with an invalid validation outcome[12].

The way in which ROAs are used to validate the origin of routes in BGP differs from many previous proposals for securing BGP. In this framework the ROAs are published in the RPKI distributed repository framework. Each RP can use the locally cached collection of valid ROAs to create a validation filter collection, with each element of the set containing an address, prefix size constraints, and an originating AS. It is this filter set—rather than the ROAs themselves—that are fed to the local routers[13]. An example of the way in which ROAs can be used to detect prefix hijack attempts is shown in Figure 2.

Figure 2: Use of ROAs to detect Unauthorized Route Origination



The model of injecting validation of origination into the BGP domain is an example of a highly modular and piecemeal deployment. There are no changes to the BGP protocol for this origin validation part of the secure routing framework.

The process of securing origination starts with the address holder, who generates local keys and requests certification of their address space from the entity from whom their addresses were allocated or assigned. With this Certification Authority resource certificate, the address holder is then in a position to generate an EE certificate and a ROA that assigns an authority for a nominated AS to advertise a route for an address prefix drawn from its address holdings. The one condition here is that if an address holder issues a ROA for an address prefix providing an authority for one AS to originate a route for this prefix, then the address holder is required to issue ROAs for all the ASs that have been similarly authorized to originate a route for this address prefix. The address holder publishes this ROA in its publication point in the distributed RPKI repository structure.

Relying parties can configure a locally managed cache of the distributed RPKI repository and collect the set of valid ROAs. They can then, with the dedicated RPKI cache-to-router protocol[13], maintain, on a set of "client" routers, the set of address prefix/originating AS authorities that are described in valid ROAs. The BGP-speaking router can use this information as an input to the local route decision process.

This model of operation supports piecemeal incremental deployment, wherein individual address holders may issue ROAs to authorized routing advertisements independent of the actions of other address holders. Also, ASs may deploy local validation of route origination independently of the actions of other ASs. And given that there are no changes to the operation of BGP, then there are no complex inter-dependencies that hinder piecemeal incremental deployment of this particular aspect of securing routing.

Securing Route Propagation: BGPsec

Origin validation as described earlier does not provide cryptographic assurance that the origin AS in a received BGP route was indeed the originating AS of this route. A malicious BGP speaker can synthesize a route as if it came from the authorized AS. Thus, it is very useful in detecting accidental misannouncements, but origination validation does little to prevent malicious routing attacks from a determined attacker.

In looking at the operation of the BGP protocol, some parts of the protocol interaction are strictly local between two BGP-speaking peers, such as advising a peer of local attributes. Another part of the BGP protocol is a "chained" interaction, in which each AS adds information to the protocol object. This attribute of a BGP update, the *AS Path,* is not only useful to detect and prevent routing loops, it is also used in the BGP best-path-selection algorithm.

A related routing security question concerns the validity of this "chained" information, namely the AS Path information contained in a route. Within the operation of the BGP protocol, each AS that propagates an update to its AS neighbors is required to add its AS number to the AS Path sequence. The inference is that at any stage in the propagation of a route through the interdomain routing system, the AS Path represents a viable AS transit sequence from the local AS to the AS originating the route. This AS Path attribute of a route is used for loop detection. Locally, the AS Path may also be used as input to a local route policy process, using the length of the AS Path as a route metric.

Attacks on the AS Path can be used to subvert the routing environment. A malicious BGP speaker may manipulate the AS Path to prevent an AS from accepting a route by adding its AS number to the AS Path, or it may attempt to make a particular route more likely to be selected by a remote AS by stripping out ASs from the AS Path. Accordingly, it is important to equip a secure BGP framework with the ability to validate the authenticity of the AS Path presented in a BGP update[14].

When attempting to validate an AS path, many potential validation questions must be addressed.

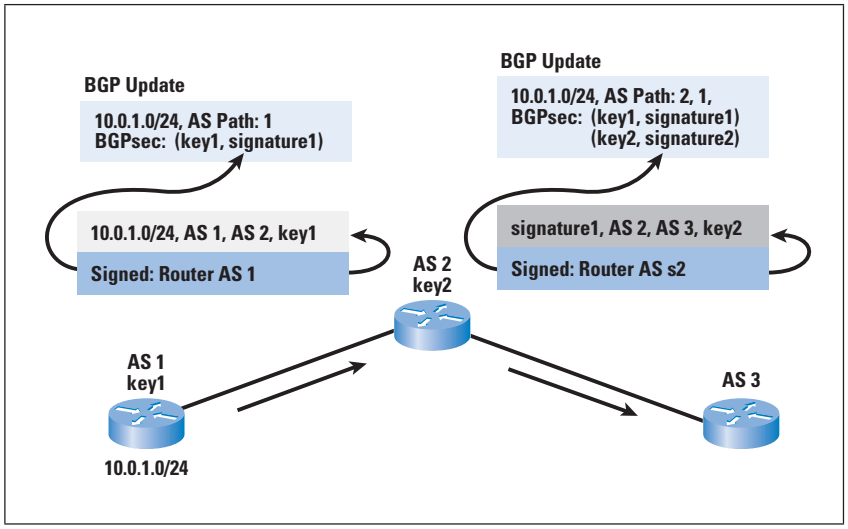
- The first and weakest question is: Are all ASs in the AS Path valid ASs?
- A slightly stronger validation question is: Do all the AS pairs in the AS Path represent valid AS adjacencies (where both ASs in the pair-wise association are willing to attest to their mutual adjacency in BGP)?
- A even stronger question is: Does the sequence of ASs in the AS Path represent the actual propagation path of the BGP route object?

This last question forms the basis for the SIDR activity in defining an AS Path validation framework, BGPsec. This attempt is to assure a BGP speaker that the operation of the BGP protocol is operating correctly and that the content of a BGP update correctly represents the inter-AS propagation path of the update from the point of origination to the receiver of the route. This tool is not the same as a policy validation tool and it does not necessarily assure the receiver of the route that this update conforms to the routing policies of neighboring BGP speakers. This route also does not necessarily reflect the policy intent of the originator of the route. The BGPsec framework proposed for securing the AS Path also uses a local RPKI cache, but it includes an additional element of certification. The additional element of the security credentials used here is an extension to the certification of AS numbers with a set of operational keys and their associated certificates used for signing update messages on *External Border Gateway Protocol* (eBGP) routers in the AS. These "router certificates" can sign BGP update attributes in the routing infrastructure, and the signature can be interpreted as being a signature made "in the name of" an AS number.

In the BGPsec framework, eBGP-speaking routers within the AS have the ability to "sign" a BGP update before sending it. In this case, the added signature "covers" the signature of the received BGP update, the local AS number, the AS number to which the update is being sent, as well as a hash of the public key part of the router key pair used to sign route updates.

The couplet of the public key hash and the signature itself are added to the BGP protocol update as BGPsec update attributes. As the update traverses a sequence of transit ASs, each eBGP speaker at the egress of each AS adds its own public key hash and digital signature to the BGPsec attribute sequence (Figure 3).

Figure 3: BGPsec AS Path Protection



This interlocking of signatures allows a receiver of a BGP update to use the interlocking chain of digital signatures to validate (for each AS in the AS Path) that the corresponding signature was correctly generated "in the name of" that AS in the AS Path, and that the next AS in the path matches the next AS in the signed material. The "forward signing" that includes the AS to which the update is being sent prevents a man-in-the-middle attack of the form of taking a legitimate outbound route announcement destined for one neighbor AS and redirecting it to another AS. But this signing of the AS Path is not quite enough to secure the route update, because the AS Path needs to be coupled to the actual address prefix by the route originator. The route originator needs to sign across not only the local AS and the AS to whom the route update is being sent, but also the address prefix and the expiry time of the route. This action allows the path to be "bound" to the prefix and prevents a man-in-the-middle from splicing a signed path or signed-path fragment against a different prefix.

If the signatures that "span" the AS Path in the BGP update can all be validated, then the receiver of the BGP update can validate, in a cryptographic sense, the currency of the routing update. It can also validate that the route update was propagated across the inter-AS routing space in a manner that is faithfully represented in the AS Path of the route.

The expiry time of the EE certificates used in conjunction with signed route updates introduces a new behavior into BGPsec. In the context of BGP, an announced route remains current until it is explicitly withdrawn or until the peer session that announced the route goes down. This property of BGP introduces the possibility of "ghost-route" attacks in BGP, wherein a BGP speaker fails to propagate a withdrawal in order to divert the consequent misdirected traffic from its peers.

In BGPsec, all route advertisements are given an expiry time by the originator of the route. This expiry time corresponds to the "notAfter" time of the EE certificate used to sign the protocol update, after which time the route is considered invalid. The implication is that a route originator is required to readvertise the route, and refresh the implicit expiry timer of the associated digital signature at regular intervals.

This approach to route-update validation is not quite the "light-touch" of origination validation. In this case the mechanism requires the use of a new BGP attribute and negotiation of a new BGP capability between eBGP peers, in turn meaning that the model of incremental deployment is one that is more "viral" than truly piecemeal. By "viral" we mean that this model is one of incremental deployment in which direct eBGP peers of a BGPsec-speaking AS will be able to speak BGPsec between themselves in a meaningful way. In turn these adjacent ASs can offer to speak BGPsec with their eBGP peers, and so on. This reality does not imply that BGPsec deployment must necessarily start from a single AS, but it does imply that communities of interconnected ASs all speaking BGPsec will be able to provide assurance via BGPsec on those routes originated and propagated within that community of interconnected ASs. It also implies that the greatest level of benefit to adopters of secure BGP will be realized by ASs that adopt BGPsec as a connected community of ASs.

Other changes to the behavior of BGP are implied by this mechanism. BGP conventionally permits "update packing," where numerous address prefixes can be placed in a single update message if they share a common collection of attributes, including the AS Path. At this stage it appears that such update packing would not be supported in secure BGP, and each update in secure BGP would refer to a single prefix. Obviously this situation would have some effect on the level of BGP traffic, but early experiments suggest not at an unreasonable cost.

There are further effects on BGP that have not been fully quantified in studies to date. The addition of a compound attribute of a signature and a public key identifier for every AS in the AS Path has size implications on the amount of local storage a secure BGP speaker will need to store these additional per-prefix per-peer attributes. It also has broader implications if used in conjunction with current proposals for multipath BGP where multiple paths, in addition to the "best" path, are propagated to eBGP peers. Also, the computational load of validation of signatures in secure BGP is significantly higher in terms of the number of cryptographic operations that are required to validate a BGP update.

However, BGPsec is not intended to "tunnel" across those parts of the interdomain routing space that do not support BGPsec capabilities. When an update leaves a BGPsec realm, the BGPsec signature attributes of the route are stripped out, so the storage overheads of BGPsec are not seen by other BGP speakers.

Similarly, the periodic updates that result from the expiry timer should not propagate beyond the BGPsec realm. If the boundary is prepared to perform BGP update packing to non-BGPsec peers, then even the unpacked update overhead is not carried outside of the BGPsec realm.

It is also noted that the "full" load of BGPsec would only necessarily be carried by "transit" ASs; that is, those ASs that propagate routes on behalf of other ASs. Historically we see some 15 percent of ASs are "transit" ASs, while all other ASs behave as "stub" ASs that only originate routes and do not appear to transit routes for others. Such stub ASs can support a "lightweight" simplex version of BGPsec that can either point a default route to its upstream AS provider or trust its upstream ASs to perform BGPsec validation. In this case the stub AS needs to provide BGPsec signed originated routes to its upstream ASs, but no more.

Conclusion

The work on the specification of the RPKI itself and the specification of origin validation is nearing a point of logical completion of the first phase of standardization within the IETF, and the working draft documents are being passed from the working group into the review process leading to their publication as proposed standard RFCs. The RIRs are in the process of launching their RPKI services based on these specifications, and the initial deployment of working code has been made by numerous parties, who are also working on integration of origination validation in BGP implementations.

The work on securing the AS Path is at an earlier phase in the development process, and the SIDR Working Group is considering the initial design material. It is expected to take a similar path of further review and refinement in light of developing experience and study of the proposed approach.

The RPKI has been designed as a robust and simple framework. As far as possible, existing standards, technologies, and processes have been exploited, reflecting the conservatism of the routing community and the difficulty in securing rapid, widespread adoption of novel technologies.

Acknowledgements

The work described here is the outcome of the efforts of many individuals who have contributed to securing BGP over a period that now spans two decades, and certainly too many to ensure that all the contributors are recognized here. Instead, the authors would like to acknowledge their work and trust that the mechanisms described here are a faithful representation of the cumulative sum of their various contributions.

References

[0] Miek Gieben, "DNSSEC: The Protocol, Deployment, and a Bit of Development," *The Internet Protocol Journal,* Volume 7, No. 2, June 2004.

[1] Stephen Kent, Charlie Lynn, and Karen Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE Journal on Selected Areas in Communications,* Volume 18, No. 4, pp 582–592, April 2000.

[2] Russ White, "Securing BGP through secure origin BGP," *The Internet Protocol Journal,* Volume 6, No. 3, September 2003.

[3] Paul van Oorschot, Tao Wan, and Evangelos Kranakis, "On Inter-domain Routing Security and Pretty Secure BGP (psBGP)," *ACM Transactions on Information and System Security,* Volume 10, No. 3, July 2007.

[4] Geoffrey Goodell, William Aiello, Timothy Griffin, John Ioannidis, and Patrick D. McDaniel, "Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing," *Proceedings of Internet Society Symposium on Network and Distributed System Security* (NDSS '03), February 2003.

[5] Tony Bates, Randy Bush, Tony Li, and Yakov Rekhter, "DNS-based NLRI origin AS verification in BGP," Internet Draft, Work in Progress, July 1998.

[6] David Cooper et al., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280, May 2008.

[7] Charlie Lynn, Stephen Kent, and Karen Seo, "X.509 Extensions for IP Addresses and AS Identifiers," RFC 3779, June 2004.

[8] Matt Lepinski and Stephen Kent, "An Infrastructure to Support Secure Internet Routing," Internet Draft, Work in Progress, February 2008.

[9] Geoff Huston, George Michaelson, and Robert Loomans, "A Profile for X.509 PKIX Resource Certificates," Internet Draft, Work in Progress, September 2008.

[10] Matt Lepinski, Stephen Kent, and Derrick Kong, "A Profile for Route Origin Authorizations (ROAs)," Internet Draft, Work in Progress, July 2008.

[11] Russ Housley, "Cryptographic Message Syntax (CMS)," RFC 3852, July 2004.

[12] Geoff Huston and George Michaelson, "Validation of Route Origination using the Resource Certificate PKI and ROAs," Internet Draft, Work in Progress, November 2010.

[13] Randy Bush and Rob Austein, "The RPKI/Router Protocol," Internet Draft, Work in Progress, March 2011.

[14] Kim Zetter, "Revealed: The Internet's Biggest Security Hole," *WIRED,* August 2008, **http://www.wired.com/threatlevel/2008/08/revealed-the-in/**

[15] Geoff Huston, "Resource Certification," *The Internet Protocol Journal,* Volume 12, No. 1, March 2009.

[16] Stephen Kent, "Securing the Border Gateway Protocol," *The Internet Protocol Journal,* Volume 6, No. 3, September 2003.

Ed.: A version of this article also appeared in *The IETF Journal,* Volume 7, Issue 1, July 2011. *The IETF Journal* can be obtained from: **http://isoc.org/ietfjournal/**

GEOFF HUSTON, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of numerous Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005; he served on the Board of Trustees of the Internet Society from 1992 until 2001. E-mail: **gih@apnic.net**

RANDY BUSH is a Research Fellow and Network Operator at Internet Initiative Japan (IIJ), Japan's first commercial ISP. He specializes in network measurement, especially routing, network security, routing protocols, and IPv6 deployment. Randy has been in computing for 45 years, and has a few decades of Internet operations experience. He was the engineering founder of Verio, which is now NTT/Verio. He has been heavily involved in transferring Internet technologies to developing economies for more than 20 years. E-mail: **randy@psg.com**

Views of IPv6 Site Multihoming

by Fred Baker, Cisco Systems

In today's Internet, *site multihoming*—an edge network configuration that has more than one service provider but does not provide transit communication between them—is relatively common. Per the statistics at **www.potaroo.net**, almost 40,000 *Autonomous Systems* are in the network, of which about 5,000 seem to offer transit services to one or more customers. The rest are in terminal positions, possibly meaning three things. They could be access networks, broadband providers offering Internet access to small companies and residential customers; they could be multihomed edge networks; or they might be networks that intend to multihome at some point in the future. The vast majority, on the order of 75 percent, are multihomed or intend to multihome. That is but one measure; you do not have to use *Border Gateway Protocol* (BGP) routing to have multiple upstream networks. Current estimates suggest that there is one multihomed entity per 50,000 people worldwide, and one per 18,000 in the United States.

We also expect site multihoming to become more common. A current proposal in Japan suggests that each home might be multihomed; it would have one upstream connection for Internet TV, and one or more other connections provided by *Internet Service Providers* (ISPs), operating over a common *Digital Subscriber Line* (DSL) or fiber-optic infrastructure. That scenario has one multihomed entity for every four people.

Why do edge networks multihome? Reasons vary. In the Japanese case just propounded, it is a fact of life—users have no other option. In many cases, it is a result of a work arrangement, or a strategy for achieving network reliability through redundancy.

For present purposes, this article considers scaling targets derived from a world of 10 billion people (circa 2050), and a ratio of one multihomed entity per thousand people—on the order of 10,000,000 multihomed entities at the edge of the Internet. Those estimates may not be accurate 40 years from now, but given current trends they seem like reasonable guesses.

RFC 1726[1], the technical criteria considered in the selection of what at the time was called *IP Next Generation* (IPng), did not mention multihoming per se. Even so, among the requirements are scalable and flexible routing, of which multihoming is a special case. When IPv6 was selected as the "next generation," multihoming was one of the topics discussed. The Internet community has complained that this particular goal was not fulfilled. Several proposals have been proffered; unfortunately, each has benefits, and each has concerns. No single perfect solution is universally accepted.

In this article, I would like to look at the alternatives proposed and consider the effects they have. In this context, the goals set forth in RFC 3582[2] are important; many people tried to state what they would like from a multihoming architecture, and the result was a set of goals that solutions only asymptotically approach.

The proposals considered in this article include:

- *Provider Independent Addressing,* also known as *BGP Multihoming*
- *Exchange-Based Addressing*
- *Shim6,* also known as *Level 3 Multihoming*
- *Identifier-Locator Network Protocol* (ILNP)
- *Network Prefix Translation,* also known as *NAT66*

BGP Multihoming

BGP Multihoming involves a mechanism relatively common in the IPv4 Internet; the edge network either becomes a member of a *Regional Internet Registry* (RIR) [APNIC, RIPE, LACNIC, AFRINIC, ARIN] and from that source obtains a *Provider-Independent* (PI) prefix, or obtains a *Provider-Allocated* (PA) prefix from one provider and negotiates contracts with others using the same prefix. In any case, it advertises the prefix in BGP, meaning that all ISPs—including in the PA case—the provider that allocated it, must carry it as a separate route in their routing tables.

The benefit to the edge is easily explained, and in the case of large organizations it is substantial. Consider the case of Cisco Systems, whose internal network rivals medium-sized ISPs for size and complexity. With about 30 *Points of Attachment* (PoAs) to the global Internet, and at least as many service providers, Cisco has an IPv6 /32 PI prefix, and hundreds of offices to interconnect using it. One possible way to enumerate the Cisco network would be to use the next five bits of its address (32 /37 prefixes) at its PoAs, and allocate prefixes to its offices by the rule that if their default route is to a given PoA, their addresses are derived from that PoA. By advertising the PoAs /37 and a backup /32 into the Internet core at each PoA, Cisco could obtain effective global routing. It would also obtain relative simplicity for its internal network—only one subnet is needed on any given *Local-Area Network* (LAN) regardless of provider count or addressing, and routing can be optimized independently from the outside world.

The problem that arises with PI addressing, if taken to its logical extreme, is that the size of the routing table explodes. If every edge network obtains a PI prefix—neglecting for the moment both BGP traffic engineering and the kind of de-aggregation suggested in Cisco's case—the logical outcome of enumerating the edge is a routing table with on the order of 107 routes. The memory required to store the routing table, and in the *Secure Interdomain Routing* (SIDR) case the certificates that secure it, is one of the factors in the cost of equipment. The volume of information also affects the time it takes to advertise a full routing table, and in the end the amount of power that a router uses, the heat it produces, and a switching center's air conditioning requirements. Thus both the capital cost of equipment used in transit networks and the cost of operations would be affected. In effect, the Internet becomes the "poster child" for the *Tragedy of the Commons*.

Exchange-Based Addressing

Steve Deering proposed the concept of exchange-based addressing at the IETF meeting in Stockholm in 1995, under the name *Metropolitan Addressing*. In this model, prefixes do not map to companies, but to Internet exchange consortia, likely regional. One organizing principle might be to associate an Internet exchange with each commercial airport worldwide, about 4000 total, resulting in a global routing table on the same order of magnitude in size. Edge networks, including residential networks, within that domain obtain their prefix from the exchange, and they are used by any or all ISPs in the region. Routes advertized to other regions, even within the same ISP, are aggregated to the consortium prefix.

The benefits to the edge network in exchange-based addressing are similar to the benefits of PI addressing for a large corporation. In effect, the edge networks served by an exchange consortium behave like the "departments" of a "user consortium," and they enjoy great independence from their upstream providers. They can multihome or move between providers without changing their addressing, and on a global scale the routing table is contained to a small multiple of the number of such consortia.

However, the benefit to users is in most cases a detriment to their ISPs; the ISPs are forced to maintain routes to each user network served by the consortium—or at least routes for their own customers and a default route to the exchange. Thus, the complexity of routing is moved from the transit core to the access networks serving regional consortia. In addition, if there is no impediment to a user flitting among ISPs, users can be expected to flit, imposing business costs.

The biggest short-term effect on the ISP might well be the reengineering of its transit contracts. In today's Internet, a datagram sent by users to their ISPs is quickly shuttled to the destination's ISPs, which then carry it over the long haul. In an exchange-based network, there is no way to remotely determine which local ISP or ISP instance is serving a given customer.

Hence, the sender's ISP carries the datagram until it reaches the remote consortium, whence it switches to the access network serving the destination. One could argue that a "sender-pays" model might have benefits, but it is very different from the present model.

The edge network has problems, too. If the edge network is sufficiently distributed, it will have services in several exchange consortia, and therefore several prefixes. Although there is nothing inherently bad about that, it may not fit the way a cloud computing environment wants to move virtual hosts around, or miss other requirements.

Level 3 Multihoming: Shim6

The IETF's *shim6* model[9] starts from the premise that edge networks obtain their prefixes from their upstream ISPs—PA Addressing. If a typical residential or small business does so, there is no question of advertising its individual route everywhere; the ISP can route internally as its needs to, but globally, the number of ISPs directs the size of the routing table. If that is, as `potaroo` suggests, on the order of 10,000, the size of the routing table will be on the same order of magnitude.

The benefit to the ISP should be obvious; it does not have to change its transit contracts, and although there will be other concerns, it does not have the routing table ballooning memory costs or route exchange latencies.

However, as exchange-based addressing moves operational complexity from the transit core to the access network, shim6 moves such complexities to the edge network itself and to the host in it. If a network has multiple upstream providers, each LAN in it will carry a subnet from each of those providers—not one subnet per LAN, but as many as the providers of the host's LAN will use. At this point, the ingress filtering of RFC 3704[21] at the provider becomes a problem at the edge; the host must select a reasonable address for any session it opens, and must do so in the absence of specific knowledge of network routing. A wrong guess can have dramatic effects; a session routed to the wrong provider may not work at all, and an unfortunate address choice can change end-to-end latency from tens of milliseconds to hundreds or worse by virtue of backbone routing.

Application layer referrals and other application uses of addresses also have difficulties. Although the address a session is using will work both within and without the network, if a host has more than one address, one of the other addresses may be more appropriate to a given use. Hence, the application that really wants to use addresses is saddled with finding all of the addresses that its own host or a peer host might have.

There is also an opportunity. TCP today associates sessions with their source and destination addresses. The shim6 model, implemented in the *Stream Control Transmission Protocol* (SCTP)[17] and *Multipath TCP* (MPTCP)[16], allows a session to change its addresses, meaning that a session can survive a service provider outage. Doing the same in TCP requires the insertion of a shim protocol between IP and TCP; at the Internet layer, the address might change, but the shim tracks the addresses for TCP.

There are, of course, ways to solve the outstanding problems. For simple cases, RFC 3484[3, 4] describes an address-selection algorithm that has some promise. In the Japanese case, a residential host might use link-local addresses within its own network, addresses appropriate to the television service on its TV and set-top box, and an ISP's prefix for everything else. If there is more than one router in the residential LAN serving more than one ISP, exit routing can be accomplished by having the host send data using an ISP's source address to the router from which it learned the prefix. When the network becomes more complex, though, we are looking at new routing protocols that can route based on a combination of the source and the destination addresses, and we are looking at network management methodologies that make address management simpler than it is today, adding and dropping subnets on LANs—and as a result renumbering networks—without difficulty. It also implies a change to the typical host implementing the shim protocol. Those technologies either do not exist or are not widely implemented today.

Identifier-Locator Network Protocol

The concept of separating a host's identity from its location has been intrinsic to numerous protocol suites, including the *Xerox Network Systems* (XNS), *Internetwork Packet Exchange* (IPX), and *Connectionless Network Service* (CLNS) models. In the IP community, it was first proposed in Saltzer's ruminations on naming and binding, RFC 1498[5], and in Noel Chiappa's NIMROD routing architecture, RFC 1992[6]. In short, a host (or a set of applications running on a host, or a set of sessions it participates in) has an identifier independent of its network topology, and sessions can change network paths by simply changing the topological locations of their endpoints. Mike O'Dell, in Internet Drafts in 1996 and 1997 called *8+8* and *GSE,* suggested an implementation of this scenario using the prefix in the IPv6 address as a locator and the interface identifier as an identifier. One implication of the GSE model is the use of a network prefix translation between an edge network and its upstream provider whatever prefix the edge network uses internally, in the transit backbone, the locator appears to be a PA prefix allocated by the ISP in question. As a result, the routing table, as in shim6, enumerates the ISPs in the network—on the order of 10,000.

The *Identifier-Locator Network Protocol* (ILNP) takes the solution to fruition, operating on that basic model and adding a *Domain Name System* (DNS) Resource Record and a random number nonce to mitigate on-path attacks that result from the fact that the *IPv6 Interface Identifier* (IID) is not globally unique.

As compared to the operational complexities and costs of PI Addressing, Exchange-Based Addressing, and shim6, ILNP has the advantage of being operationally simple. Each LAN has one subnet, when adding or changing providers no edge network renumbering is required, and, as noted, the cost of the global routing table does not increase. Additionally, it is trivial to load-share traffic across points of attachment to multiple ISPs, because the locator is irrelevant above the network layer. And unlike IPv4/IPv4 *Network Address Port Translation* (NAPT), the translation is stateless; as a result, sessions using *IP Security* (IPsec) *Encapsulation Security Protocol* (ESP) encryption can cross it.

In this case, the complexities of the network are transferred to the application itself, and to its transport. The application must, in some sense, know all of its "outside" addresses. It can learn them, of course, by using its domain name in referrals and other uses of the address; in some cases however, the application really wants to know the address itself. If it is communicating those addresses to other applications—the usual usage—the assumption that its view of its address is meaningful to its remote peer is, in the words of RFC 3582[2], *Unilateral Self-Address Fixing* (UNSAF), and the concerns raised in RFC 2993[7] are the result. To mitigate those concerns, ILNP excludes the locator from the TCP and *User Datagram Protocol* (UDP) pseudo-headers (and as a result from the checksum).

The implication of ILNP is, as a result, that TCP and UDP must be either changed or exchanged for other protocols such as *Stream Control Transmission Protocol* (SCTP) or *Multipath TCP* (MPTCP), and that applications must either use DNS names when referring to themselves or other systems in their network—sharply dividing between the application and network layers—or devise a means by which they can determine the full set of their "outside" addresses.

Network Prefix Translation, Also Known as NAT66

Like ILNP, *Network Prefix Translation* (NPTv6) derives from and can be considered a descendant of the GSE model. It differs from ILNP in that it defines no DNS Resource Record, defines no end-to-end nonce, and requires no change to the host, especially its TCP/UDP stacks. To achieve that, the translator updates the TCP/UDP checksum in the source and destination addresses.

If the ISP prefix is a /48 prefix, this prefix allows for load sharing of sessions across translators leading to multiple ISPs; if the ISP prefix is longer, such as a /56 or /60, the checksum update must be done in the IID, and as a result load sharing can be accomplished only across translators between the same two networks. Like ILNP and unlike IPv4/IPv4 NAPT, the translation is stateless; as a result, sessions using IPsec ESP encryption can cross it.

The complexities of the network are again transferred to the application itself, but not to its transport. The application must, in some sense, know all of its "outside" addresses. Using its domain name in referrals and other uses of the address can determine these addresses; in some cases, however, the application really wants to know the address itself. If it is communicating those addresses to other applications—the usual usage—the assumption that its view of its address is meaningful to its remote peer is, again in the words of RFC 3582[2], "UNSAF," and some of the concerns raised in RFC 2993[7] result.

The implication of NPTv6 is that applications must either use DNS names when referring to themselves or other systems in their network—sharply dividing between the application and network layers—or devise a means by which they can determine the full set of their "outside" addresses. However, the IPv6 goal of enabling any system in the network to communicate with any other given administrative support is retained.

Ways Forward

From the perspective of this author, the choice of multihoming technology will in the end be an operational choice. The practice of multihoming is proliferating and will continue to do so. There is a place for provider-independent addressing; it may not in reality make sense for 40,000 companies, but it probably does for the largest edge networks. At the other extreme, shim6-style multihoming makes sense in residential networks with a single LAN; as described earlier, there are simple approaches to making that work through reasonable policy approaches.

For the vast majority of networks in between, policy suggestions that do not substantially benefit the network or users who implement them do not have a good track record. Hence, while Exchange-Based Addressing materially assists in edge network problems, there is no substantive reason to believe that the transit backbone will implement it. Similarly, although shim6 materially helps with the capital and operational expenses of operating the transit backbone, it is not likely that edge networks will implement it.

We also have a poor track record in changing host software. For example, SCTP is in many respects a superior transport protocol to TCP—it allows for multiple streams, it is divorced from network layer addressing, and it allows endpoints to change their addresses midsession.

In a 2009 "Train Wreck" workshop at Stanford University, in which various researchers argued all day in favor of the development of a new transport with requirements much like those of SCTP, the research community acted as if ignorant of it when the protocol was brought up in conversation.

NPTv6 is not a perfect solution, but this author suspects that it will be operationally simple enough to deploy and manage and close enough to the requirements of edge networks and applications that it will, in fact, address the topic of multihoming.

References

[1] Craig Partridge and Frank Kastenholz, "Technical Criteria for Choosing IP The Next Generation (IPng)," RFC 1726, December 1994.

[2] Joe Abley, Benjamin Black, and Vijay Gill, "Goals for IPv6 Site-Multihoming Architectures," RFC 3582, August 2003.

[3] Richard Draves, "Default Address Selection for Internet Protocol version 6 (IPv6)," RFC 3484, February 2003.

[4] Arifumi Matsumoto, Jun-ya Kato, and Tomohiro Fujisaki, "Update to RFC 3484 Default Address Selection for IPv6," Internet Draft, Work in Progress, March 2011, **http://tools.ietf.org/html/draft-ietf-6man-rfc3484-revise**

[5] Jerome Saltzer, "On the Naming and Binding of Network Destinations," RFC 1498, August 1993.

[6] Isidro Castineyra, Noel Chiappa, and Martha Steenstrup, "The Nimrod Routing Architecture," RFC 1992, August 1996.

[7] Tony Hain, "Architectural Implications of NAT," RFC 2993, November 2000.

[8] Leslie Daigle, Ed., IAB "IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation," RFC 3424, November 2002.

[9] Erik Nordmark and Marcelo Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6," RFC 5533, June 2009.

[10] Ole Troan, David Miles, Satoru Matsushima, Tadahisa Okimoto, and Dan Wing, "IPv6 Multihoming without Network Address Translation," Internet Draft, Work in Progress, **http://tools.ietf.org/html/draft-ietf-v6ops-ipv6-multihoming-without-ipv6nat**

[11] Margaret Wasserman and Fred Baker, "IPv6-to-IPv6 Network Prefix Translation,"
Internet Draft, Work in Progress, http://tools.ietf.org/html/draft-mrw-nat66

[12] Ran Atkinson and Scott Rose, "DNS Resource Records for ILNP," Internet Draft, Work in Progress,
http://tools.ietf.org/html/draft-rja-ilnp-dns

[13] Ran Atkinson, "ICMP Locator Update message," Internet Draft, Work in Progress,
http://tools.ietf.org/html/draft-rja-ilnp-icmp

[14] Ran Atkinson, "ILNP Concept of Operations," Internet Draft, Work in Progress,
http://tools.ietf.org/html/draft-rja-ilnp-intro

[15] Ran Atkinson, "ILNP Nonce Destination Option," Internet Draft, Work in Progress,
http://tools.ietf.org/html/draft-rja-ilnp-nonce

[16] Alan Ford, Costin Raiciu, Mark Handley, and Olivier Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," Internet Draft, Work in Progress,
http://tools.ietf.org/html/draft-ietf-mptcp-multiaddressed

[17] Randall Stewart, Ed., "Stream Control Transmission Protocol," RFC 4960, September 2007.

[18] Randall Stewart, Qiaobing Xie, Michael Tuexen, Shin Maruyama, and Masahiro Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration," RFC 5061, September 2007.

[19] Jon Postel, "User Datagram Protocol," RFC 768, August 1980.

[20] Jon Postel, "Transmission Control Protocol," RFC 793, September 1981.

[21] Fred Baker and Pekka Savola, "Ingress Filtering for Multihomed Networks," RFC 3704 [BCP 84], March 2004.

[22] David Meyer, "The Locator Identifier Separation Protocol (LISP)," *The Internet Protocol Journal,* Volume 11, No, 1, March 2008.

FRED BAKER, a Cisco Fellow, has been active in technology development and Internet standardization since the 1980s. He participated in early development of IEEE 802.1d switching and IP routing. In the IETF, he has written or edited RFCs on a variety of topics, and chaired both working groups and the IETF itself. At this time, he is the IETF's Voting Member on the U.S. NIST Smart Grid Interoperability Panel, a member of the SGIP's Architecture Committee, and co-chair of the IETF IPv6 Operations Working Group. At Cisco, his group supports research at universities; he is looked to for research advice and mentorship both within and outside the company. E-mail: **fred@cisco.com**

Reflecting on World IPv6 Day

by Phil Roberts, ISOC

On June 8, 2011, many websites around the world made their main webpage reachable over IPv6 for 24 hours, and many of those that did this left their sites IPv6-accessible afterward.

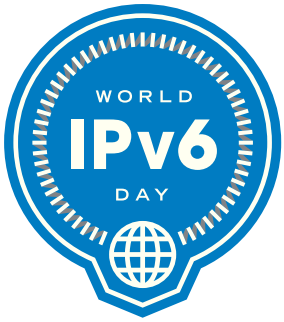
Major worldwide websites enabled IPv6 on their main page. Google enabled not only its main website but also YouTube and Blogger. Facebook and Yahoo! both enabled their main webpages as well. These websites are the five most visited websites in the world according to Alexa rankings. Other major worldwide websites that enabled IPv6 include Yahoo! Japan, Bing, Microsoft, BBC, CNN, and AOL.

Important local websites in countries around the world also joined in. In South Korea both Naver and Daum (the first and fourth most visited sites in South Korea according to Alexa) joined the event. In the Czech Republic four of the top 25 local websites joined. There were also major sites from Brazil, Portugal, and Indonesia.

Purposes

Enabling IPv6 in this way served numerous purposes:

- Network operators clearly saw that content is going to be available on IPv6. Although the major websites may not be quite there yet, it is clear that they are seriously moving in that direction.
- The industry worked to improve problems with IPv6 connectivity. Some immediate improvement resulted, and more fixes are underway to further improve IPv6 connectivity.
- Setting a public date created a deadline that accelerated deployment for many of the organizations that contacted us.
- It was important to be compared with Google, Facebook, and Yahoo!. Participants in this experiment wanted to be seen doing the same thing as the industry giants.
- This event was a clear example of how the Internet industry can work together to deploy technology that is for the good of the Internet, without intervention from outside entities. The multi-stakeholder model of Internet development continues to function well.



More than 1000 organizations contacted the Internet Society. Many of these organizations had already permanently enabled IPv6. Of the 430 or so websites the Internet Society monitored on the day, roughly two-thirds have continued to provide IPv6 access after the day.

In addition, major hosting companies enabled IPv6 for large numbers of domains, including Domain Factory, which, as a result of participating in World IPv6 Day, has made IPv6 "on by default" for all of its more than 800,000 domains. Another hosting company, Stratos, left IPv6 on after June 8 for its more than 4 million domains.

RIPE Labs did extensive measurements of IPv6 leading up to, on, and after the day, and it has published results indicating an increase in IPv6 traffic on the day—and an overall increase in IPv6 traffic also after the day.

References

[1] Phil Roberts, "World IPv6 Day," *The Internet Protocol Journal,* Volume 14, No. 1, March 2011.

[2] RIPE Labs, "Measuring World IPv6 Day—Long-Term Effects," **http://labs.ripe.net/Members/emileaben/measuring-world-ipv6-day-long-term-effects**

[3] RIPE Labs, "Measuring World IPv6 Day—Some Glitches And Lessons Learned," **http://labs.ripe.net/Members/emileaben/measuring-world-ipv6-day-glitches-and-lessons-learned**

[4] RIPE Labs, "Measuring World IPv6 Day—First Impressions," **http://labs.ripe.net/Members/mirjam/measuring-world-ipv6-day-first-impressions**

PHIL ROBERTS joined the Internet Society (ISOC) in 2008. Prior to that he spent several years with Motorola in research and product development, all in the area of mobile broadband systems. He has been active in the IETF for more than a decade. He can be reached at: **roberts@isoc.org**

Letters to the Editor

Hi Geoff,

Thanks you for your contribution to the March 2011 issue of *The Internet Protocol Journal*. Your description in "A Rough Guide to Address Exhaustion" and the article on "Transitional Myths" were very insightful into the whole issue of IPv4 to IPv6, and the issues concerning migration. Some of your thoughts on the migration hit home, as I am speaking to customers about the planning for the transition and I see a lot of "Got You" that I must now incorporate in my discussions with my customer.

If you do have a means of updating the technical community with activities in the area of IPv6 and how to move customers to this protocol platform, can you please point me in that direction? I like your approach and so would like to stay close to what you are doing in this area. Again, thank you for your contribution!

Ole, thanks for getting this type of information out to the technical community. Great work.

—Joel Smith, Verizon Business, Toronto, Ontario, Canada
joel.smith@one.verison.com

The author responds:

Hi Joel,

Thank you for your comments.

Running IPv6 in a dual-stack configuration certainly presents some issues, some of which are unique to particular networks and configurations, some of which appear to be common to particular roles (such as content delivery platform, Internet Service Provider, Enterprise Provider, and end user), and some of which are common across most, if not all, circumstances.

In assisting to set up some dual-stack services a year ago, I wrote down some of the issues that I found helpful in an article: "Two Simple Hints for Dual Stack Servers" (**http://www.potaroo.net/ispcol/2010-05/v6hints.html**). You may find those hints to be of some value to your work. Some other sites that have a good collection of information are: **http://www.ipv6actnow.org/** and the community site **http://www.getipv6.info/index.php/Main_Page**, which also contains a wealth of information of a technical nature.

The basic guideline is to approach adding IPv6 to a network like any other engineering project: exercise care and attention to detail, and you will find it to be very straightforward!

Kind regards,

—Geoff Huston, APNIC
gih@apnic.net

Geoff and Ole,

Many thanks for your excellent papers in the March 2011 issue of IPJ. You have brought all the issues together in one place. They are clearly explained. Now I'll do my small part by suggesting to one and all that they read it. My IPv6 service comes from a manually configured tunnel from Hurricane Electric.

—Dan Cotts
dcotts@lisco.com

The author responds:

Thanks, Dan, for this feedback. It's certainly the right time for both users and content providers to act now to ensure that we continue to enjoy an Internet that still operates with a coherent end-to-end architecture into the future. The only way we can ensure that this happens is to act now and insist on IPv6—everywhere!

—Geoff Huston, APNIC
gih@apnic.net

Hello,

I enjoyed the recent IPv6 issue (Volume 14, No. 1, March 2011), but was dismayed by the lack of any frank discussion of the IPv6 "any-to-any" mantra versus the benefits of IPv4 *Network Address Translation* (NAT).

Internet purists don't hide their desire to rid the world of NAT and return to an any-to-any Internet where they could use FTP to/from any host. But for the past 15 years, NAT, RFC 1918, and perimeter security have been great for the Internet and for home and enterprise networking. When dealing with billions of endpoints, the implicit security of NAT far outweighs any alternative. Just think back to the pre-broadband/NAT days when hosts were attacked within seconds of dialing into an ISP.

Of the ~1.7 billion publicly addressed Internet devices, the vast majority would be perfectly happy behind *Carrier-Grade NAT* (CGN). In fact, as ISPs begin introducing NAT offerings, millions will stampede to them for their lower cost. Mobile phone networks are the lowest-hanging fruit, followed by residential broadband. ISPs will still offer public IP products, of course, just at a higher price point.

The IETF needs to stop pussy-footing around the issue. CGN is not just an IPv6 transitional technology; it could very well become the de facto operating standard for the next decade.

The IETF desperately needs to:

- Amend RFC 5382 ("NAT Behavioral Requirements for TCP") to allow endpoint-independent mapping. This will improve CGN scalability by several orders of magnitude. For example, rather than 2000 hosts per public IP mentioned in Mr. Huston's "Rough Guide" on address sharing, CGN could support 200,000 or more hosts per public IP.
- Develop an IETF standard for P2P connection establishment. It took 8+ years for the IETF to take an interest in P2P mechanics (RFC 5128). Now it's time to show leadership. If a CGN-compatible P2P establishment standard were drafted, it would be adopted by P2P libraries overnight. While they're at it, look at standards for tying *Universal Plug and Play* (uPnP) into CGN.
- Help coordinate a discussion of operational issues with ISP administration, law enforcement, DMCA enforcement, geolocation services, black/white lists, etc. Perhaps it's time to extol the benefits of millisecond-accurate IPFIX logs with NAT extensions, or develop a new TCP option to embed NAT details?
- Legitimize common ISP self-preservation tactics, such as restricting SMTP, metering connections/sec, and so on.

Most importantly, IPv6 proponents should stop taking CGN as a personal affront. There is no malice; it's simply the path of least resistance for the IPv4 conundrum.

—*Craig Weinhold, Madison, Wisconsin*
craig.weinhold@cdw.com

The author responds:

Thank you for your note, Craig.

The discussion of how far the Internet could scale with integration of NATs into the interior of the network as well as the current pattern of NATs at the edge is not a new discussion. The *Realm Specific IP* (RSIP) Working Group was active over a decade ago in the IETF, looking at how a network would operate that consisted of a union of distinct realms, each of which was, in address terms, a discretely addressed IP network. With the benefit of hindsight, the outcomes of that effort in supporting a case for infrastructure NATs as a long-term architectural direction for the Internet were not overly encouraging.

From the perspective of the technology community, it reinforced the conclusion that IPv6 represented the best possible response to the recognized problem of IPv4 address exhaustion. NATs were a poor compromise in so far as, at the most basic level, NATs add state into the interior of the network. This imposition of state into the network infrastructure imposes a cost in terms of service fragility and network robustness that cannot be avoided.

There was an assumption some years ago that the industry would grapple with the transition to IPv6 well before the exhaustion of IPv4 addresses, and we would never have to deal with a dual-stack transition where one-half of the dual stack, the IPv4 part, would need to operate in a mode that included infrastructure NATs. We now appear to be beyond choice here—for the Internet to continue to grow by a further 300 million new services per year at present, and grow by yet more in the coming years, there is no choice but to operate the IPv4 part of the dual-stack environment with infrastructure NATs.

But this is a short-term hack, as distinct from a tenable longer-term position. The address pool of IPv4 is not getting any larger, and as more and more new services are added into a dual-stack network, the growth in the IPv4 part of the network can be absorbed only by progressive reduction of the number of available ports to each client of the infrastructure NAT. Services become more fragile and the network becomes less resilient. The inevitable next step in progressive scarcity of IPv4 addresses in the face of such inexorable growth is to drop the entire notion of end-to-end service and introduce application-level proxies into the IPv4 network. At this point we lose any ability to further sustain an open IPv4 Internet. The only applications that could be supported are those that are supported by the application-level proxies, and all other applications simply fail. The segregation of one Internet into a number of effectively disconnected "walled gardens" of networking is a rapid outcome in such a scenario.

One of the strengths of the Internet is its openness and neutrality. The open architectural model allows novel services to be added into the network by simply equipping clients and services with the service, leaving the interior of the network untouched. The interior of the network is entirely neutral to such innovations, as it is unaware of the content or intent of the packets that are passed through its switching infrastructure.

So the long-term path of greatest common benefit to all in the Internet is a network that, as far as possible, simply vanishes! It is an Internet where content and services can rendezvous with users without having to negotiate with any network elements. It is a network that is free of toll gates. And the network has now grown to such an extent that the only path from here that can sustain that architectural simplicity and sustain yet more growth is one that shifts determinedly and rapidly to IPv6. With the limited time and resources available, attempting to improve upon NATs is, in my opinion, not the best use of the resources we can apply to this problem.

Regards,

—Geoff Huston, APNIC
gih@apnic.net

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

Fragments

RFC Series Editor Search Announcement

The *Internet Engineering Task Force* (IETF) is seeking an *RFC Series Editor* (RSE). The RSE has overall responsibility for the quality, continuity, and evolution of the *Request for Comments* (RFC)[3] Series, the Internet's seminal technical standards and publications series. The position has operational and policy development responsibilities. The overall leadership and supervision of RFC Editor function is the responsibility of the RFC Series Editor. The RSE is a senior professional who must be skilled in leading, managing and enhancing a critical, multi-vendor, global information service. The following qualifications are desired:

- Leadership and management experience. In particular, demonstrated experience in strategic planning and the management of entire operations. Experience that can be applied to fulfill the tasks and responsibilities described in "RFC Editor Model (version 2)"[1].
- Excellent written and verbal communication skills in English and technical terminology related to the Internet a must; additional languages a plus.
- Experience with editorial processes.
- Familiar with a wide range of Internet technologies.
- An ability to develop a solid understanding of the IETF, its culture and RFC process.
- Ability to work independently, via e-mail and teleconf, with strong time management skills.
- Willingness and ability to travel as required.
- Capable of effectively functioning in a multi-actor and matrixed environment with divided authority and responsibility; ability to work with clarity and flexibility with different constituencies.
- Experience as an RFC author desired.

More information about the position can be found on the RFC Editor Webpage[2]. The RSE reports to the *RFC Series Oversight Committee* (RSOC). Expressions of interest in the position, Curriculum Vitae (including employment history), compensation requirements, and references should be sent to the RSOC search committee at **`rse-search@iab.org`**. Questions are to be addressed to the same e-mail address. Applications will be kept confidential. The RSOC will interview interested parties at the IETF meeting in Quebec City that begins July 24, 2011, but the application period is open until the position is filled.

—Fred Baker, Chair, RFC Series Oversight Committee

References

[1] **http://www.ietf.org/id/draft-iab-rfc-editor-model-v2-02.txt**

[2] **http://www.rfc-editor.org/rse/RSE-position.html**

[3] Leslie Daigle, "RFC Editor in Transition: Past, Present, and Future," *The Internet Protocol Journal,* Volume 13, No. 1, March 2010.

Global IPv6 Deployment Monitoring Survey 2011

The *Global IPv6 Deployment Monitoring Survey 2011* is now online at: **http://www.surveymonkey.com/s/GlobalIPv6survey2011**

This survey has been designed by GNKS Consult in collaboration with TNO and the RIPE NCC to further understand where the community stands on IPv6 and what needs be done to ensure that the Internet community is ready for the widespread adoption of IPv6.

Anyone can participate in this survey and we hope that the results will establish a comprehensive view of current IPv6 penetration and future plans for IPv6 deployment. The survey comprises 23 questions and can be completed in about 15 minutes. For those without IPv6 allocations or assignments or who have not yet deployed IPv6, there will be fewer questions.

The survey closes July 31, 2011. We thank you for your time and interest in completing this survey. If you have any questions concerning the survey, please e-mail: **info@gnksconsult.com**

For more information about the survey and links to previous year's survey results, please see:

https://www.ripe.net/internet-coordination/news/industry-developments/global-ipv6-deployment-monitoring-survey-2011

RFC 6127 Published

The topic of IPv4 depletion and IPv6 deployment is covered in the recently published RFC 6127 entitled "IPv4 Run-Out and IPv4-IPv6 Co-Existence." From the introduction: "When IPv6 was designed, it was expected that the transition from IPv4 to IPv6 would occur more smoothly and expeditiously than experience has revealed. The growth of the IPv4 Internet and predicted depletion of the free pool of IPv4 address blocks on a foreseeable horizon has highlighted an urgent need to revisit IPv6 deployment models. This document provides an overview of deployment scenarios with the goal of helping to understand what types of additional tools the industry needs to assist in IPv4 and IPv6 co-existence and transition." RFCs can be obtained from the RFC Editor web page, see:

http://www.rfc-editor.org/rfc.html

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

September 2011 Volume 14, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

I recently attended a conference in Japan where the attendee network offered IPv6 service only. In the past, conferences such as the *Asia Pacific Regional Conference on Operational Technologies* (APRICOT) and meetings of the *Internet Engineering Task Force* (IETF) have conducted IPv6 experiments, but these have all been "opt-in" events. The conference in Japan was different: there was no IPv4 service available. Making this work involved a few manual configuration steps, but for the most part everything worked more or less the same as it did under IPv4. Some applications, including my instant message client and Skype did not work, and all connections to IPv4-only hosts needed to use *Fully Qualified Domain Names* (FQDNs) instead of IP addresses, but overall the experience gave me confidence that IPv6 is becoming a reality. As you might expect, this IPv6-only experiment also uncovered a number of bugs and incompatibilities that were duly reported to developers around the world.

Our first article is an overview of *TRansparent Interconnection of Lots of Links* (TRILL). TRILL uses Layer 3 routing techniques to create a large cloud of links that appear to IP nodes to be a single IP subnet. The protocol has been developed in the IETF and is currently being refined and enhanced in the TRILL working group. The article is by Radia Perlman and Donald Eastlake.

Developments in Internet technologies have lead to changes that go beyond the Internet itself. Not only is *Voice over IP* (VoIP) often used in place of traditional circuit-switched telephony, the telecommunication networks themselves are evolving to incorporate IP routers in place of traditional telephone switches. This evolution also applies to cellular telephone networks, specifically to what is known as *backhaul*—the transportation of voice and data from the cell sites to the mobile operators' core networks. Jeff Loughridge explains more in "The Case for IP Backhaul."

Once again I would like to remind you about the IPJ subscription renewal campaign. Each subscriber to this journal is issued a unique subscription ID that, coupled with an e-mail address, gives access to the subscription database by means of a "magic URL." If your subscription has expired or you have lost your subscription ID, changed e-mail, postal mail, or delivery preference, just send an e-mail to ipj@cisco.com with the updated information and we will take care of the rest.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Introduction to TRILL

by Radia Perlman, Intel Labs, and Donald Eastlake, Huawei Technologies

Transparent Interconnection of Lots of Links (TRILL)[1] is an *Internet Engineering Task Force* (IETF) protocol standard that uses Layer 3 routing techniques to create a large cloud of links that appear to IP nodes to be a single IP subnet. It allows a fairly large Layer 2 cloud to be created, with a flat address space, so that nodes can move within the cloud without changing their IP addresses, while using all the Layer 3 routing techniques that have evolved over the years, including shortest paths and multipathing. An early problem and applicability statement for TRILL can be found in [6]. Additionally, TRILL supports Layer 2 features such as V*irtual Local-Area Networks* (VLANs), the ability to autoconfigure (while allowing manual configuration if so desired), and multicast/broadcast with no additional protocol.

Additionally, TRILL is evolutionary in the sense that an existing Ethernet deployment, where the links are connected with bridges, can be converted into a TRILL cloud by replacing any subset of the bridges with devices implementing TRILL. Devices implementing TRILL are called *Routing Bridges,* or *RBridges*. As bridges are replaced, nothing changes for the IP nodes connected to the cloud except that the cloud becomes more stable and uses available bandwidth more effectively.

To understand why TRILL was needed, it is helpful to explore the history of Ethernet and IP.

Network protocols are usually described in terms of *layers*. The description usually quoted in textbooks is the *Open Systems Interconnection* (OSI) *Reference Model,* which describes seven protocol layers[4]. It is important to realize that the layers are useful primarily as a way to think about networking, but actual network protocols are far more complex. Layers get subdivided or combined, and often a technology usually thought of as belonging to a lower layer (for example, Layer 2) can be layered on top of a higher layer (for example, Layer 3). Most descriptions of network layers agree on the bottom four layers, and vary according to details such as whether syntax (for example, *Extensible Markup Language* [XML][7]), which would be a *Presentation Layer* in the OSI model, is a layer or not. Such descriptive choices do not affect how protocols are built, and luckily, for understanding of TRILL, the relevant layers to focus on are just the bottom three:

- Layer 1, *Physical Layer:* Physical, electrical, and optical specification for connectors, bit signaling, etc.
- Layer 2, *Data Link Layer:* The protocol that lets neighbor nodes on a link exchange packets
- Layer 3, *Network Layer:* The protocol that provides routing to create a path from a source node to a destination node

TRILL, as we will see, is a Layer 2 and ½ protocol: It glues links together so that IP nodes see the cloud as a single link. Therefore, TRILL is below Layer 3; but, it is above Layer 2 because it terminates traditional Ethernet clouds, just like IP routers would do.

It is definitely time to be confused. Why are there multiple links at Layer 2? Isn't that the job of Layer 3?

Evolution of Layer 2 from Point-to-Point Links to LANs

In the beginning (the 1970s or so for the purposes of this article), Layer 2 really was a direct link between neighbor nodes. Most links were point-to-point, and Layer 2 protocols primarily created *framing*—a way to signal the beginning and end of packets within the bit stream provided by Layer 1—and *checksums* on packets[11]. For links with high error rates, Layer 2 protocols such as *High-Level Data Link Control* (HDLC)[12] provided message numbering, acknowledgements, and retransmissions, so the Layer 2 protocol resembled, in some ways, a reliable protocol such as TCP. HDLC and other Layer 2 technologies sometimes provided an ability to have multiple nodes share a link in a master/slave manner, with one node controlling which node transmits through techniques such as polling.

Then the concept of *Local-Area Networks* (LANs) evolved, the most notable example being Ethernet. Ethernet technology enabled interconnection of (typically) hundreds of nodes on a single link in a peer-to-peer rather than master/slave relationship. Ethernet was based on *CSMA/CD,* where CS = *Carrier Sense* (listen before talking so you don't interrupt); MA = *Multiple Access;* and CD = *Collision Detect* (listen while you are talking to see if someone starts talking while you are so you are both interfering with each other). Interestingly, although IP had a 4-byte address and was the basis of addressing for the entire Internet, Ethernet had a larger 6-byte address, with aspirations for connecting only a small number of nodes in a fairly small region such as a single building.

The reason for the larger address space for Ethernet was to avoid the need to configure addresses when plugging nodes into a network. Instead, manufacturers of equipment would purchase blocks of Ethernet addresses and embed a unique address for each device in their hardware (the "MAC address"), and an Ethernet node would then be able to use that address in any Ethernet without fear of address collision.

Evolution of Ethernet to Spanning Tree

LANs came onto the scene with such fanfare that people came to believe that LAN technology was a replacement of traditional Layer 3 protocols such as IP. People built applications that were implemented directly on Layer 2 and had no Layer 3. This situation meant that the application would be limited by the artifacts of the Layer 2 technology, because a Layer 3 router cannot forward packets that do not contain the Layer 3 header implemented by the router.

In the case of the original Ethernet, it meant the application would work only within a maximum distance of perhaps a kilometer.

When people using technologies built directly on a LAN realized they wanted networks larger (in distance and total number of nodes) than the LAN technology allowed, the industry invented the concept of "bridges"—packet-forwarding devices that forwarded Layer 2 packets.

Forwarding Ethernet packets might seem easy because the Ethernet header looks similar to a Layer 3 header. It has a source and destination address, and the addresses are actually larger than IP addresses. But Ethernet was not designed to be forwarded. Most notably absent from the Ethernet header is a *hop count* (also sometimes referred to as a "time to live," or TTL) to detect and discard looping packets. But other features of a typical Layer 3 protocol were also missing in Ethernet, such as an address that reflects where a node is in the topology, node discovery protocols, and routing algorithms. These features were not in Ethernet because the intention of the Ethernet design was that it be a Layer 2 protocol, confined to operation on a single link.

The transparent bridge was invented as a mechanism to forward Ethernet packets emitted by end nodes that did not implement Layer 3. Ethernet at the time had a hard packet size limit, so bridges could not modify the packet in any way.

The transparent bridge design, which met those constraints, consisted of having bridges listen promiscuously, remember the source addresses seen on each port, and forward based on the learned location of the destination address. If the destination was unknown, the packet would be forwarded onto all ports except the one that it was received on.

This simple method worked only if there was only one path between any pair of nodes. So the concept was enhanced with a protocol known as the *Spanning Tree Algorithm*.[8] The physical topology could be an arbitrary mesh, but bridges, using the spanning-tree algorithm, would prune the topology into a loop-free (tree) topology on which data packets were forwarded. ("Spanning" means that packets can reach all the nodes.)

As Figure 1 shows, the spanning-tree concept is that an arbitrary topology could be built using Ethernet links (horizontal lines) and bridges (circles). Bridges running the spanning-tree algorithm determine a loop-free subset of the topology, and put some ports into standby (the ones that are shown in Figure 2 as dotted lines). Data packets flow on the ports that spanning tree determines should be active. This model does not yield optimal routes, as indicated in Figure 3, where packets between A and X go through the path of bridges 11, 7, 6, 2, 14, 4, and 3.

Figure 1: A Bridged Network

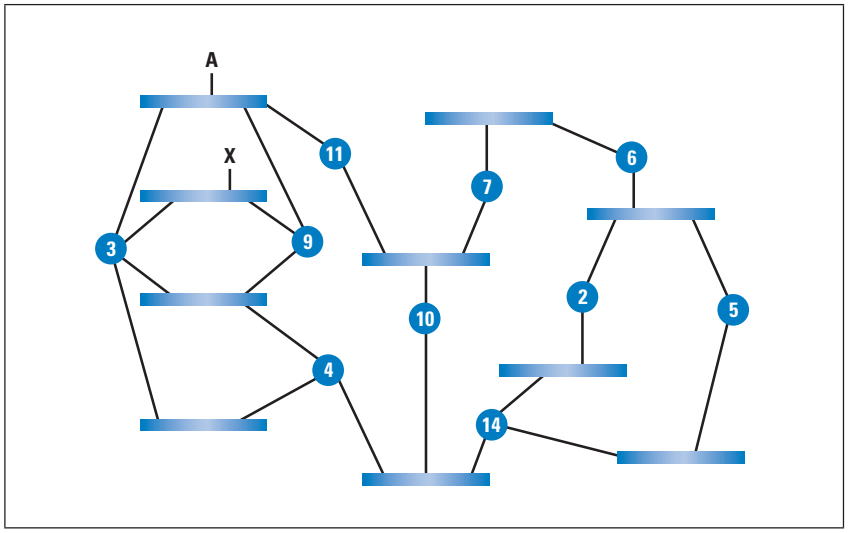


Figure 2: Bridged Network with Spanning Tree

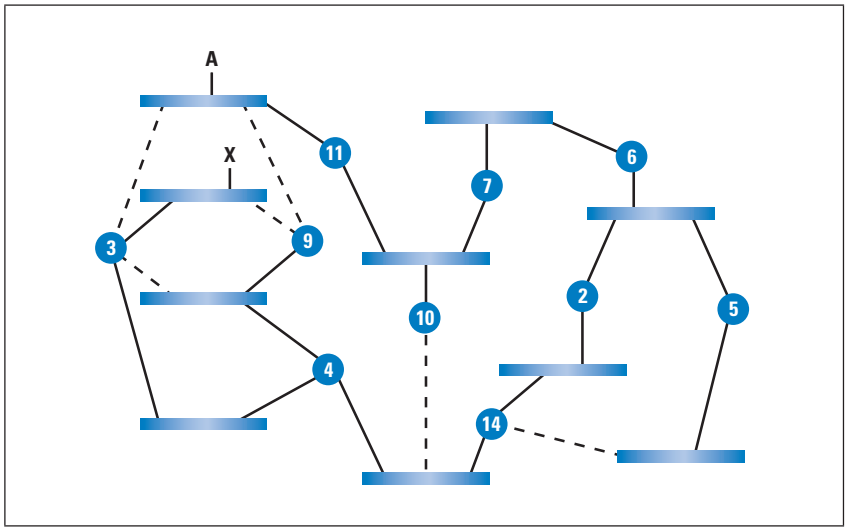
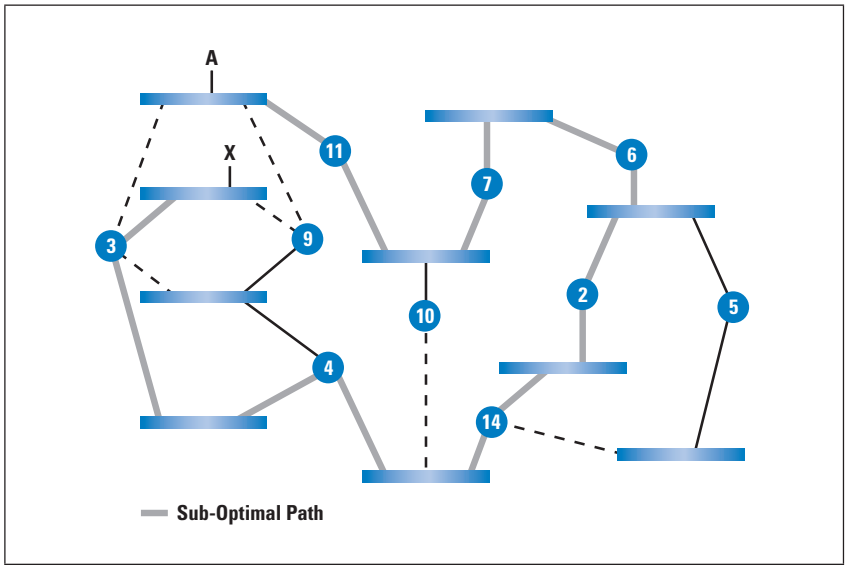


Figure 3: A Sub-Optimal Path



The spanning-tree algorithm is also inherently unstable. It requires bridges to be engineered to be able to examine every incoming packet at wire speed, to determine if the packet is a spanning-tree message, and if so, process it. The spanning-tree algorithm requires a bridge to forward unless there is a "more qualified" neighbor bridge on the link. Details of the spanning-tree algorithm, fascinating as they are, are beyond the scope of this article. If a bridge loses enough spanning-tree messages from its "more qualified" neighbor bridge because congestion overwhelms its ability to process incoming messages, the bridge will conclude that it does not have a more qualified neighbor, and therefore should start forwarding onto the link. This situation is extremely dangerous without a hop count, a field that would naturally be included in a protocol designed to be Layer 3 and forwardable.

The originally invented Ethernet, CSMA/CD, is pretty much nonexistent. Almost all Ethernet today consists of bridges connected with point-to-point links. The header still looks like Ethernet, but new fields have been added, such as VLANs discussed later in this article.

Characteristics of IP

Transparent bridging was necessitated by a quirk of history, in that applications were being built without Layer 3. But today, applications are almost universally built on top of IP. So why not replace all bridges with IP routers?

The reason is an idiosyncrasy of IP. In IP, routing is directed to a *link,* not a *node.* Each link has its own block of addresses. A node connected to multiple links will have multiple IP addresses, and if the node moves from one link to another, it must acquire a new IP address within the block for that link.

This property is not an inherent property of Layer 3, just a characteristic of IP. An alternative technology, proposed in 1992 as a replacement to IPv4, was *Connectionless-mode Network Protocol* (CLNP), an ISO packet format that had 20-byte addresses (actually, variable length). Its address, like IP, was hierarchical, routing to the longest matching address prefix in the forwarding table that matched the destination address. But in IP, the bottom level of routing was to a single link. In CLNP, the bottom level of routing consisted of routing to a cloud known as an "area," that included lots of links (typically hundreds). Within the area, end nodes announced themselves and routers routed directly to the end node. An end node could move within an area without changing its Layer 3 address. Routers within an area would not need to be configured.

In contrast, with IP, a block of IP addresses needs to be carved up to assign a unique block to each link, IP routers need to be configured with the address block for each of their ports, and nodes that move from one link to another have to change their Layer 3 addresses. Therefore, it is still popular to create large bridged Ethernets, because a bridged set of links looks to IP like a single link.

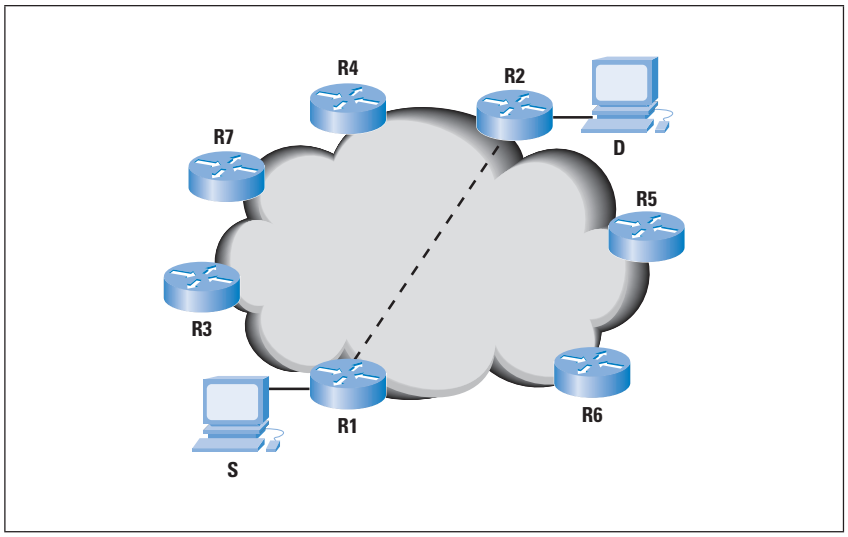
TRILL: Best of Both Worlds

TRILL allows the ease of configuration of Ethernet while benefitting from the routing techniques provided at Layer 3. It also coexists with existing bridges; it is not necessary to replace all the bridges in an Ethernet, but the more bridges replaced by RBridges, the better the bandwidth usage and the more stable the cloud becomes (because the spanning trees get smaller and smaller, and ultimately disappear if all bridges are replaced by RBridges).

Figure 4 shows the basic concepts in TRILL handling a unicast packet where the location of the destination is known:

- RBridges run a link state routing protocol, which gives each of them knowledge of the topology consisting of all the RBridges and all the links between RBridges. Using this protocol, each RBridge calculates shortest paths from itself to each other RBridge, as well as trees for delivering multidestination traffic.
- When an RBridge, R1, receives an Ethernet frame from an end node S, addressed to Ethernet destination D, R1 encapsulates the frame in a TRILL header, addressing the packet to the RBridge R2, to which D is attached. The TRILL header contains an "ingress RBridge" field (R1), an "egress RBridge" field (R2), and a hop count.
- When R2 receives the encapsulated packet, R2 removes the TRILL header and forwards the Ethernet packet on to D.

Figure 4: RBridging



What the TRILL header looks like, how R1 knows that R2 is the correct "egress RBridge," and some of the concepts in the link state protocol *Intermediate System-to-Intermediate System* (IS-IS) are described in the next section. We also explain how TRILL handles multidestination frames, VLANs, and IP Multicast.

The TRILL Header

The main fields in the TRILL header are: ingress RBridge nickname (16 bits), egress RBridge nickname (16 bits), hop count (6 bits), and a multidestination flag bit (1 bit). A typical Layer 3 header would contain a source, a destination, and a hop count. So TRILL is basically an encapsulation header with flat 16-bit addresses. How RBridges obtain "nicknames" is described later in this article.

This header is very simple for core RBridges to forward, compared with either an IP or an Ethernet header. The destination field is just 16 bits, so it can be a simple table lookup to find the entry in the output port, as opposed to the Ethernet 6-byte destination, which typically requires content-addressable memory or hashing, or the longest prefix matching of IP.

Learning End-Node Locations

How does R1 know that R2 is the correct egress RBridge for some destination D? The default mechanism is learning the correspondence between (ingress RBridge, source MAC address) when the egress RBridge decapsulates a packet. If R1 does not know where the destination MAC is located, R1 encapsulates the packet in a TRILL header with the multidestination flag set, indicating that it should be transmitted through a tree to all the RBridges.

An additional mechanism, which is optional, is known as *End-Station Address Distribution Information* (ESADI). ESADI allows R1 to announce some or all of the end nodes that are attached to R1. Both announcing to and listening to ESADI are optional. This mechanism has advantages over flooding and learning from data packets:

- ESADI packets can have cryptographic protection.
- R1 might have a more definite reason to know that S is attached to R1 than simply seeing a packet with the S address in the header. For instance, R1 might have been configured to lock down a port to the S MAC address. Or there might be a cryptographically protected enrollment protocol when S attaches to R1.
- R1 might be able to have tighter timers on verifying the location of local end nodes; for instance, if they are IP nodes, R1 might be able to ping them.

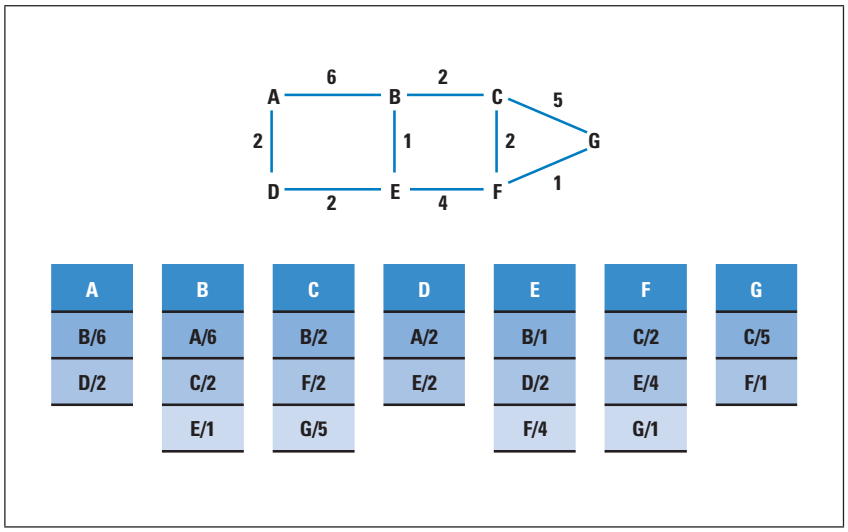
It is also possible to have a directory that lists not only (RBridge nickname, {set of attached end-node MAC addresses}) but also {(end-node IP address, end-node MAC address)} pairs. The first RBridge, or a *hypervisor,* or the end-node process itself, might query the directory about the destination, and encapsulate packets, rather than flooding, and thus also be able to bypass the IPv4 *Address Resolution Protocol* (ARP) and the *IPv6 Neighbor Discovery* (ND) protocols.

Link State Protocols

A *link state* protocol is a routing protocol in which each router R determines who its neighbors are, and broadcasts (to the other routers) a packet, known as a *Link State Packet* (LSP), that consists of information such as "I am R," and "My neighbor routers are X (with a link cost of c1), Y (cost c2), and Z (cost c3)." The commonly deployed link state protocols are *Intermediate System-to-Intermediate System* (IS-IS)[2][9] and *Open Shortest Path First* (OSPF)[10]. IS-IS, designed in the 1980s to route DECnet, was adopted by the *International Organization for Standardization* (ISO). IS-IS can route IP traffic and is used by many *Internet Service Providers* (ISPs) to route IP. IS-IS was a natural choice for TRILL because its encoding easily allows additional fields, and IS-IS runs directly on Layer 2, so that it can autoconfigure, whereas OSPF runs on top of IP and requires all the routers to have IP addresses.

Figure 5 shows a small network (at the top), consisting of 7 routers. In the bottom half of the figure, the LSP database is shown; all the routers have the same LSP database because they all receive and store the most recently generated LSP from each other router. The LSP database gives all the information necessary to compute paths. It also gives enough information for all the routers to calculate the same tree, without needing a separate spanning-tree algorithm. As we will see, TRILL requires a tree (at least one tree) for distribution of multidestination packets.

Figure 5: Router Network and Link State



Acquiring Nicknames

Given that the most recently generated link state packet of each RBridge is broadcast to, and stored by, each other RBridge, it is possible to spread other information through the link state packets, such as a protocol for acquiring a unique nickname. Each RBridge chooses a nickname at random, avoiding nicknames already acquired by other RBridges (as discovered by examining the LSP database).

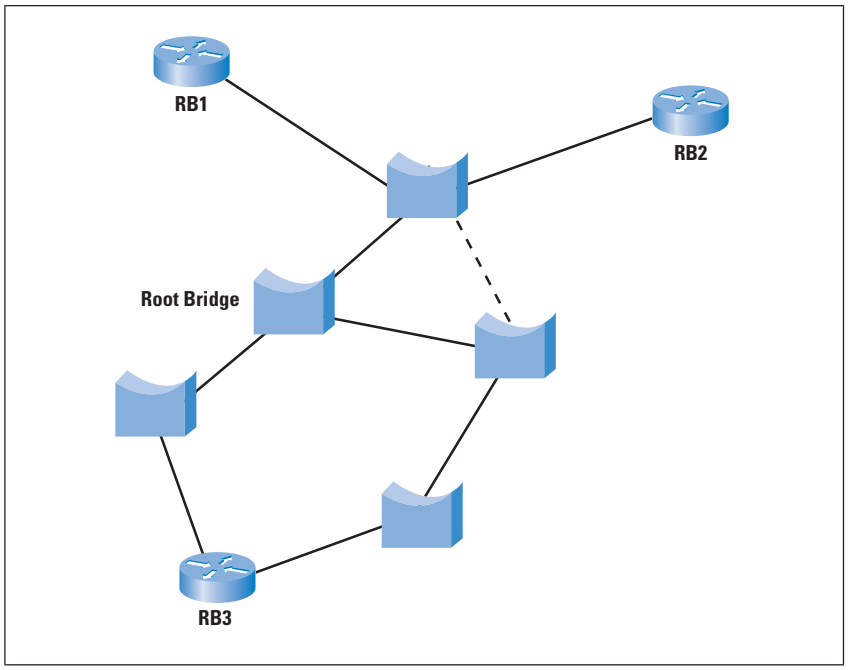
If two RBridges choose the same nickname, there is a tie-breaker, based on configured priority and 6-byte system ID. One of the RBridges gets to keep the nickname and the other RBridge has to choose another nickname that appears not to be in use.

It is possible to configure RBridges with nicknames, in which case a configured nickname takes priority over one that was randomly chosen. And in the case of misconfiguration, where two RBridges have been configured with the same nickname, again, ID and priority choose a winner, and the other one has to choose a different nickname.

Mixing RBridges with Bridges

TRILL is designed so that any subset of bridges in an Ethernet can be replaced by RBridges. A set of links connected by bridges will be perceived by RBridges as a single shared link connecting the RBridges on that link. The bridges inside that link will behave as ordinary bridges, forming a spanning tree and forwarding packets along that tree. Figure 6 illustrates an Ethernet connected by several bridges, with one port (indicated by the dashed line) selected by the spanning tree as being in backup.

Figure 6: RBridges Connected by Bridged LAN



The RBridges RB1, RB2, and RB3 perceive the link as in Figure 7, a single shared link, on which RB3 has two ports.

Introducing RBridges into a bridged Ethernet partitions the spanning trees into smaller spanning trees. RBridges operate on a topology consisting of the RBridges themselves, connected with "links" that are either bridged Ethernets or point-to-point links.

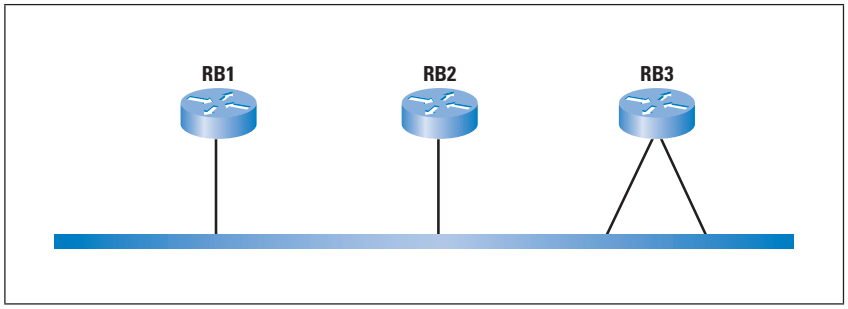


Figure 7: Figure 6 as Perceived by RBridges: a Single Shared Link Where RB3 Has 2 Ports onto the Same Link

Link Types and the Hop-by-hop Header

In addition to the TRILL header, when RBridge R1 is forwarding a TRILL-encapsulated frame to neighbor RBridge R2, there is an additional header that is specific to the type of link connecting R1 and R2. Although TRILL carries Ethernet inside, a link between two or more RBridges could be an arbitrary type of link; for example, besides Ethernet, it could be a *Point-to-Point Protocol* (PPP) link[13], an IP or *IP Security* (IPsec) tunnel, *Multiprotocol Label Switching* (MPLS) path, etc.

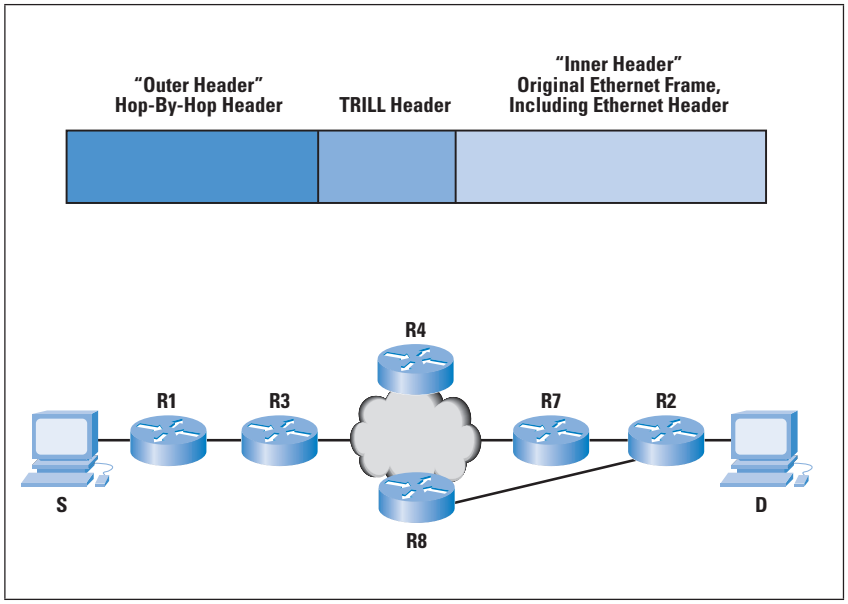
If the link is an Ethernet link, the "outer" header is an Ethernet header. If it is a PPP link, the outer header is a PPP header. The outer Ethernet header (on an Ethernet link) serves two purposes:

- If there are bridges on the link, they will perceive the packet as a normal Ethernet packet, and forward it through the spanning tree. The learning tables of the bridges on the link will see only the addresses of the RBridges on that link.
- It allows R1, when forwarding onto a link with multiple neighbors (say R2 and R3), to specify which of R2 or R3 is chosen by R1 to forward the packet by unicasting the packet to the chosen next-hop RBridge. For example, it could be that both R2 and R3 are equal costs to the destination, so R1 would need to specify which of them should forward the packet. Otherwise, both might forward the packet, and the packet would be duplicated.

So, as illustrated in Figure 8, a TRILL-encapsulated packet might have three headers:

- The outer header, or hop-by-hop header, which is stripped off at each hop, is specific to the type of link connecting neighbor RBridges, and, when forwarded between R1 and R2, it specifies R1 as source and R2 as destination
- The TRILL header, which similarly to a Layer 3 header remains in place as the packet travels from the first RBridge to the last RBridge, specifying the first RBridge (the one that encapsulated the packet with a TRILL header) as the ingress RBridge, and the last RBridge (the one that will decapsulate the packet) as the egress RBridge
- The inner Ethernet header, which specifies the communicating end-node pair as source and destination

Figure 8: TRILL Packet Headers



Again referring to Figure 8, assume S transmits an Ethernet packet to D. In the inner Ethernet header, Source = S, Destination = D.

R1 encapsulates it with a TRILL header, where ingress RBridge = R1 and egress RBridge = R2. R1 forwards it to R3, putting on a link header appropriate to the link. If the link is an Ethernet link, the outer Ethernet header will indicate S = R1, D = R3. When R3 forwards to R7, R3 leaves the TRILL header as is (other than decrementing the hop count), strips the outer header, and puts in a new outer header indicating S = R3, D = R7. Likewise, R7 forwards to R2. If it is a PPP link, there is no source or destination. When R2 forwards to D, R2 strips off the TRILL header and D sees the Ethernet packet exactly as transmitted by S.

VLANs

Ethernet has a concept known as a *Virtual LAN* (VLAN), which partitions communities of end nodes sharing the same infrastructure (links and bridges), such that end nodes in the same set can talk directly to each other (using Ethernet), whereas those in different VLANs have to communicate through a router. IP nodes, although generally unaware of Ethernet VLAN tags, perceive different VLANs to be different IP subnets.

Typically, a bridge is configured with a VLAN for each port, and the bridge adds a tag to the Ethernet header that indicates which VLAN the packet belongs to. A bridge with a port that is configured to be VLAN x will deliver only packets tagged as VLAN x to that port, and will usually strip the VLAN tag before forwarding.

The original Ethernet did not have a VLAN concept. In today's Ethernet standard, each packet must be associated with a VLAN. A bridge might be configured with a default VLAN for a port, meaning that if no VLAN tag is in the packet, the bridge will treat it as if it is that default VLAN. A bridge B might be configured in various ways that make VLANs more complex:

- B might be configured to drop a set of VLANs rather than forward them onto a particular port, even though the port is a transit port.
- B might be configured to modify the VLAN tag to a different value when forwarding from one port to another.
- B might be configured to remove the VLAN tag when forwarding onto a particular port.

Appointed Forwarders

If there are multiple RBridges on the same link, together with end nodes, it is important that only one of them encapsulate a packet from an end node. As illustrated in Figure 9, if both R1 and R2 were to encapsulate a unicast packet from S, two copies would be delivered to the destination. However, if S were to transmit a multidestination packet (such as a multicast, or an unknown destination), then the copy that R1 encapsulates would be forwarded through the campus, received by R2 (which likely would not know that the packet originated on its port to R1), and R2 would decapsulate it. Then R1 would see a native packet from S, exactly as the first copy, and again encapsulate it and send it into the campus.

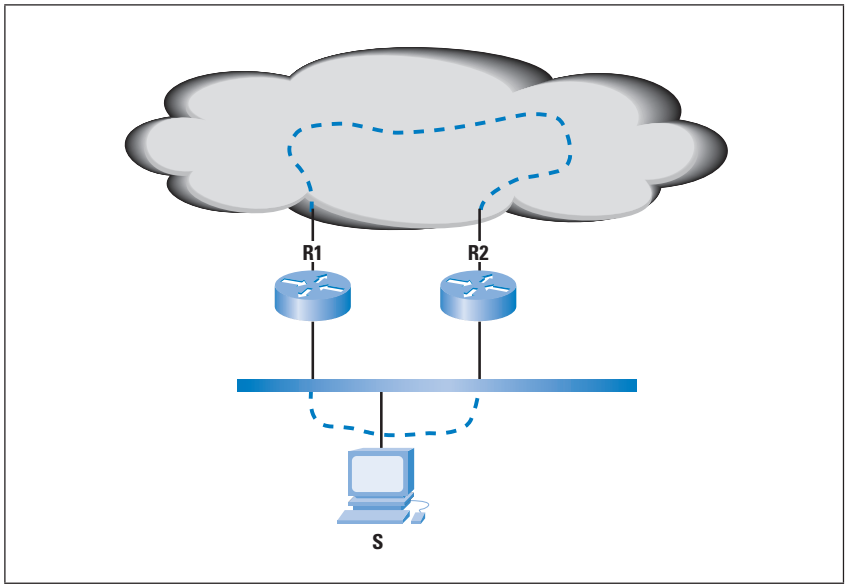


Figure 9: Link with Multiple RBridges. Note: No Hop Count Protection on Native Frame.

The hop count in the TRILL header would not solve this loop, because the hop count does not exist while the packet is not encapsulated with a TRILL header.

IS-IS has an election protocol in which one of the RBridges is elected as the *Designated RBridge* (DRB). In order to allow load-splitting the task of encapsulating and decapsulating traffic, the DRB may delegate the job of encapsulation/decapsulation based on VLAN. In other words, if R1 is DRB, R1 can delegate to R2 the task of encapsulating/decapsulating traffic for a set of VLANs, say VLANs x, y, and z, and delegate to R3 a different set of VLANs, and R1 might handle the rest.

Implications of VLANs on TRILL

TRILL treats VLANs strictly as a way of partitioning the end nodes, in contrast with IEEE, which allows bridges to drop transit traffic based on VLAN. Consequently, an Ethernet link connecting TRILL RBridges R1 and R2 might be able to deliver packets tagged with VLAN x, but not deliver packets tagged with VLAN y.

It is important, as shown in Figure 9, that all the RBridges on a link know about each other; otherwise they might both encapsulate a packet.

The IS-IS election is done through Hello messages, whereby RBridges announce themselves. Unfortunately, possible configuration of bridges, whether intentional or by mistake, can partition a link for traffic marked as VLAN y, but have the link be connected for traffic marked as VLAN x. This situation complicates the IS-IS election. When transmitting a Hello message onto an Ethernet link, an RBridge R1 must assign it to a VLAN. If R1 chooses VLAN y, its neighbor R2 might not see the Hello message. And then, unaware that there were multiple RBridges on the link, both R1 and R2 might encapsulate a VLAN x packet.

TRILL handles this situation by having the DRB (by default) transmit Hello messages on all the VLANs for which it is enabled on the port. The DRB chooses a VLAN, say VLAN A, for inter-RBridge communication on the link, and informs the other RBridges on the link that they should use VLAN A. The other RBridges transmit IS-IS messages (including Hello messages and LSPs) and encapsulated TRILL packets, putting VLAN A in the outer header. The VLAN tag in the inner header is the one that represents the community that the end node belongs to. The VLAN tag in the outer header is only for the purpose of traversing an Ethernet hop between RBridges.

Additionally, (by default), an RBridge that is Appointed Forwarder for a VLAN, transmits Hello messages on that VLAN.

If it is known that there are no bridges, the RBridges (including the DRB) can be configured to send Hello messages only on the single VLAN specified by the DRB.

Modified Hello Protocol

IS-IS has an election protocol in which routers (or RBridges in the case of TRILL) send Hello messages. Not only does the Hello message transmitted by R1 announce R1 to its neighbors, but the R1 Hello message contains a list of neighbors that R1 has heard Hello messages from. R2 will not consider R1 to be a neighbor unless R2 sees itself listed in the Hello messages of R1, indicating connectivity is two-way. When choosing a DRB, R2 ignores any routers for which connectivity to R2 is not two-way. Therefore, if there were a shared link with strange connectivity properties, the routers on the link might partition into cliques, each with its own DRB, each clique representing a separate link to the rest of the routers.

A surprising aspect of the use of IS-IS for TRILL was that the Hello protocol had to be modified slightly. In Layer 3 IS-IS, Hello messages are padded to the maximum size, because a possible hardware failure mode was that a link between R1 and R2 might be able to transmit small packets, but not large packets. In Layer 3, the IS-IS assumption was that R1 and R2 would rather not see that they were potential neighbors than use a flaky link. In IS-IS, LSP packets can be fragmented only by the source R1. All routers agree upon the maximum size of an LSP fragment that is guaranteed to be able to traverse all the links. Links that cannot forward packets of that size are not reported in the topology, and indeed, in Layer 3 IS-IS, would not even be discovered in the topology, because the Hello message (padded to that size) would not be seen by the neighbor router.

But with TRILL, it is important that only a single RBridge be elected DRB, because the DRB determines which RBridge will encapsulate/decapsulate packets for each VLAN. One of the first implementations of TRILL wound up forming a loop, where two RBridges, R1 and R2, both performed encapsulation/decapsulation. This situation resulted because neighbors R1 and R2 did not see each other's Hello messages, because the R1 Hello, padded to classic Ethernet maximum size by R1, became too large to forward when a VLAN tag was added, so did not reach R2.

To ensure that only a single RBridge on a link would be elected DRB, TRILL modified the Hello protocol as follows:

- Limit the size of Hello messages and do not pad them (in order to remove artificial impediments to receipt by neighbors).
- Elect a DRB based solely on priority (not two-way connectivity as in Layer 3 IS-IS). In other words, defer to a higher-priority RBridge R1 even if R1 does not list you as a neighbor.
- Have a separate mechanism for probing, using packets of different sizes, to see what size packets can be forwarded on the link.

In addition to solving the multiple-DRB problem, this design enables TRILL to discover which links can handle jumbo-grams, so that paths can be engineered that can forward jumbo-grams.

If the link between R1 and R2 is not acceptable because it cannot handle the assumed LSP fragment size, or because connectivity is not two-way, the link is not reported in LSPs. The capability of a link to handle larger sizes can be reported in LSPs.

There was enough confusion about this minor change to the Hello protocol, and skepticism that the Hello mechanism, which has worked correctly for Layer 3 for decades, would need to be modified for TRILL, that an additional RFC was written [3] to specifically explain the TRILL Hello mechanism.

Multidestination Frames

Multiple Trees

The original design for TRILL had the RBridges compute a single, shared tree, based on the LSP database, and all multidestination traffic was forwarded along that tree. But, to be able to load-split the use of links for multidestination traffic, a facility for using multiple trees was added early in the development of the TRILL standard.

In TRILL, the RBridge with the highest priority to be a TREE root announces to the other RBridges (through its LSP) how many trees, and which trees, should be calculated. A tree is calculated as a tree of shortest paths from a given Root, with a deterministic tie-breaker so that all RBridges calculate the same tree. The Root can be an RBridge or a pseudonode. In some cases, a Root is particularly well-situated in the topology such that its tree forms good paths for all pairs of nodes, but it is desirable to have multiple different trees, choosing different tie-breaker links, calculated from the same Root. TRILL accomplishes this setup by having that Root acquire multiple nicknames, one for each tree, and using the tree number in the tie-breaker algorithm, so that although all the trees from that Root will still be shortest-path trees, different links will be chosen in the different trees.

When R1 encapsulates a multidestination frame, R1 sets the "multi-destination" flag and specifies the tree Root nickname in the "egress RBridge" field in the TRILL header.

Filtering

A multidestination frame will be tagged with a VLAN (in the inner header). The frame need not be delivered to all RBridges—just those that are connected to a port with end nodes in that VLAN. So RBridges announce, in their LSPs, which VLANs they are attached to, where "attached to," means that they are acting as Appointed Forwarder.

Additionally, TRILL provides for filtering based on Layer 2 MAC addresses derived from IP Multicast groups. RBridges announce the set of such MAC addresses they wish to receive. The first RBridge that accepts an IP Multicast control message, such as *Internet Group Management Protocol* (IGMP), snoops on it [5] and learns what multicast listeners or multicast router is attached. This snooping is used so R1 can report in its LSP the IP Multicast groups it wishes to receive (or all groups if a multicast router is attached).

One other refinement to multidestination is the *Reverse Path Forwarding* (RPF) check. To safeguard against loops, when R is calculating which subset of its ports belong to a particular tree, R also calculates, for each port, the set of ingress RBridges whose traffic on that tree should arrive on that port.

So, the processing of a multidestination frame received by R, with TRILL header indicating Ingress = R1 and Egress/tree Root = R2, is as follows:

- If the port on which R receives the packet is not included in the tree "R2," discard the packet.
- If the port on which R receives the packet is in tree R2 but R1 is not listed in the RPF information for that port for tree R2, discard the packet.
- For each other port in R2, if the specified VLAN is reachable through that port and the IP Multicast address is requested by an RBridge along the path through that port, forward the packet on that port.

IS-IS Pseudonodes

If there is a link with N RBridges, rather than modeling the link as having on the order of N^2 links to be reported in LSPs, IS-IS has the DRB model the link as a pseudonode. The DRB gives the pseudonode a name, and the RBridges on the link report connectivity just to the pseudonode. The DRB generates an LSP on behalf of itself, reporting connectivity to the pseudonode, but additionally generates an LSP on behalf of the pseudonode, reporting connectivity to all the RBridges on the link. This portion of IS-IS is as designed from the beginning (from its origin as Phase V DECnet routing).

When IS-IS was originally designed, Ethernets tended to be very large shared links. But today, most Ethernets are simply point-to-point links (unless there are bridges making them appear to be shared links). So it would be wasteful for RBridges to always create a pseudonode for each Ethernet link. In Layer 3 it is not as unreasonable to always treat an Ethernet as a large shared link because an "Ethernet" link, as perceived by Layer 3, is likely to be a large collection of point-to-point links glued together with either bridges or RBridges.

But RBridges are likely to often see Ethernet links with just a single neighbor, especially in a topology with no bridges. So TRILL has the ability for the DRB to specify to its neighbor RBridges whether to report the link as a pseudonode or to report connectivity to all the RBridge neighbors as separate links. By default, the DRB R sets a flag known as the "bypass pseudonode" flag in its Hello message on the link, unless at some point since R rebooted R has seen two simultaneous neighbor RBridges on that link. With this mechanism, true point-to-point Ethernet links will be reported as a link between R1 and R2 rather than a pseudonode P, with links R1–P, R2–P, and P–R1 and P–R2 reported.

TRILL Implementations

TRILL is being widely implemented. TRILL fast-path hardware is included in chips available from all major merchant silicon manufacturers. A successful interoperability test was held at the University of New Hampshire *InterOperability Laboratory* in late 2010, and TRILL products are announced and shipping.

Future Potential TRILL Enhancements

Here are just three enhancements to TRILL being considered:

- Data centers require more VLANs than can be specified in 12 bits with a single VLAN tag. A TRILL extension to optionally include the ability to encode 24 bits of VLAN-like labeling in TRILL data frames is being considered.
- By optionally giving a pseudonode a nickname and having the appointed forwarder use that nickname in the ingress RBridge field, if the appointed forwarder changes, the end-node learning cache of distant RBridges will still be correct.
- A proposal is being made allowing IS-IS to be hierarchical in a TRILL campus. IS-IS hierarchy partitions the LSP database so that any single RBridge LSP database will be smaller, its path computation will be less computation-intensive, and it will lower the amount of LSP traffic. In particular, it shields the effects of a link that is cycling quickly from most of the campus, because only the RBridges in the region with the link will see reports of the state of that link.

Summary

The TRILL standard creates a cloud with a flat Ethernet address, so that nodes can move around within the cloud and not need to change their IP address. Although nodes attached to the cloud perceive the cloud as an Ethernet while the packet is traversing the cloud, it is encapsulated with a TRILL header, which like a Layer 3 technology, contains a source (ingress RBridge), destination (egress RBridge), and hop count. The addresses in the TRILL header are 16 bits, enabling a TRILL campus to support 64,000 RBridges. Transit RBridges do not learn about location of end nodes—only the existence of, and path to—other RBridges.

TRILL can use all the Layer 3 techniques, including shortest paths, *Equal Cost Multipath* (ECMP), and traffic engineering. It also supports VLANs and multicast. TRILL can calculate multiple trees, so that multidestination traffic can be split across links. Multi-destination frames can be filtered based on VLAN and IP (v4 or v6) Multicast groups.

TRILL is compatible with existing Ethernet bridges (switches), so a bridged Ethernet can be gradually upgraded by replacing any subset of the bridges with RBridges. The more that are upgraded, the better the bandwidth usage, and the more stable the network becomes.

References

[1] Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification," RFC 6325, July 2011.

[2] "Information technology—Telecommunications and information exchange between systems—Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)," ISO/IEC 10589:2002.

[3] Eastlake 3rd, D., Perlman, R., Ghanwani, A., Dutt, D., and V. Manral, "Routing Bridges (RBridges): Adjacency," RFC 6327, July 2011.

[4] ITU-T, "X.200: Information technology—Open Systems Interconnection—Basic Reference Model: The basic model," July 1994.

[5] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches," RFC 4541, May 2006.

[6] Touch, J. and R. Perlman, "Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement," RFC 5556, May 2009.

[7] W3C, "XML Base (Second Edition)," W3C Recommendation 28 January 2009,
`http://www.w3.org/TR/2009/REC-xmlbase-20090128/`

[8] Perlman, R., "A Protocol for Distributed Computation of a Spanning Tree in an Extended LAN," *9th Data Communications Symposium,* Vancouver, 1985.

[9] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments," RFC 1195, December 1990.

[10] Moy, J., "OSPF Version 2," RFC 2328, April 1998.

[11] Simpson, W., "The Point-to-Point Protocol (PPP)," RFC 1661, July 1994.

[12] **http://www.interfacebus.com/HDLC_Protocol_Description.html**

[13] Carlson, J. and Eastlake 3rd, D., "PPP Transparent Interconnection of Lots of Links (TRILL) Protocol Control Protocol," RFC 6361, August 2011.

RADIA PERLMAN is a Fellow at Intel Labs, working on the design of various network routing and security protocols. She is the inventor of the Spanning Tree Algorithm, the designer of IS-IS, and the original concept for TRILL. She is the author of the textbook *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*. She is an IEEE Fellow and holds a Ph.D. from MIT.
E-mail: **radiaperlman@gmail.com**

DONALD EASTLAKE 3rd is Co-Chair of the IETF TRILL Working Group and a voting member of IEEE 802.1. He is the author of 56 IETF RFCs and a Principal Engineer with Huawei Technologies working on advanced network product research. Previously, he was a Principal Engineer at Cisco Systems and before that a Distinguished Member of Technical Staff at Motorola Laboratories, working on network protocols, security, and mesh networking.
E-mail: **d3e3e3@gmail.com**

The Case for IP Backhaul

by Jeff Loughridge, Brooks Consulting LLC

In any hierarchical network, designers must specify how the access layer delivers traffic to the core. In *Mobile Network Operator* (MNO) networks, the transport of voice and data from the cell sites to the wireless MNOs' core networks is called *backhaul*. *Time Division Multiplexing* (TDM) backhaul has dominated backhaul deployments since the inception of wireless communication. Leasing the backhaul access of multiple T1s/E1s for every cell site becomes prohibitively expensive in terms of operating expenses, particularly for providers that do not own the last mile. Today's 3G/4G cellular technologies have spurred a major change in the backhaul network: the transition from TDM to packet backhaul.

Ethernet is the most widespread packet-based backhaul technology. While this service is a vast cost and scale improvement over TDM backhaul, carrier Ethernet is a stepping stone in the evolution of backhaul networks. Expect MNOs to move to true IP backhaul networks to meet the scalability needs of their expanding networks. In this article, we will explain mobile backhaul evolution, shortcomings in carrier Ethernet backhaul, and how evolving service requirements will motivate cell site backhaul vendors to add IP-awareness to their networks.

Legacy Backhaul

Cellular systems were initially designed to carry only voice traffic. Since transporting digitized voice was a mature and well-understood technology, there was no need to take a divergent path for the backhaul of voice traffic in early cellular systems. Using TDM had obvious advantages among those being:

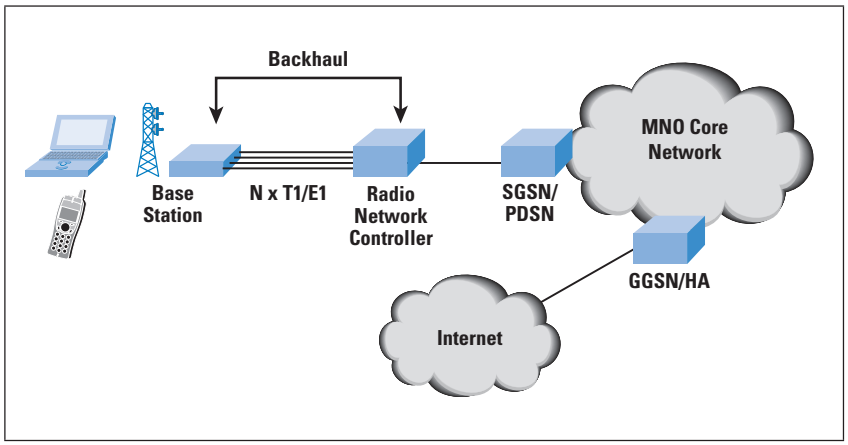
- Use of the same equipment used in wireline voice transmission
- Technical staffs' familiarity with TDM concepts and troubleshooting
- Ability to use existing *Operations, Administration, Maintenance, and Provisioning* (OAM&P) systems
- Ubiquity of the T1/E1 service

The initial work to offer data service on cellular systems naturally focused on adding data transmission to the existing voice infrastructure. Standards such as *Global System for Mobile Communications* (GSM) and *Interim Standard 95* (IS-95) took similar approaches in borrowing TDM time slots for data. The data services of the 1990s were very slow, even when compared to consumer modems of the time. Standards developed in the late 1990s and deployed in the early 2000s (*Enhanced Data rates for GSM Evolution* [EDGE] and *CDMA2000*) improved data transfer speeds.

TDM was clearly entrenched as a foundational technology for data communication in cellular networks going into the early 3G technology deployments (*Universal Mobile Telecommunications System* [UMTS] and *Evolution Data Optimized* [EV-DO]).

Figure 1 depicts the backhaul portion of the MNO network and how it fits into the broader architecture.

Figure 1: The Backhaul Network in the MNO Architecture



As data traffic usage for 3G networks grew, shortcomings of TDM backhaul began to materialize. The two prominent areas were bandwidth and cost. Cell sites with TDM access are typically equipped with multiple T1/E1s. With faster radio interfaces, the backhaul became the bottleneck in the network. Some smartphones became consumers of multi-megabyte data rates. User experiences were poor on some wireless networks as a result of a dearth of bandwidth in the backhaul segment. Continuing to increase the number of TDM lines or increase their capacity was not a viable option since the growth increments were too small and the operating expenses were too high.

The second limitation of TDM in 3G networks is cost. Although the cost of T1/E1s decreased considerably over the years, the costs piled up given the number of cell sites and number of T1/E1s per site. This figure became the highest contributor to the cost of the backhaul network. The MNOs that owned the last mile were at a distinct competitive advantage compared with the carriers who had to pay another party (often in a minimally competitive marketplace) for TDM access. For MNOs to continue their incredible traffic growth rates, a new access model was needed.

Carrier Ethernet Adoption

Ethernet quickly emerged as the most popular backhaul technology to replace TDM access infrastructure (other providers moved forward with microwave access with varying levels of success). The various iterations of Ethernet from 1970s to 2000s had trumped other LAN technologies in the market, and at the turn of the century gigabit Ethernet leveraged its success in the LAN to become popular in the WAN. The technology had several major advantages:

- *Large drop in cost per bit:* Ethernet would allow providers to drastically alter their access cost model by supplanting the aging and costly TDM infrastructure. With the price that consumers were willing to pay per month of data service staying relatively stagnant, this adjustment to the cost model was critical.
- *Ethernet can be carried over more underlying technologies: Synchronous Optical Networking/Synchronous Digital Hierarchy* (SONET/SDH), *Generic Framing Procedure* (GFP), *Dense Wavelength Division Multiplexing* (DWDM), and *Multiprotocol Label Switching* (MPLS) are a few examples. A key benefit Ethernet's ability to operate over these technologies was that many providers could consolidate their wireless access with their existing and speedier wireline access networks.
- *Ethernet interfaces ubiquitous and inexpensive:* Ethernet won the battle for LAN dominance. The technology was not restricted to traditional personal computers and servers—printers, phones, game consoles, *Digital Video Recorders* (DVRs), and home media center hubs are some examples of other equipment that often included Ethernet interfaces. This ubiquity in the business and consumer spaces results in a diverse supplier set and economies of scale for the vendors and suppliers.
- *Ease of bandwidth upgrade:* TDM circuits have an implementation time measured in months. This slow turn-around time for upgrades is a poor fit for an environment in which data usages is increasing at fast rates. Ethernet is much different. An increase in bandwidth to a network end-point will not require a change in equipment unless moving between the established tiers of 10, 100, 1000 Mb/s. Since the Ethernet service vendor likely uses a "policer" to keep customers within the purchased bandwidth level, a change in software configuration is usually all that is required to upgrade bandwidth. Another advantage is that bandwidth can be upgraded in granular increments. With the right back-end systems, an upgrade will take a matter of minutes. For companies looking to increase the velocity of service deployment, the ability to quickly move to high speeds is very favorable.

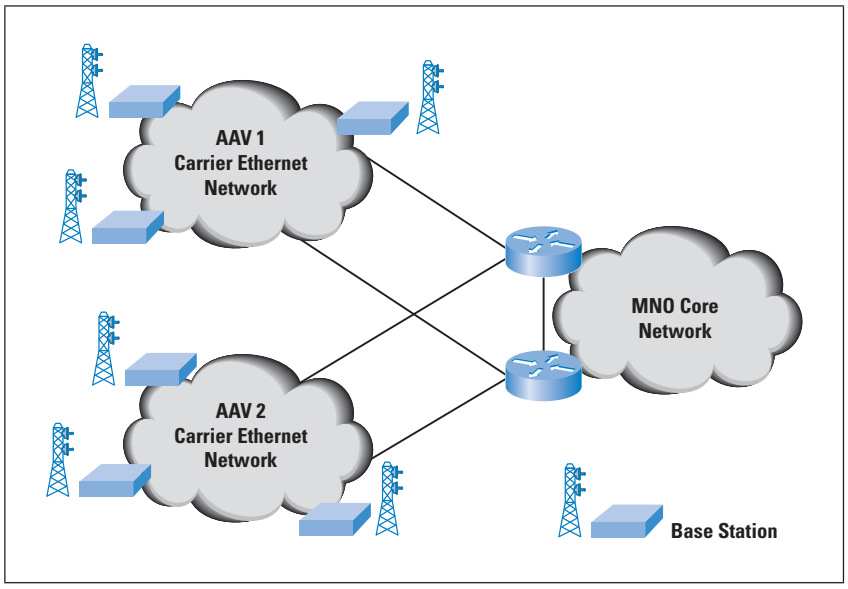
Established in 2001, the *Metro Ethernet Forum* (MEF) played a critical role in the acceptance of carrier Ethernet by wireless and wireline providers. The MEF is not a standards organization like the *Internet Engineering Task Force* (IETF). Instead, the MEF builds upon the work of standards bodies to establish common terminology, service requirements, and network interface requirements. The MEF created an architecture framework along with measurement and testing specifications. Although the MEF did not eliminate wireless providers' concerns about packet backhaul—particularly in the areas of jitter, delay, and packet delivery, the forum did increase the comfort level associated with metro Ethernet services. The MEF's E-LINE service definition established a connection-oriented path, a concept much more pleasing to traditional telcos than the perceived "anything goes" nature of packet switched networks. For more detail on the MEF's service definitions, see [0].

By the second half of the 2000s, many wireless providers were planning the deployment of Ethernet-based backhaul for new *High Speed Packet Access* (HSPA), *Worldwide Interoperability for Micro-wave Access* (WiMAX), and *Long-term Evolution* (LTE). In making this radical change, the providers often had to consider protecting existing revenue streams from voice and data (providers electing to move forward with greenfield deployments were at a luxury). Pseudowire technologies enabled the carriage of TDM traffic over IP/Ethernet networks, thus preserving investment in existing infrastructure.

Rather than build carrier Ethernet infrastructure, the MNOs that were not facilities-based (or had limited last mile footprints) purchased services from other parties, known as *Alternate Access Vendors* (AAV) in telco parlance In the United States, the *Local Exchange Carriers* (LECs) and cable companies were well positioned for this business. MNOs often used multiple AAVs in a given market to cover the cell site footprint. Getting fiber to cell sites outside of major metropolitan areas was not always possible, which led some MNOs to use hybrid backhaul solutions that included microwave and TDM inverse muxing in addition to carrier Ethernet.

Figure 2 illustrates how MNOs rely on AAVs to cover their cell site footprint in a given market.

Figure 2: Alternative Access Vendors



The adoption of carrier Ethernet services by MNOs was not without challenges. Mobility gear such as *Radio Network Controllers* (RNC), base stations, and *Home Location Registers* (HLR) historically relied on T1/E1 interfaces for connection to the network. Telecom vendors had to implement Ethernet interfaces along with IP stacks. The providers had to completely revamp provisioning, service monitoring, performance monitoring, and service assurance systems and processes. Consider the following example.

For years, operations groups at telcos counted on near-immediate notification with an alarm indication signal in the *Time Division Multiple Access* (TDMA) frame. TDMA frames arrive every 125 µsec (8,000 times a second). Packet-switched networks do not share the synchronous nature of TDM and do not have OAM fields in framing bits. The operators now had to rely on nascent specifications such as Y.1731 and 802.1ag for service monitoring.

Timing and synchronization—necessities in mobile networks—are gleaned from the physical layer in TDM networks. Asynchronous networks such as Ethernet/IP do not have an inherent mechanism for timing and synchronization. Keeping a single T1/E1 at the cell site is one method to ensure timing and synchronization in a carrier Ethernet scenario; however, the use of upper layer protocols is more appropriate, particularly for new builds that have no legacy TDM circuits. *Synchronous Ethernet* (SyncE), *Precision Time Protocol* (PTP, also known as IEEE 1588v2), and *Network Time Protocol version 4* (NTPv4) were deployed in backhaul networks to provide timing and synchronization. Note that SyncE transports timing information over the physical layer much like the TDM timing model, while PTP and NTP use IP for transport and are not dependent on an Ethernet physical layer.

The learning and flooding aspects of all Ethernet networks present inherent scaling challenges for very large networks. Spanning tree and its derivatives are commonly used to address these issues at low and medium scale. For larger networks that provide service to multiple customers, the service must scale in terms of its ability to offer service to multiple entities and in terms of the many switches required for an expansive footprint. Many protocols have arisen to solve one or both of these challenges. Examples are *Virtual Private LAN Service* (VPLS), *Multiprotocol Label Switching–Transport Profile* (MPLS-TP), and *Provider Backbone Bridging–Traffic Engineering* (PBB-TE). Being relatively new technologies, these can and do present challenges for operations groups. The breakages can occur in ways that are very difficult for the Carrier Ethernet provider and wireless provider to jointly troubleshoot.

The Next Step – IP Backhaul

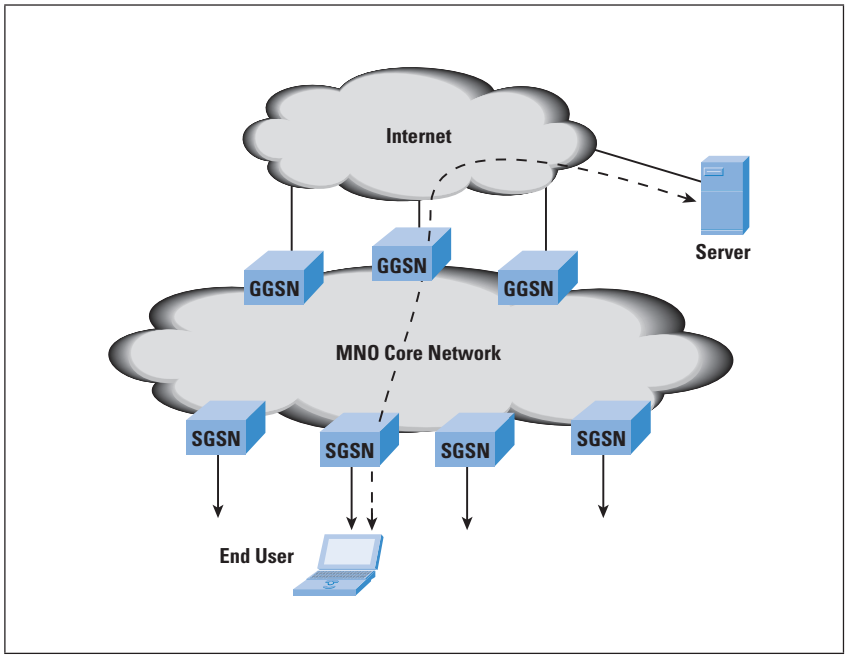
The phrase "all-IP" is frequently used to describe the most recent wireless technologies such as HSPA+, WiMAX, and LTE. This is applicable as the majority of network elements, including the handsets, are IP enabled. The existence of large-sized carrier Ethernet networks in the network architecture undermines the IP-centric argument. IP has superior scaling properties over Layer 2 networks. The footprint and number of nodes for carrier Ethernet networks continues to expand rapidly as the MNOs deploy 3G and 4G networks. The author sees evidence that protocols used to overcome Ethernet scalabilities issues will become increasingly complex and push MSOs and AAVs toward Layer 3-centric backhaul networks.

Before delving into the drivers of IP backhaul, let's examine a typical data traffic flow for today's wireless networks. We'll use the 3GPP's *GSM Packet Radio System* (GPRS) as this is the most common in world-wide deployments. Data flows are very centralized in this architecture. Macro-level mobility is controlled by two types of *GPRS Support Nodes* (GSN): *Gateway GPRS Support Nodes* (GGSN) and *Serving GPRS Support Nodes* (SGSN). GGSNs are typically deployed within the mobile core network at locations with Internet access. This is often at centralized mobile switching centers. SGSNs can be deployed closer to the network edge and multiple SGSNs can be served by a single GGSN.

The GGSN is the mobility anchor, much like the home agent in wireless networks that use Mobile IP. The SGSN is akin to the foreign agent in Mobile IP. GPRS network tunnel traffic between SGSN and GGSN using an IP-in-IP tunneling protocol called *Generic Tunneling Protocol* (GTP). Although GTP has several purposes in the GPRS core network, our focus will be on its tunneling of packets between SGSN and GGSN (called the *Gn* interface). The movement of the subscriber to a region served by another SGSN will trigger a macro-mobility event. A new GTP tunnel is formed using the original GGSN for session continuity [2].

Since all traffic from the *Mobile Subscriber* (MS) must traverse the GGSN as the mobility anchor, the traffic flow from the MS follows a very predictable path to a centralized location. Note that there is not a 1:1 relationship between SGSNs and GGSNs. As mentioned earlier, typical deployment of GGSNs is very centralized. Figure 3 depicts the flow.

Figure 3: Data flow in a GPRS Network



Although technologies like LTE are touted as flat IP networks, this only holds true from a *Radio Access Network* (RAN) perspective. What if a subscriber wants to communicate with another subscriber in the same building or local machine-to-machine traffic is highly sensitive to latency? The packets will be sent to the mobility anchor, perhaps hundreds of kilometers away. Routing decisions can be made in the RAN and core network; however, the decision is restricted since traffic must traverse the predefined tunnel endpoints.

Wireless networks will gradually decentralize and distribute mobility management. In 3G networks, some providers have been extending the core network closer to the subscriber as mobile gateways (GSNs and their equivalents in non-3GPP networks) become more cost-competitive. By deploying mobile gateways at what were previously aggregation *Points Of Presence* (POPs) and buying Internet connectivity at these locations, Internet-bound traffic exits the network quickly, consuming fewer resources for the provider. Other signs of this shift are evident in LTE and WiMAX. LTE's S1-flex interface allows the RAN to be connected to multiple core networks. The WiMAX reference model separates the *Network Access Provider* (NAP) and *Network Service Provider* (NSP). The NAP, which provides radio access functionality, can connect to multiple NSPs for Internet connectivity.

To fully realize the benefits of an IP-centric backhaul, steps must be taken to go beyond simply distributing mobility management. New solutions are needed to eliminate mobility anchoring via tunneling. Vendors, providers, and universities have already started to examine how to dispose of tunneling in the mobile environment [2].

The IP-centric backhaul network has many advantages over the carrier Ethernet networks that enable many of today's packet backhaul networks. Various advantages benefit the wireless providers, the IP backhaul provider, or both. These advantages are most prevalent when the MSOs have a highly distributed mobility management architecture.

- *Backhaul Offload:* Today's mobile elements at the cell tower have no ability to influence routing decisions; there is only one path to the core network. Adding egress points to the cell site or backhaul network reduces the distance and amount of traffic that must be backhauled. To accomplish the addition of egress points in a carrier Ethernet network, connection-oriented mechanisms such as Ethernet Virtual Circuits would require that the MSO and AAV modify multiple network elements' configurations. Offloading traffic with an IP network is substantially more simple and scalable. Offloading packets from the backhaul will represent a huge savings in access costs. The base station could be capable of hot potato routing traffic directly to an ISP instead of backhauling commodity Internet traffic to the MSO, where the costs of equipment, power, and software licenses quickly accumulate.

- *Multicast:* The reliance on tunneling as described earlier in this piece severely restricts the usefulness of multicast in current wireless networks. Distributing the mobility elements controlling the tunneling closer to the subscriber will mitigate these effects as would the elimination of mobility anchoring via tunneling techniques. The implementation of a true flat IP network would extend multicast capability into the RAN and position both MNOs and IP backhaul providers to realize the efficiency gains of multicast.
- *Localized Content and Peering:* With localized egress points, local content could be reached directly rather than traversing the core network. This would position wireless providers to peer with other providers at the local or regional level, a benefit that would be substantial for wireless providers operating in countries with non-meshy Internet infrastructure and expensive wide-area communications lines. In addition, caches could be implemented much closer to the subscriber to improve the user experience for video and other content types.
- *Machine-to-Machine (M2M) and Peer-to-Peer (PtP):* When the communication is device to device in close geographic proximity, the traversal of the core network only adds latency, complexity, and cost. A distributed mobility management architecture and IP backhaul network engender an optimized path for M2M and PtP. The mobility anchor point could be placed at the cell tower or local aggregation point, providing a much improved communication path for subscribers and machines connected to the wireless network.
- *Uptime and Reliability:* Wireless providers have experienced challenges with carrier Ethernet service. Some of these problems can be chalked up to the relative newness of using carrier Ethernet for cell site backhaul. One has to wonder though, what experience exists in the industry for maintaining giant Layer 2 networks? The number of mobile devices will expand exponentially, triggering the deployment of thousands of new cell sites, microcells, and picocells. The author is less than confident that any underlying technology that enables carrier Ethernet will scale to the necessary degree while maintaining the uptime and reliability that users expect from their data service.

For large IP networks, the industry has over fifteen years' experience in designing, engineering, and operating IP networking carrying traffic at staggering capacities. The staff expertise, software maturity, and systems support exists today to maintain sizable IP networks. There are established best practices for Tier 1 ISPs that help ensure long uptime, speedy convergence upon failure, and sound network design.

Delivering an IP Backhaul Service

IP backhaul offerings could be delivered in a variety of ways. The simplest design for IP backhaul providers would be a shared IP transport network that commingles traffic between customers.

The wireless providers could then use protocols such as *Layer 2 Tunneling Protocol version 3* (L2TPv3) to build an MPLS/VPN-like overlay to provide logical separation and address overlap prevention. The preferred approach for MNOs would likely be a Layer 3 VPN service from the AAV, thereby offloading much of the routing complexity from the MNO.

An IP backhaul service must be capable of routing IPv6 packets, as the useful lifetime of an IPv4-only service is limited. MNOs cannot obtain new IPv4 addresses to number the base stations, and using RFC 1918 space is not a scalable approach. Using IPv6-only to address mobility equipment at cell sites (and equivalent radio interfaces) is the preferred method for overcoming the scarcity of IPv4 addresses.

The shift from carrier Ethernet to IP backhaul should not be a monumental one for many carrier Ethernet providers. The heavy lifting of installing fiber and deploying a packet switched infrastructure has already been accomplished. In addition, carriers that implement carrier Ethernet with protocols like VPLS already have an infrastructure that is ready for IP. The most challenging aspect of the transition will be the work needed to prepare OAM&P systems for an IP service. Of course, this may vary based on carrier Ethernet implementation and systems.

Conclusion

Carrier Ethernet service for cell site backhaul is a vast scale and cost improvement over TDM backhaul and has been extremely successful. OSI Layer 3 IP networks have superior scaling properties that will replace Layer 2 backhaul networks of today. Advances in wireless networking systems, the proliferation of new devices, and the development of new mobility services will be best served with a truly IP-centric backhaul network.

References

[0] Santitoro, Ralph, "Metro Ethernet Services—A Technical Overview," 2003, **http://metroethernetforum.org/metro-ethernet-services.pdf**

[1] M. Grayson, K. Shatzkamer, and S. Wainner, *IP Design for Mobile Networks,* Cisco Press, 2009.

[2] *Distributed Mobility Management in Future Wireless Networks* (DiMoWiNe), **http://conference.researchbib.com/print.php?category=event&id=10232&uid=6**

JEFF LOUGHRIDGE is the principal consultant and owner of Brooks Consulting LLC, a firm that specializes in Tier 1 ISP best practices and the design, engineering, and operations of large-scale wireline and wireless IP/MPLS networks. Prior to founding Brooks Consulting, Jeff spent over ten years supporting Sprint's global IP network in both technical and managerial capacities. He earned a bachelor's degree in computer science from Duke University and an MBA from the University of Phoenix—Northern Virginia campus.
E-mail: **jeffl@brooksconsulting-llc.com**

Fragments

Global INET 2012

To help mark its 20-year-anniversary, the *Internet Society* (ISOC) is hosting a global forum that will bring together visionaries and thought leaders from around the world to focus on issues that will impact the future of the Internet.

The *Global INET 2012,* which is scheduled to take place in Geneva, Switzerland from April 22–24, will feature high-powered speakers, thought-provoking panel discussions, and interactive workshops to develop a vision for the explosive growth of the Internet over the next 20 years.

Thought leaders from across the Internet community will collaborate on topics critical to the global Internet's future, including privacy, net neutrality, IPv6, security, digital content and innovation, and human rights and freedom of expression.

Since its beginnings in 1992, ISOC has been dedicated to helping keep the Internet open, accessible, and defined by users—regardless of where they live, what they do, their abilities, or who they are.

Registration for Global INET 2012 is scheduled to begin in October 2011.

For more information:

[1] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, Stephen Wolff, "A Brief History of the Internet," December 2003, also published in ACM's *Computer Communication Review,* Volume 39, Number 5, October 2009.
http://www.isoc.org/internet/history/brief.shtml
http://www.sigcomm.org/ccr/papers/2009/October/1629607.1629613

[2] "The Internet Society's Principles and Goals,"
http://www.isoc.org/isoc/mission/principles/

[3] **http://www.isoc.org/isoc/conferences/inet/12/gva.shtml**

IPv6 Week

IPv6 Week will be a coordinated test of the new Internet Protocol, held February 6–12, 2012. Websites, content providers, Internet Services Providers, Network Service Providers, as well as end users are invited to participate. This is a Brazilian initiative, but anyone can participate.

For more information visit: **http://www.ipv6.br/IPV6/WeekIPv6**

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

December 2011 Volume 14, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

Depletion of the IPv4 address space and the transition to IPv6 has been a "hot topic" for several years. In 2011, interest in this topic grew considerably when the *Asia Pacific Network Information Centre* (APNIC) became the first *Regional Internet Registry* (RIR) to start allocating addresses from its final /8 IPv4 address pool. Although depletion dates are difficult to predict accurately, there is no question that the day will come when it will no longer be possible to obtain IPv4 space from the RIRs. News stories about IP addresses being sold for considerable sums of money are becoming more common.

Numerous organizations have been working diligently to promote, test, and deploy IPv6 through efforts such as the *World IPv6 Day,* while the *Internet Engineering Task Force* (IETF) continues to develop solutions to aid in the transition. One such effort, the *Port Control Protocol* (PCP), is described in our first article by Dan Wing.

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will soon begin accepting applications for new *Top-Level Domains* (TLDs). It is not yet known how many new TLDs will eventually be deployed, but the plans have prompted several studies focused on the resiliency and scalability of the *Domain Name System* (DNS). Bill Manning discusses some of the technical challenges associated with a vastly expanded TLD space.

The IETF *Homenet Working Group* "...focuses on the evolving networking technology within and among relatively small 'residential home' networks. For example, an obvious trend in home networking is the proliferation of networking technology in an increasingly broad range and number of devices. This evolution in scale and diversity sets some requirements on IETF protocols." Geoff Huston gives an overview of some of the challenges facing this Working Group.

The product of the IETF is a set of documents, mainly protocol specifications and related material. These documents start life as *Internet Drafts* and proceed through a series of iterative refinements toward eventual publication as *Request For Comments* (RFCs). Over time, several *tools* have been developed to aid in the document development process, and they are now organized at the IETF Tools webpage. We asked Robert Sparks to give us an overview of some of the most important tools and the process involved in their development.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Port Control Protocol

by Dan Wing, Cisco Systems

After the transition to *Internet Protocol Version 6* (IPv6), hosts will often be behind IPv6 firewalls. But before the transition, mobile wireless devices will want to reduce their keepalive messages, and hosts of all sorts will share IPv4 addresses using a variety of address-sharing technologies. To meet these needs, the IETF formed the *Port Control Protocol Working Group* in August 2010 to define a new protocol for hosts to communicate with such devices. The initial output of this Working Group is the *Port Control Protocol* (PCP)[1]. Interoperability between two independently developed implementations of PCP was demonstrated at the IETF meeting in July 2011, highlighting the importance of this protocol to the industry. After it becomes a standard, PCP is expected to be deployed in various operating systems, IPv6 home gateways, IPv4 home gateways (*Network Address Translators* [NATs]), mobile third- and fourth-generation (3G and 4G, respectively) gateways (*Gateway GPRS Support Nodes* [GGSNs]), and *Carrier-Grade NATs* (CGNs).

Introduction to PCP

PCP performs two major functions: It allows packets to be received from the Internet to a host (such as to operate a server), and allows a host to reduce keepalive traffic of connections to a server. PCP can be extended in two ways: with new *OpCodes* or with new *Options*. The base PCP specification defines two OpCodes: MAP and PEER, and defines several Options that can be carried with those OpCodes.

To operate a server, packets are sent from a host on the Internet to a server. The IP model expects devices to be connected to a network and be able to exchange packets with each other. However, few deployed networks actually permit hosts to receive packets from the Internet because of business needs (for example, to protect wireless spectrum from malicious or accidental packets originated on the Internet) or because of technology restrictions (for example, IPv4 address-sharing devices such as *Network Address and Port Translators* [NAPT]). To operate a server, a host uses the MAP OpCode.

To reduce keepalives, a host needs to send traffic before a middlebox will destroy an idle connection. Many middleboxes, such as firewalls or NATs, maintain state and will destroy mappings if the connection has been idle. Today, in order to prevent destruction of mappings, hosts send keepalive traffic to keep those mappings alive. The keepalive traffic has several disadvantages, including reduction of battery lifetime, network chatter, and server scalability (servers have to discard the keepalive traffic). PCP allows a host to determine how aggressively a middlebox will destroy an idle connection, allowing the host to reduce its keepalive traffic with the PEER OpCode.

PCP is encoded in binary and carried over the *User Datagram Protocol* (UDP), which eases implementation on clients and servers. The client is responsible for retransmitting messages, and all messages are idempotent. The PCP client can be part of the operating system (much like a *Dynamic Host Configuration Protocol* [DHCP] client or a *Universal Plug and Play* [UPnP] *Internet Gateway Device Protocol* [IGD] client) or the PCP client can be coded entirely in an application (much like any other application-level protocol such as the *Network Time Protocol* [NTP]). A major feature of PCP is its flexibility and simple messaging, so it can be implemented easily in a variety of systems and at high scale.

Security

When installing an IPv4 NAPT on a residential network, the NAPT has a side effect: it prevents unsolicited incoming traffic from reaching hosts inside the home. Traffic that originates inside the home can traverse the NAPT toward the Internet. This function is expected by many users to such a degree that when IPv6-capable routers were first installed on residential networks, users complained that their IPv6 hosts were seeing traffic from the Internet. This visibility meant that IPv6 printers, webcams, and other hosts had to be protected from malicious traffic from the Internet. Based on this experience, IPv6 *Customer Premises Equipment* (CPE) routers intended for installation in the residential market filter most unsolicited incoming traffic by default[3]. Thus, IPv6 CPE routers provide filtering similar to what users experience today with IPv4 NAPT devices.

With both IPv4 NAPT and RFC 6092 IPv6 routers, outgoing traffic from a host creates a mapping that then allows bidirectional traffic to a specific (*Transmission Control Protocol* [TCP] or UDP) port on the internal host, meaning when a host sends a TCP SYN, a SYN ACK can be returned to the host. Neither IPv4 NAPT devices nor RFC 6092 IPv6 routers have to do any additional filtering of that mapping, and after that mapping is created will allow traffic from any host on the Internet to reach the internal host—not just traffic from that particular host. This lack of filtering is necessary for certain applications to function.

PCP was built with a security model similar to that deployed on home networks. With PCP, a host can send a PCP packet requesting a mapping so that any host on the Internet can now initiate communications with the internal host. Similarly, without PCP, a host could send a TCP SYN from a specific port (for example, port 80), thereby creating a mapping nearly identical to a PCP mapping. As with sending a TCP SYN, PCP allows a host to open mappings only for itself, unless the network administrator has taken the extra step to enable the PCP THIRD_PARTY option.

You may wish to have additional restrictions for some networks. PCP is extensible to support authorization, and there is ongoing work to support authentication and authorization within PCP[8].

PCP is extensible and there are already several proposed extensions to the protocol, including a way to control which IP address pool is assigned to a mapping[5], bulk port allocation to optimize acquiring a large set of ports[6], and rapid recovery after NAT failure or network renumbering[7].

PCP Scenarios

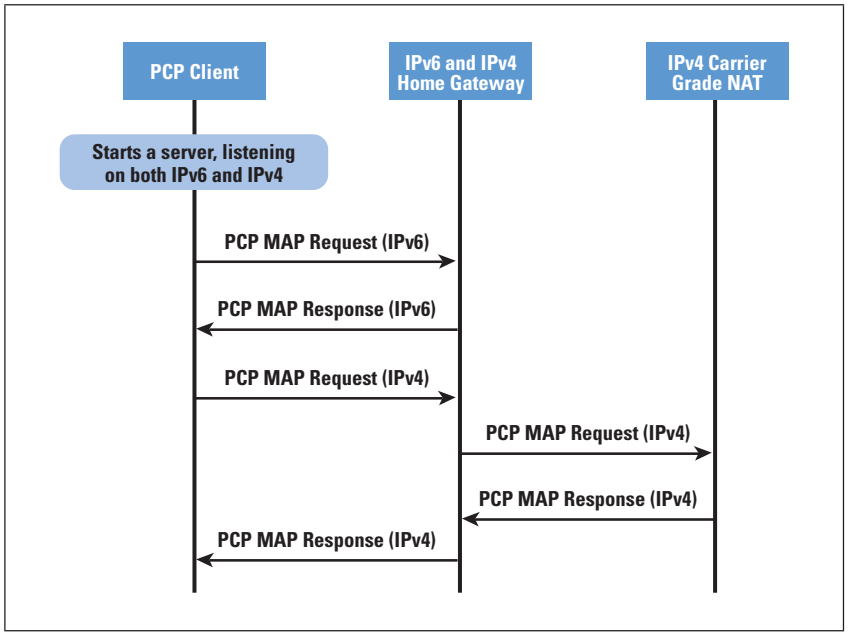
PCP works in all scenarios with IPv4 address sharing (using an IPv4 NAT or using other techniques), an IPv4 or IPv6 firewall, and NATs that translate from IPv6 to IPv4, IPv4 to IPv6, or IPv6 to IPv6. When working with nested NAT, such as a NAT in the home and a NAT operated by the *Internet Service Provider* (ISP), PCP can create the NAT mappings in both devices. When working with IPv6, PCP can create mappings in an IPv6 CPE router. In some networks we expect to see IPv6-only devices that IPv4 clients may need to access. For those devices to work, an IPv6/IPv4 translator (NAT64)[10, 11] can translate between IPv6 and IPv4. PCP can work with an IPv6/IPv4 translator as well. In other scenarios IPv6/IPv6 translation may be necessary, and although translating IPv6 to IPv6 is far from desirable, PCP can also support IPv6/IPv6 (NPTv6)[12].

A server, such as a one running on a sensor (for example, thermometer or electric meter), can use PCP to determine its publicly routable IPv4 or IPv6 address and port, and then populate a *Rendezvous* server with that IP address and port. For example, an IPv6-only thermostat might want to be accessible over IPv6 and IPv4, so it can be accessed by both the power company (to push new electricity rate information to the thermostat) and the homeowner (who might have IPv4 access only at work). The thermostat can use PCP to create a TCP mapping in the IPv6 CPE router (necessary because the IPv6 CPE router will, by default, filter unsolicited incoming IPv6 packets) and use PCP to create a TCP mapping in a NAT64 (necessary so the homeowner can access the thermostat). The IPv6 address and its TCP port, and the IPv4 address and its TCP port, can be published to the *Domain Name System* (DNS) (using DNS Server [SRV] records) or published to some other Rendezvous server. Then the power company or the homeowner can use the DNS (or the other Rendezvous server) to communicate directly with the thermostat.

Because PCP can inform the PCP client of address changes, network renumbering can be communicated immediately to hosts—something that cannot be done with most other NAT or firewall control mechanisms. Therefore, devices running on nomadic networks, such as in a connected vehicle, that use PCP will immediately learn when they have connected to a new network. This knowledge can allow them to update information in the DNS or in some other Rendezvous server so they remain accessible from the Internet.

PCP is expected to be implemented in home gateways and Carrier-Grade NATs, which provide value for both IPv6 (to operate a server and learn keepalive timeouts) and IPv4. Figure 1 shows how a dual-stack host would use PCP to operate an IPv6 or IPv4 server.

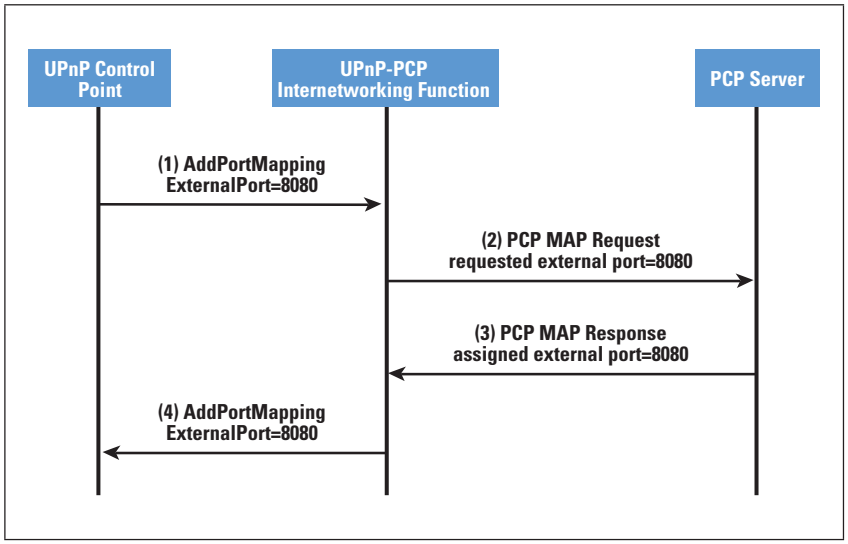
Figure 1: PCP Mapping IPv6 and IPv4



PCP Interworking with UPnP IGD

UPnP IGD Version 1 is widely available on residential-class NAT devices and host operating systems (Windows and OS X). However, because of security concerns it is often disabled by vendors, ISPs, or end users. UPnP IGD itself only works with a single layer of NAT, but it is possible to interwork between UPnP IGD and PCP[4]. To do this interworking, a home gateway (NAT) processes UPnP IGD messages on its LAN interface and translates those messages to PCP messages on its WAN interface, as depicted in Figure 2.

Figure 2: UPnP-to-PCP Interworking, Showing AddPortMapping Success



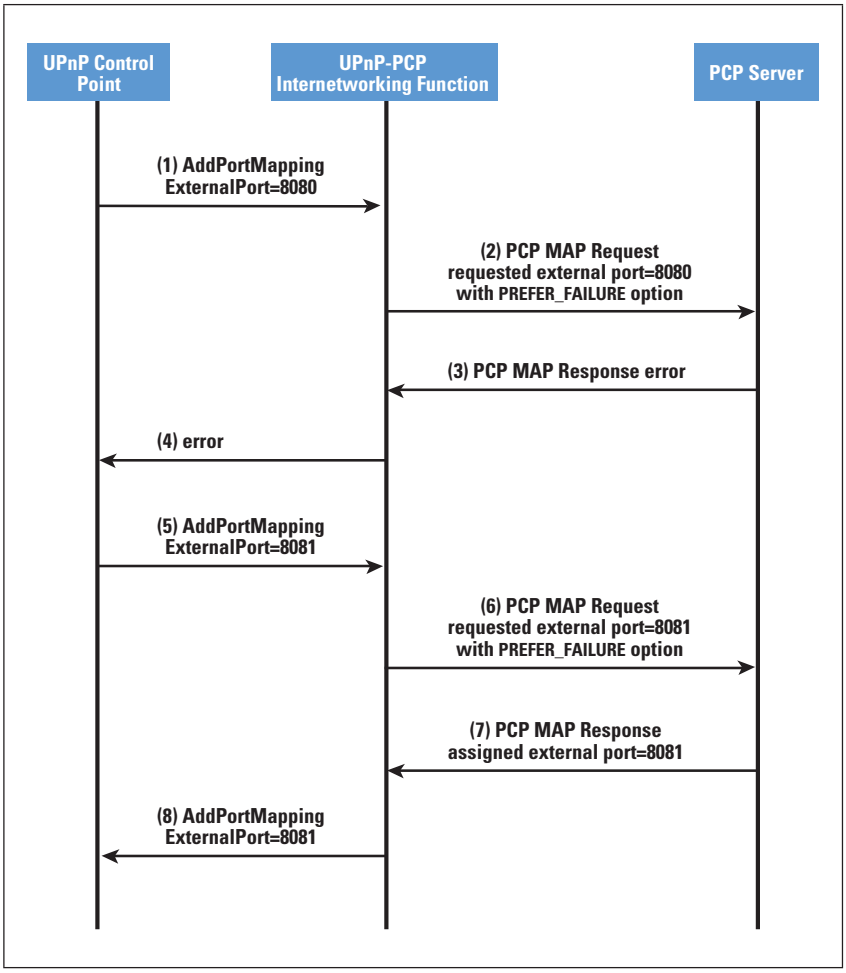
One difficulty with UPnP IGD is its AddPortMapping action, which maps a specific port on the home gateway. If that requested port is already mapped to another host, that port cannot be mapped to a new host (because it is already mapped to a different host). This problem exists today with UPnP IGD if two hosts in a home need the same port (for example, TCP port 80) because only one of them can map the port. In a CGN environment, where many subscribers share one IPv4 address, it is almost guaranteed that another subscriber has already mapped a "good" port (for example, 80 for HTTP, 8080 for HTTP, 5001 for Slingbox, 5060 for *Session Initiation Protocol* [SIP], etc.). Today, when a UPnP IGD port mapping is refused, the application may overwrite the first host's mapping (causing significant problems), "hunt" for an available port, or simply give up and display an error to the user. The "hunting" is often sequential (trying the next-higher port number) but is sometimes random, and is done by the application itself, the operating system UPnP framework, or both.

UPnP IGD Version 2[2] introduced the *AddAnyPortmapping* action, which avoids the need to "hunt" for an available port and allows the NAT to assign an available port. But UPnP IGD Version 2 is not yet widely available in home gateways, operating systems, or applications. Until IPv6 is ubiquitously available, applications (and users) will need to practice better port agility than has been practiced in the past, because "good" ports will simply not be available when IPv4 addresses are shared.

To ease the interworking with the UPnP IGD AddPortMapping action, the base PCP specification includes a PREFER_FAILURE option, which avoids creating a mapping if the requested port is unavailable. A message flow of this behavior is shown in Figure 3.

In a *Dual-Stack Lite*[9] deployment, the home gateway is typically operated without a NAT function. In that configuration, the home gateway is expected to interwork between UPnP IGD (within the home) and PCP (toward the service provider's CGN). The PCP packets sent by the home gateway will have the source IP address of the home gateway, rather than the IP address of the host that initiated the UPnP IGD action. To accommodate that situation, the home gateway populates the THIRD_PARTY option with the IP address of the internal host needing the mapping. The THIRD_PARTY option is useful in other scenarios as well, including interworking with other protocols (such as the *NAT Port-Mapping Protocol* [NAT-PMP][13]) to PCP, using PCP to create mappings for a device that does not support PCP (for example, an IP-enabled webcam), or using it as the protocol between a web portal operated by the ISP and its CGN.

Figure 3: UPnP-to-PCP Interworking, Showing AddPortMapping Failure



Conclusion

PCP provides functions necessary for IPv6 hosts on home networks; it is a simple, scalable protocol that supports simple firewalling of IPv6 and IPv4 hosts, and to accommodate the transition to IPv6 also supports every conceived IPv4/IPv6 translation mechanism.

References

[1] Dan Wing, ed., Stuart Cheshire, Mohamed Boucadair, Reinaldo Penno, and Paul Selkirk, "Port Control Protocol (PCP)," Internet Draft, work in progress, July 2011, `draft-ietf-pcp-base`

[2] "UPnP Gateway committee: IGD:2 improvements over IGD:1," March 2009, `http://www.upnp.org/resources/documents/UPnPIGD2vsIGD1d10032009.pdf`

[3] James Woodyatt, ed., "Recommended Simple Security Capabilities in Customer Premises Equipment for Providing Residential IPv6 Internet Service," RFC 6092, January 2011.

[4] Mohamed Boucadair, Reinaldo Penno, Dan Wing, and Francis Dupont, "Universal Plug and Play (UPnP) Internet Gateway Device (IGD)-Port Control Protocol (PCP) Interworking Function," Internet Draft, work in progress, February 2011, **draft-bpw-pcp-upnp-igd-interworking**

[5] Reinaldo Penno, "PCP Support for Multi-Zone Environments," Internet Draft, work in progress, June 2011, **draft-penno-pcp-zones**

[6] Cathy Zhou, Tina Tsou, Xiaohong Deng, Mohamed Boucadair, and Qiong Sun, "Using PCP To Coordinate Between the CGN and Home Gateway Via Port Allocation," Internet Draft, work in progress, July 2011, **draft-tsou-pcp-natcoord**

[7] Stuart Cheshire, "PCP Rapid Recovery," Internet Draft, work in progress, June 2011, **draft-cheshire-pcp-recovery**

[8] Margaret Wasserman, Sam Hartman, and Dacheng Zhang, "Port Control Protocol (PCP) Authentication Mechanism," Internet Draft, work in progress, October 2011, **draft-wasserman-pcp-authentication**

[9] Alain Durand, Ralph Droms, James Woodyatt, and Yiu L. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion," RFC 6333, August 2011.

[10] Congxiao Bao, Christian Huitema, Marcelo Bagnulo, Mohamed Boucadair, and Xing Li, "IPv6 Addressing of IPv4/IPv6 Translators," RFC 6052, October 2010.

[11] Marcelo Bagnulo, Philip Matthews, and Iljitsch van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers," RFC 6146, April 2011.

[12] Margaret Wasserman and Fred Baker, "IPv6-to-IPv6 Network Prefix Translation," RFC 6296, June 2011.

[13] Stuart Cheshire, Marc Krochmal, and Kiren Sekar, "NAT Port Mapping Protocol (NAT-PMP)," Internet Draft, (expired), April 2008, **draft-cheshire-nat-pmp-03.txt**

DAN WING is the editor of the Port Control Protocol base specification and co-author of the PCP-UPnP interworking function specification. Dan has co-chaired the IETF's BEHAVE Working Group since 2006. He is a Distinguished Engineer at Cisco Systems, where he works on IPv6 transition technologies.
E-mail: **dwing@cisco.com**

Infrastructure Challenges to DNS Scaling

by Bill Manning

This article looks a few steps beyond the *Root Scaling Study* report from 2009.[1] In 2009, the *Internet Corporation for Assigned Names and Numbers* (ICANN) board commissioned a report to evaluate the effect of scaling the root zone from its current size to an undefined but larger root zone. Attributes considered were *Domain Name System Security Extensions* (DNSSEC), *Internet Protocol Version 6* (IPv6), *Internationalized Domain Names* (IDNs), and a larger number of entries in the zone. The report itself focused on the editorial processes and presentation of the finished root zone to the greater Internet. The report concluded that with prudence and with the addition of some "watch & warn" systems in place, the root zone could accommodate adding IPv6, DNSSEC, and IDNs along with other new *Top-Level Domain* (TLD) entries in a controlled manner. What the report did not consider was the effects of the deployed Internet infrastructure on the ability to get this new information into the rest of the *Domain Name System* (DNS) infrastructures of the Internet. Early experimental evidence[7, 8] suggests that the current state of infrastructure deployment will create problems for the deployment of these attributes.

Until recently the root zone of the DNS has enjoyed two important stabilizing properties:

- It is relatively small—currently the root zone holds delegation information for 280 generic, country-code, and special-purpose TLDs, and the size of the root zone file is roughly 80,000 bytes.
- It changes slowly—on average, the root zone absorbs less than one change per TLD per year, and the changes tend to be minor.

The root system has therefore evolved in an environment in which information about a small number of familiar TLDs remains stable for long periods of time. However, the type, amount, and volatility of the information that is contained in the root zone are expected to change as a result of the following four recent or pending policy decisions:

- Support for DNSSEC, or "signing the root"
- The addition of IDN TLDs
- Support for the additional larger addresses associated with IPv6
- The addition of new TLDs

These changes are placed in a backdrop of an infrastructure that is fundamentally changing, removing a third attribute of a stable DNS that was the presumption of a common transport protocol with well-defined constraints.

Core Design Principles

The DNS was designed so that queries and responses would have the greatest chance of survival and broadest reachability by using an IPv4 default *User Datagram Protocol* (UDP) packet size of 512 bytes for the initial bootstrapping. Larger packet sizes are supported and the *Transmission Control Protocol* (TCP) was defined and used as an alternate transport protocol—but expected to be infrequently used.

With these core principles intact, the DNS was able to successfully evolve into a highly decentralized dynamic system. The geographic and organizational decentralization of the root system arises from a deliberate design decision in favor of diversity and minimal fate-sharing coordination, which confers substantial stability and robustness benefits on the global Internet.

Simple quantitative extrapolation from a baseline model of the current DNS does not predict realistic future states of the system beyond the very short term, because:

- Each part of the system adapts in different ways to changes in the quantity, type, and update frequency of information, while also responding to changes in the rest of the Internet.
- These adaptations are not—and cannot be—effectively coordinated.
- For some, if not all, of the actors, nonquantifiable considerations dominate their individual adaptation behavior (both strategically, in a planning context, and tactically, in an operations context).

The risks associated with adding DNSSEC and IPv6 addresses to the DNS simultaneously change the basic assumption for DNS Query/Response reachability. Signing DNS data would, by itself, immediately increase the size of any zone by roughly a factor of 4 and increase the size of the response message[2]. The consequences of the second of these effects could be absorbed by replanning in order to recover lost headroom by adding bandwidth. Adding IPv6 addresses would in addition increase the size of any response. However, simply adding additional bandwidth may be insufficient when there are middleboxes, application layer gateways, or divergent transport options between the query path and the response path.

In these cases more information has to be carried in the packets that are returned in response to a query, meaning that the required amount of network bandwidth needed to support the operations of the server increases. As the DNS messages get bigger, they will no longer fit in single 512-byte packets forwarded by the UDP transport mechanism of the Internet. This situation will lead to clients being forced to resend their queries using UDP "jumbograms" or the TCP transport mechanism—a mechanism that has much more overhead and requires the end nodes to maintain much more state information. It also has much more overhead in terms of "extra packets" sent just to keep things on track. The benefit is, of course, that it can carry much larger pieces of information.

Moving the root system from its default UDP behavior to UDP "jumbograms" or TCP will not only have the undesirable effects mentioned previously, it will also affect the current trend of deploying servers using IP *anycast*[10]. Anycast works well with single packet transactions (such as UDP), but is much less well suited to handle TCP packet streams. If TCP transactions become more prevalent, the anycast architecture may require changes.

The point of view from the client side is worth mentioning. In certain client configurations, where firewalls are incorrectly configured[3], the following scenario can occur:

A resolver inside the misconfigured firewall receives a DNS request that it cannot satisfy locally. The query is sent to the root servers, usually over UDP, and a root server responds to this query with a referral, also over UDP. Today, this response fits nicely in 512 bytes. It is also true that for the past 6 years, the *Internet Systems Consortium* (ISC) has been anticipating DNSSEC and has shipped resolver code that, by default, requests DNSSEC data. After the root is signed, the response no longer fits into a 512-byte message. Estimates from the *National Institute of Standards and Technology* (NIST), using standard key lengths, indicate that DNSSEC will push the response to at least 2048 bytes or larger. This larger response will not be able to get past a misconfigured firewall that restricts DNS packets to 512 bytes, not recognizing the more modern extensions to the protocol that allow for bigger packets.

Upon not receiving the answer, the resolver on the inside will then retry the query, setting the buffer size to 512 bytes. The root will resend the response using smaller packets, but because it does not fit in a 512-byte packet, will fragment the response into a series of 512-byte replies, and the root server will set the "fragmented" and "truncated" flags in the packets, indicating to the resolver that the answer was fragmented and truncated, and encouraging the resolver to retry the query once more using TCP transport. The resolver will do so, and the root server will respond using TCP, but the misconfigured firewall also will reject DNS over TCP, because this transport has not been considered a normal or widely used transport for DNS queries.

In this worst case, a node will be unable to get DNS resolution after the root zone is signed, and the DNS traffic will triple, including one round in which TCP state must be maintained between the server and the resolver. There are of course ways around this problem, the most apparent ones being to configure the firewall correctly, or to configure the resolver to not ask for DNSSEC records.

Effect of IPv6 on Priming Queries

The basic DNS protocol specifies that clients, resolvers, and servers be capable of handling message sizes of at least 512 bytes. They may support larger message sizes, but are not required to do so.

The 512-byte "minimal maximum" was the original reason for having only nine root servers. In 1996 Bill Manning, Mark Kosters, and Paul Vixie presented a plan to Jon Postel to change the naming of the root name servers to take advantage of DNS label compression and allow the creation of four more authoritative name servers for the root zone. The outcome was the root name server convention as it stands today.

The use of 13 "letters" left a few unused bytes in the priming response, which were left there to allow for changes—which soon arrived. With the advent of IPv6 addressing for the root servers, it was no longer possible to include both an IPv4 "A" record and an IPv6 "AAAA" record for every root server in the priming response without truncation; AAAA records for only two servers could be included without exceeding the 512-byte limit. Fortunately the root system was able to rely on the practical circumstance that any node asking for IPv6 address information also supported *Extension Mechanisms for DNS* (EDNS0)[4].

DNSSEC also increases the size of the priming response, particularly because there are now more records in the Resource Record set and those records are larger. In [5] the authors make the following observation: "The resolver MAY choose to use DNSSEC OK[6], in which case it MUST announce and handle a message size of at least 1220 octets."

EDNS and MTU Considerations

The changes described will also affect other parts of the Internet, including (for example) end-system applications such as web browsers; intermediary "middleboxes" that perform traffic shaping, firewall, and caching functions; and *Internet Service Providers* (ISPs) that "manage" the DNS services provided to customers.

Although modern DNS server software defaults to using EDNS0, current measurement[7] collected from several of the RFC 1918[11] servers suggests that EDNS0 usage has not yet reached generally accepted levels of usefulness. Over the 12-month study, the ratio of ENDS0 queries received at these nodes remained at roughly 65 percent of the total queries received, with about 33 percent being non-EDNS queries. In the "other" camp are queries that set EDNS0 but then restrict packet sizes to 512 bytes. These queries cannot use the larger, negotiable *Maximum Transmission Unit* (MTU) sizes for larger UDP responses and therefore must use TCP to support larger responses. Some evidence suggests that with signed data, there is a pattern of retransmission of queries when responses larger than 512 bytes are generated and blocked. Such retransmissions can take as long as 7 seconds before timing out.

Lack of EDNS0 support in DNS caches suggests that many parts of the Internet will be constrained to using the traditional UDP sizes or will fall back to using TCP. Even where EDNS0 is indicated as being available, there are increased difficulties in knowing or negotiating a consistent *Path Maximum Transmission Unit* (Path MTU)[8].

The data supports an argument that the expectation of a useful UDP "jumbogram" or enough resources to manage hundreds of thousands or millions of TCP connections is unfounded because of historical expectations on "normal" DNS packet profiles. Clean, clear Internet paths that will allow larger packet sizes are rare, particularly when crossing the Internet. Locally, it is much more likely that larger packet sizes will be found and supported, raising the question for wide-scale deployment of IPv6 or DNSSEC because both attributes require larger packet sizes regardless of transport. If neither larger UDP packets nor TCP will be viable, what other choices are there?

Recent work inside the *Internet Engineering Task Force* (IETF) is exploring the use of the *Hypertext Transfer Protocol* (HTTP) as an alternative transport protocol for DNS messages.[9] It might be possible to augment the deployed DNS base to understand the addition of a third transport protocol.

The augmentation of the DNS protocol to support multiple transport protocols will require additional logic on the part of the servers to keep track of which transport a query was received on and select that transport when sending back the response. It will also require more complex logic to determine failover selection from one transport to another.

With the efforts going into making the infrastructure of the Internet IPv6-capable, it is possible that the underlying MTU problems may be corrected faster than adoption of a new transport protocol for the DNS. Certainly MTU problems have been considered for many years and for slightly different reasons[8] principally related to faster signaling rates and changes in the types of data being moved through the Internet. Regardless, this transition will take considerably more time than a simple DNS code refresh. Full support for larger packet sizes in the DNS will require changes in the equipment and code that comprise the baseline Internet infrastructure—and such changes may take decades.

References

[1] Jaap Akkerhuis, Lyman Chapin, Patrik Fältström, Glenn Kowack, Lars-Johan Liman, and Bill Manning, "Report on the Impact on the DNS Root System of Increasing the Size and Volatility of the Root Zone, Prepared by the Root Scaling Study Team," Version 1.0, September 2009.

[2] "DNSSEC and Its Impact on DNS Performance," 17 August 2009, `http://www.dnsops.gov/dnssec-perform.html`

[3] Ray Bellis and Lisa Phifer, "Test Report: DNSSEC Impact on Broadband Routers and Firewalls," SAC035, 16 September 2008, **http://www.icann.org/en/committees/security/ssac-documents.htm**

[4] Paul Vixie, "Extension Mechanisms for DNS (EDNS0)," RFC 2671, August 1999.

[5] Peter Koch and Matt Larson, "Initializing a DNS Resolver with Priming Queries, Internet Draft, expired, July 2008, **http://tools.ietf.org/id/draft-ietf-dnsop-resolver-priming-01.txt**

[6] Roy Arends, Rob Austein, Matt Larson, Dan Massey, and Scott Rose, "DNS Security Introduction and Requirements," RFC 4033, March 2005.

[7] EDNS Support: **http://www.ripe.net/data-tools/dns/as112/edns**

[8] Matt Mathis, "The Case for Raising the Internet MTU," July 2003, **http://staff.psc.edu/mathis/papers/Cisco200307/index.html**

[9] Mohan Parthasarathy and Paul Vixie, "Representing DNS Messages Using XML," Internet Draft, work in progress, September 2011, **http://www.ietf.org/id/draft-mohan-dns-query-xml-00.txt**

[10] Ted Hardie, "Distributing Authoritative Name Servers via Shared Unicast Addresses," RFC 3258, April 2002.

[11] Yakov Rekhter, Robert G Moskowitz, Daniel Karrenberg, Geert Jan de Groot, and Eliot Lear, "Address Allocation for Private Internets," RFC 1918, February 1996.

BILL MANNING has been in the network field since 1979, most recently with Booz Allen Hamilton. He has been an IETF Working Group chair, RFC author, and an ARIN Trustee, and he has been on numerous ICANN committees. He has worked as part of the teams that run Internet Root name servers, built the first Internet Exchange points, and worked on transitioning from NSFnet to commercial services. Current client work is focused on Internet Policy and Governance, Risk Analysis, and the future of naming systems. E-mail: **bmanning@sfc.keio.ac.jp**

Networking @ Home

by Geoff Huston, APNIC

One of the more interesting sessions at the *Internet Engineering Task Force* (IETF) meeting in Quebec City in July 2011 was the first meeting of the recently established *Homenet Working Group*[1]. What is so interesting about networking the home? Well, if you regard challenges as "interesting," then just about everything is interesting when you look at networking in the home!

It has been a very long time since the state of the art in home Internet involved plugging the serial port of the PC into the dialup modem. The *Asymmetric Digital Subscriber Line* (ADSL) modem, even when combined with some form of Wi-Fi base station, is looking distinctly passé these days. Today, the home network is seeing the intersection of a whole set of interests, including phone service, television service, home security services, energy management, utility service metering, other forms of home device monitoring, and, of course, connecting laptops and mobile devices to the net. The home network is not just a wired *Local-Area Network* (LAN), Wi-Fi home networks are commonplace, and there are also various Bluetooth devices. Maybe sometime soon it will be common for the home network to host some form of *Third-Generation* (3G) femtocell mobile cell phone repeater as well. But these days even that level of network complexity is not enough. Increasingly, the home office is part of the work office, and if numerous residents are at home, then the home network may be an endpoint for several corporate and institutional *Virtual Private Networks* (VPNs)[2].

Within the home network we want sophisticated security. This security involves not just protecting the network from the neighbors; the security requirements include the ability for individuals to partition off their work-VPN part of the home network from other home users. For resiliency we might want a second network provider, so we might want to add site-based multihoming to the mix. And we need to make all this work for both IPv4 and IPv6.

That set of requirements represents a massive agenda. But to make this situation truly challenging, we cannot expect every home to come with an IT Operational Service Manager to ensure that all the various devices you bring into the home and connect to the network function as required for the particular requirements of the home. Indeed, we cannot expect any home to be so lavishly supported, nor can we afford to support home networking with a bevy of specialized call centers with on-demand support specialists, expert in the panoply of consumer devices that are being sold today.

With today's home networks, consumers are effectively on their own; and all this equipment better just work straight out of the box. No configuration, no buttons, it just has to work!

Routing @ Home

The evolution of networking at home has progressed from a single computer to a basic LAN, and from there to an Ethernet-bridged network with numerous Wi-Fi and wired LAN segments. All these environments have a single common architecture with a single "boundary" unit that acts as a point of demarcation between the *Internet Service Provider* (ISP) and the home network. This unit is generally called *Customer Premises Equipment* (CPE), and typically encompasses the functions of a modem; an IPv4 *Network Address Translator* (NAT); a *Dynamic Host Configuration Protocol* (DHCP) server for both IPv4 and IPv6; as well as security firewall, bridge, and rudimentary router functions.

But it is unrealistic to assume that home networks will continue to use a centralized model that places all of the management functions of the home network in a single unit. So how should we view home networks? Should home networks be a single bridged LAN, or are we seeing the evolution of home networks into multiple distinct domains with a routing fabric to glue them together? And if that is the case, what routing protocol should be used?

I have noticed in the low end of the CPE market it is not uncommon to see a rudimentary routing function supported by the *Routing Information Protocol* (RIP)[3]. Thankfully, it is RIP Version 2, so the routing protocol can be configured with variable-length subnet masks, but even so, RIP is a very basic and simple routing protocol. But perhaps in this environment, that might be a positive factor rather than a liability in so far as RIP is simple enough to be autoconfigurable. On the other hand, if there is an emergent need for more complex functions, then maybe we need to look a little harder at the available options.

One of these more complex functions is *subnet management*. In IPv6, the CPE will collect an IPv6 address prefix. This process differs from the conventional IPv4 environment where the CPE is typically assigned a single IPv4 address. So the ensuing question is: Is it possible to automate the distribution of IPv6 subnets across the entire home network? What form of management protocol is appropriate for this role?

Of course the situation gets much more complicated if the home network has two (or more) service providers. In the IPv6 environment, this task becomes a challenging one, not only with the distribution of multiple subnets across the home network, but also in the matter of exit path selection. If the home network is exercising due diligence to prevent source address spoofing, it is also necessary for the home routing infrastructure to deliver an outgoing packet to the "right" exit ISP, where the source address of the outgoing packet needs to match the address prefix provided by the corresponding ISP service. In other words, there is a requirement for source address routing in the home.

This challenge was not really addressed by the *Site Multi-Homing by IPv6 Intermediation Working Group* (SHIM6)[4], despite the best of intentions, and it represents an even greater challenge if the intent is to provide mechanisms that can achieve such routing in an unmanaged home network environment.

I must admit to some concern here. We have managed to keep Internet routing working by using two principles. The first is to try to keep the routing task as simple as possible. Routing propagates a single "best" path to a destination. It does not necessarily do this propagation quickly, nor necessarily does it carry around with it a whole set of alternatives. It does just one job. The second principle is to admit that we have never really succeeded with the first principle of functional simplicity and we have always had expertise at hand to oversee the routing function and apply manual patches as required. The specialized requirements for the home network appear to be breaking both principles. The requirements are certainly not simple, and I see a mix of routing techniques—including various forms of policy-based routing requirements—entering the discussion. Secondly, there is no assurance that if things fail expertise is at hand to mend the failure. Indeed, the more complex the routing environment, the greater the potential for complex forms of failure. As we contemplate ever more complex requirements in the home network, we face a greater risk of encountering failure "by design," where it is just not possible to design products for this environment that will "just work."

Names @ Home

What should I call my printer? More to the point, how should I identify my Wi-Fi printer to all those devices at home that want to use it to print? I am sure that I would not like to use a proprietary naming scheme that requires me to add additional name resolution software to every device at home that wants to print something, nor do I want to transcribe IP addresses into everything. I would like my printer to get dynamically assigned IPv4 and IPv6 addresses when the device is plugged in and switched on, and have the name of the printer published via a generic name resolution mechanism, namely the *Domain Name System* (DNS).

But most of the time the rest of the world has no need to know the name of my printer at home, and I am not sure that it is a good move, securitywise, to gratuitously publish information in the public DNS. So what I would like for my printer is some form of "local" or "scoped" DNS, where I can name my printers, my disk servers, and other devices that I have at home in the context of my home and not have this information leak further afield. Is this scoped form of name resolution, split horizon DNS, or split views, possible in the context of the DNS without invoking further elements of configuration management?

Multicast DNS (mDNS) is perhaps one of the strongest candidates for this role. In essence, mDNS replaces the explicit client-server structure of the DNS with a scoped name subdomain of `.local` that is inherently scoped to the associated multicast domain.

This setup allows a client to perform DNS-like name resolution functions on a local network without the need to configure a conventional DNS server environment, and without the need to obtain global delegation of a site name in the global DNS.

An alternative approach is to use a conventional DNS delegation and conventional unicast DNS queries and responses. Clients are able to use DNS *Dynamic Updates*[5] to provide the local DNS server with their details as they come online. This approach requires either open access from anyone to the nameserver or a security mechanism such as *Transaction SIGnature* (TSIG)[6]. TSIG generally requires manual configuration, and alternatives are either little used—such as *Transaction KEY* (TKEY)[7]—or involve further intricacies, such as Microsoft's *Active Directory,* which uses other user authentication mechanisms to bootstrap the TSIG part using the *Generic Security Service Algorithm for Secret Key Transaction* (GSS-TSIG)[8]. The DNS server itself can be advertised to all clients via the *Simple Service Discovery Protocol* (SSDP), as part of the larger *Universal Plug and Play* (UPnP) framework.

Sensing and Serving @ Home

Where to go from here? It is certainly the case that electronics has managed to pervade just about every device at home. Electricity meters are morphing into household energy-management systems, and many other household appliances are now controlled by internal processors. But individually configuring each of these devices is a forbidding task. Even adding an interface to allow manual configuration can often be a challenging objective.

The objective here is to define a standard mechanism to allow sensors to sense their local environment when powered up, obtain an IP address, advertise their existence and capabilities to the network, and, as appropriate, rendezvous with the sensor controller or controllers across the home network.

This example is another instance of a more generic class of automating the installation and use of services in "lightly" managed or even unmanaged networks, and it intersects significantly with the objectives encompassed with SSDP and UPnP. The potential volume of such devices places this example more squarely into a class of IPv6-only services, I suspect, which is a significant extension to the existing IPv4-centric UPnP frameworks.

What is needed is a bootstrap protocol that can provide a connecting device with:

- Address configuration
- Routing setup
- Name management and name server discovery
- Discovery of other services and controllers
- Security capabilities

Security @ Home

One of the most significant concerns with home networks lies in the area of security management. Host computers in a home network often want to place a very high level of implicit trust in their immediate network neighbors at the same home. It is not unusual for hosts in a home network to share printers, file servers, data, and even user profiles. Indeed, it is probably commonplace. But beyond this local security domain a host should become paranoid and treat all connection attempts with suspicion. But where does the local trust domain start and stop? What is the "local" security boundary?

This question is difficult to answer in an automated fashion. It is no longer the local LAN, particularly as home networks transition into routed networks. The security boundary is related to the local multicast scope, but this supposition assumes that it is possible to define a multicast scope that encompasses the local trust domain of the home network, and this assumption brings us back to the same question.

Even if you thought you might have a clean answer to the boundary question, you need to remind yourself about telecommuting. With telecommuting, there is a requirement to partition out an entire local network segment from the rest of the home environment and the home security domain and transplant it into the work security domain.

Everything @ Home

Home is certainly the new field of engagement for networked goods and services. However, it is one of the most challenging places to operate in from the perspective of attempting to deliver coherent services in a reliable and secure manner. The components are sourced from various vendors, and constructed incrementally over extended periods of time. It is an environment where older components need to coexist with new devices, and the overall engineering of the environment is at best piecemeal, and perhaps more often not engineered at all. In this environment out-of-the-box interoperability is of paramount importance, and therefore it is an environment where good standards really matter. Perhaps unsurprisingly, given these constraints, networking in the home is one of the environments that appear to raise the most challenges. It is an unforgiving environment where there is no real substitute for simplicity and reliability in a "plug-and-play" world.

The IETF Homenet Working Group has a lot of work to do. The Working Group will have to examine the diverse set of approaches in use today, add IPv6 functions, and produce a coherent set of outcomes in the form of standards that support robust, capable home networks that work in an unmanaged environment.

Ahhh home! There really is no place quite like it!

References

[1] Homenet Working Group: **http://www.ietf.org/dyn/wg/charter/homenet-charter**

[2] Paul Ferguson and Geoff Huston, "What Is a VPN?" (Part One and Part 2), *The Internet Protocol Journal*, Volume 1, No. 1 and No. 2, June and September 1998.

[3] Gary Malkin, "RIP Version 2," RFC 2453, November 1998.

[4] Shim6 Working Group (concluded): **http://wiki.tools.ietf.org/wg/shim6/charters**

[5] Paul Vixie, ed., Yakov Rekhter, Susan Thomson, and Jim Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)," RFC 2136, December 1997.

[6] Paul Vixie, Olafur Gudmundsson, Donald E. Eastlake 3rd, and Brian Wellington, "Secret Key Transaction Authentication for DNS (TSIG)," RFC 2845, May 2000.

[7] Donald E. Eastlake 3rd, "Secret Key Establishment for DNS (TKEY RR)," RFC 2930, September 2000.

[8] Stuart Kwan, Praerit Garg, James Gilroy, Levon Esibov, Randy Hall, and Jeff Westhead, "Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)," RFC 3645, October 2003.

GEOFF HUSTON, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of numerous Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005; he served on the Board of Trustees of the Internet Society from 1992 until 2001.
E-mail: **gih@apnic.net**

IETF Tools—Making It Easier to Make the Internet Work Better

by Robert Sparks

Many activities are associated with defining and refining an *Internet Engineering Task Force* (IETF) protocol, and all of them are detail-oriented. As IETF Working Groups are formed, mailing list discussions proceed, documents are written and reviewed, and interoperability is evaluated, participants encounter tasks that can be significantly simplified with the help of software tools. Fortunately, those participants frequently are also skilled software developers, and they create and share these tools as the need arises. A new paradigm has evolved recently: When a pressing need for a tool is identified—particularly one that has a large scope—the *IETF Administrative Oversight Committee* (IAOC) accelerates the creation of the tool by working with the community to gather requirements and financing the development of a solution. Comprehensive lists of available tools are maintained at [1] and at [2]. This article introduces a few important tools and discusses how you can help improve them or develop new ones.

Document Tools

The *Extensible Markup Language to Request For Comments* (XML2RFC)[0] tool was developed to assist with Internet-Draft composition. Marshall Rose created and maintained the initial versions, capturing its input language and operation instructions in RFC 2629[18]. This tool simplifies draft creation and maintenance by automatically producing documents that satisfy the RFC Editor's layout requirements, and assists in including the appropriate boilerplate as defined by the *IETF Trust*. It also simplifies the task of the *RFC Production Center*[19, 20]. Starting with XML input rather than a draft in text form reduces the work required to create the RFC. The IAOC is currently funding a reimplementation of XML2RFC to reflect many years of user feedback, simplify maintenance—particularly of boilerplate handling—and make it easier for volunteers to contribute improvements. This reimplementation is currently available at [3]. Tony Hansen has been very active in gathering the requirements for and evaluating the reimplemented version. Julian Reschke also maintains *Extensible Stylesheet Language Transformations* (XSLT) code at [4] that translates RFC 2629-based input into several output formats.

After a new draft is prepared, Henrik Levkowetz' *Internet-Draft Nit Checker* (idnits) tool at [5] can scan it for any problems with the RFC Editor's checklist and guidelines and for other problems that drafts frequently encounter later in review. There are also tools for verifying sections of the document containing formal languages such as *Augmented Backus-Naur Form* (ABNF) or XML.

When an editor is satisfied that the document is ready to place in the repository, the automated *ID Submission tool*[6] assists with an easy upload. At any point two versions of a draft can be compared with *rfcdiff*[7], a flexible comparison program created by Henrik Levkowetz.

As a draft progresses, its history and current status can be tracked using the *Internet-Drafts Tracker* (ID Tracker) tool[8]. This tool provides powerful search capabilities into the entire Internet-Draft repository, and a comprehensive view into the lifecycle of each Internet-Draft. With its roots in a tool to help the *Internet Engineering Steering Group* (IESG) keep track of drafts in IESG evaluation, the ID Tracker has evolved into a portal touching almost all aspects of IETF work. Each step of that evolution has improved efficiency and transparency, and has simplified access to the history of the development of each document.

Recent additions to the tracker allow for an easier capture of the details of Working Group processing. *Work in progress* will provide more visibility into the Working Group chartering and rechartering processes. The tracker is also used by other document streams. Many of the enhancements to the tracker are informed by the views into documents and Working Groups maintained by Henrik Levkowetz at [2]. The tracker continues to evolve through both IAOC-funded development efforts and volunteer contributions. An extension in progress will add visibility into the RFC Editor and *Internet Assigned Numbers Authority* (IANA) actions. When this extension is done the entire lifecycle of a Draft, from -00 submission to RFC publication, can be viewed in a single place.

Working Group and Meeting Tools

At each IETF meeting, a participant can build a custom view of the agenda using the tools at the datatracker and the tools sites. For example, [9] renders an interactive JavaScript-based calendar contributed by Adam Roach showing the *Real-Time Applications and Infrastructure* (RAI) meetings at IETF82. The pages at [10] provide a quick reference to the jabber rooms and audio streams of each Working Group meeting. The meeting materials tool facilitates uploading of agendas, slides, and minutes, which become available immediately through the agenda views.

Each Working Group has a Subversion Repository and an integrated instance of Trac[21] at its disposal. The Subversion Repository can be used to maintain Working Group draft source, versioned instances of test documents, and even implementation code. IETF-specific customizations of the Trac system are described at [11]. Many Working Groups are already taking advantage of what the wiki Trac provides, and are using its ticketing feature to effectively track major Working Group document problems.

Notable examples are the problem tracking integrated into the *Hypertext Transfer Protocol Bis* (HTTPBIS) document status page at [12], and the summary of DISPATCH activity at [13]. The Trac wiki capability is also used by the Working Group Chairs at [14] and the IESG at [15].

IETF News

Keeping up with all of the activity across the IETF can be a challenge. One of the better tools for seeing what is happening is *The Daily Dose of the IETF*, created by Pasi Eronen, available at [16].

Again, this article is an introduction to just a few important tools. Comprehensive lists of available tools are maintained at [1] and [2].

Many of these tools were created because a person who needed them coded an initial version and contributed it to the community. Volunteers (and when needed, IAOC-funded efforts) then improve these tools over time. For several years, a group of volunteers have been meeting the Saturday before each IETF meeting for a day-long *Code Sprint*. If the existing tools need a minor tweak to make things work much better for you, or if you have an idea for a new tool you would like to start, please consider participating at the next Code Sprint. Between sprints, you can still help with the code. Refer to the sprint pages for an upcoming or recent sprint such as [17] and for information about getting started.

Whether or not you can contribute to the code, please discuss your ideas on the **tools-discuss@ietf.org** mailing list.

Several tool contributers have already been mentioned. Henrik Levkowetz deserves to be mentioned again. His herculean efforts maintaining **tools.ietf.org** and creating many of the tools there are of great benefit to the community.

References

[0] Marshall T. Rose and Carl Malamud, "Writing Internet Drafts and RFCs Using XML," *The Internet Protocol Journal*, Volume 10, No. 1, March 2007.

[1] **http://www.ietf.org/tools**

[2] **http://tools.ietf.org/**

[3] **http://xml.resource.org/**

[4] **http://greenbytes.de/tech/webdav/rfc2629xslt/rfc2629xslt.html**

[5] **http://tools.ietf.org/tools/idnits/**

[6] **https://datatracker.ietf.org/submit/**

[7] **http://www.ietf.org/tools/rfcdiff/**

[8] **http://datatracker.ietf.org/**

[9] **https://datatracker.ietf.org/meeting/82/agenda.html#RAI**

[10] **http://tools.ietf.org/agenda/82/**

[11] **http://trac.tools.ietf.org/misc/venue/wiki/IetfSpecificFeatures**

[12] **http://tools.ietf.org/wg/httpbis/**

[13] **http://trac.tools.ietf.org/wg/dispatch/trac/wiki**

[14] **http://wiki.tools.ietf.org/group/wgchairs/**

[15] **http://trac.tools.ietf.org/group/iesg/trac/wiki**

[16] **http://tools.ietf.org/dailydose/**

[17] **http://trac.tools.ietf.org/tools/ietfdb/wiki/IETF82Sprint**

[18] Marshall T. Rose, "Writing I-Ds and RFCs using XML," RFC 2629, June 1999.

[19] Leslie Daigle, "RFC Editor in Transition: Past, Present, and Future," *The Internet Protocol Journal,* Volume 13, No. 1, March 2010.

[20] RFC Editor, "40 Years of RFCs," RFC 5540, April 2009.

[21] **http://trac.edgewall.org/about**

ROBERT SPARKS is an Area Director for the Real-Time Applications and Infrastructure Area (RAI) in the IETF. He previously chaired the IETF's SIMPLE Working Group, which defines extensions to SIP for Presence and Instant Messaging, and the GEORPIV Working Group, which provides tools for applications to carry geographic location information and privacy rules to affect its use. Robert is a co-editor of the core SIP standard (RFC 3261), and several important SIP updates and extensions. He coordinates the premier real-time communications interoperability event, the SIPit. Robert is a Principal Software Engineer at Tekelec, and has held management and research positions at Estacado Systems, Xten (now Counterpath), dynamicsoft, Lucent, MCI Worldcom, and Texas A&M University. Robert holds a Master's degree in Mathematics and a Bachelor's degree in Computer Science from Texas A&M University. E-mail: **rjsparks@nostrum.com**

Fragments

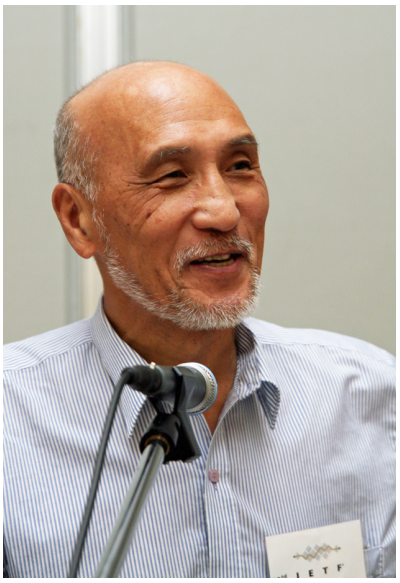


Photo: Peter Löthberg

Professor Kilnam Chon Receives 2011 Postel Service Award

The *Internet Society* (ISOC) recently announced that its prestigious *Jonathan B. Postel Service Award* was presented to leading technologist Professor Kilnam Chon for his significant contributions in the development and advancement of the Internet in Asia.

Professor Chon contributed to the Internet's growth in Asia through his extensive work in advancing Internet initiatives, research, and development. In addition, his pioneering work inspired many others to promote the Internet's further growth in the region. The international award committee, comprised of former Jonathan B. Postel award winners, noted that Professor Chon was active in connecting Asia, and that his efforts continue today in the advancement of the Internet in other regions.

The Postel Award was established by the Internet Society to honour individuals or organisations that, like Jon Postel, have made outstanding contributions in service to the data communications community.

Lynn St. Amour, President and CEO of ISOC, commented, "I met Professor Chon nearly fifteen years ago. He has long been a pioneer in the advancement of the Internet, striving to ensure its robust development. Beyond the amazing breadth of Professor Chon's work, perhaps his most remarkable achievement is his ability to inspire others. As a result of his work and the efforts of those he has motivated, Kilnam Chon has helped to ensure the global Internet is truly for everyone."

ISOC presented the award, including a US$20,000 honorarium and a crystal engraved globe, during the 82nd meeting of the *Internet Engineering Task Force* (IETF) in Taipei, November 13–18, 2011.

The Internet Society is the world's trusted independent source of leadership for Internet policy, technology standards and future development. Based on its principled vision and substantial technological foundation, ISOC works with its members and Chapters around the world to promote the continued evolution and growth of the open Internet through dialog among companies, governments, and other organizations around the world. For more information about the Postel Service Award see: `http://www.isoc.org/postel/`

Alexandre Cassen and Rémi Després Receives 2011 Itojun Service Award

The third *Itojun Service Award* was presented to Alexandre Cassen and Rémi Després at the *Internet Engineering Task Force* (IETF) meeting held in Taipei, Taiwan in November 2011. The awardees were recognized for their design and implementation of "6rd," an IETF protocol that aims to speed the transition to global deployment of IPv6, which is critical to ensuring the continued growth and evolution of the Internet.

The 6rd protocol has been implemented by several *Internet Service Providers* (ISPs) around the world, including *Free Telecom*—the second largest ISP in France—as part of their efforts to deploy IPv6.

First awarded in 2009, the Itojun Service Award honors the memory of Dr. Jun-ichiro "Itojun" Hagino, who passed away in 2007 at the age of 37. The award, established by the friends of Itojun and administered by the *Internet Society* (ISOC), recognizes and commemorates the extraordinary dedication exercised by Itojun over the course of IPv6 development.

"Alexandre and Rémi's efforts have helped to quickly bring a real IPv6 experience to hundreds of thousands of Internet users, demonstrating that IPv6 deployment can be effectively implemented on a large scale by commercial network providers," said Jun Murai of the Itojun Service Award committee and founder of the WIDE Project. "On behalf of the Itojun Service Award committee, I am extremely pleased to present this award to Alexandre and Rémi for the significant work they have done to advance IPv6 development and deployment."

The Itojun Service Award is focused on pragmatic contributions to developing and deploying IPv6 in the spirit of serving the Internet. The award, presented annually, includes a presentation crystal, a US$3,000 honorarium and a travel grant.

Alexandre Cassen said, "It is truly an honor to have been selected to receive the Itojun Service Award. As a software developer myself, It is particularly touching to receive an award created in the memory of a coding legend such as Itojun. I would also like to thank the entire team at Free Telecom who, in 2007, implemented and deployed 6rd, allowing any subscriber who asked for IPv6 to have it with a single click. As I write this, Free Telecom has more than 1,500,000 subscribers using IPv6 every day, and all new subscribers have IPv6 enabled by default. IPv6 is happening Itojun!"

Rémi Després said, "The Itojun Award is the best possible recognition that long efforts to make IPv6 deployment practicable have been useful to the Internet community. Latecomer in IPv6 standardization, I was about to send my first email to Itojun on a technical issue when I heard of his death. I was even sadder since we undoubtedly would have otherwise enjoyed sharing our ideas and our enthusiasm. Sharing the honor of this award with Alexandre Cassen perfectly illustrates the great progress possible when a dynamic network operator with a pioneer spirit and talented engineers adopts an innovative and simple design. Making IPv6 operational on a large scale in only five weeks will be remembered as a milestone of both of our professional lives."

More information on the Itojun Service Award is available at:
http://www.isoc.org/itojun

Internet Society Joins Opposition to Stop Online Piracy Act

The Internet Society Board of Trustees has expressed concern with a number of U.S. legislative proposals that would mandate *Domain Name System* (DNS) blocking and filtering by *Internet Service Providers* (ISPs) to protect the interests of copyright holders. While the Internet Society agrees that combating illicit online activity is an important public policy objective, these critical issues must be addressed in ways that do not undermine the viability of the Internet as a platform for innovation across all industries by compromising its global architecture. The Internet Society Board of Trustees does not believe that the *Protect-IP Act* (PIPA) and *Stop Online Piracy Act* (SOPA) are consistent with these basic principles.

Specifically, the Internet Society is concerned with provisions in both bills regarding DNS filtering. DNS filtering is often proposed as a way to block illegal content consumption by end users. Yet policies to mandate DNS filtering will be ineffective for that purpose and will interfere with cross-border data flows and services undermining innovation and social development across the globe.

Filtering DNS or blocking domain names does not remove the illegal content—it simply makes the content harder to find. Those who are determined to download filtered content can easily use a number of widely available, legitimately-purposed tools to circumvent DNS filtering regimes. As a result, DNS filtering encourages the creation of alternative, non-standard DNS systems.

From a security perspective, DNS filtering is incompatible with an important security technology called *Domain Name System Security Extensions* (DNSSEC). In fact, DNSSEC would be weakened by these proposals. This means that the DNS filtering proposals in SOPA and PIPA could ultimately reduce global Internet security, introduce new vulnerabilities, and put individual users at risk.

Most worrisome, DNS filtering and blocking raises human rights and freedom of expression concerns, and often curtails international principles of rule of law and due process. Some countries have used DNS filtering and blocking as a way to restrict access to the global Internet and to curb free expression.

The United States has been a strong proponent of online Internet freedoms and therefore has an important responsibility to balance local responsibilities and global impact, especially with respect to Internet policy. Given this commitment to global Internet freedom, it would be harmful to the global Internet if the United States were to implement such an approach.

"The Internet Society Board of Trustees is deeply concerned about the ramifications of the PIPA and SOPA bills on the overall stability and interoperability of the Internet," said Raul Echeberria, Chairman of the Internet Society Board of Trustees.

"The Board recognizes that there can be misuses of the Internet; however, these are greatly outweighed by the positive uses and benefits of the Internet. We believe the negative impact of using solutions such as DNS blocking and filtering to address these misuses, far outweighs any short-term legal or business benefits."

"The Internet Society believes that sustained, global collaboration amongst all parties is needed to find ways that protect the global architecture of the Internet while combating illicit online activities," said Internet Society President and CEO Lynn St. Amour. "Mandating DNS blocking and filtering is simply not a viable option for the future of the Internet. We must all work together to support the principles of innovation and freedom of expression upon which the Internet was founded."

For more details on DNS Filtering, visit:
`http://www.isoc.org/internet/issues/dns.shtml`

See also:
`https://www.eff.org/deeplinks/2011/12/internet-inventors-warn-against-sopa-and-pipa`

APNIC and JPRS Collaborate to Translate DNSSEC Technology Experiment Report

The *Asia Pacific Network Information Centre* (APNIC) has collaborated with *Japan Registry Services* (JPRS) to translate from Japanese into English the documents "DNSSEC Technology Experiment Report – Verification of Functionality and Performance" and "DNSSEC Technology Experiment Report – Operational Design."

These documents contain the latest information on *Domain Name System Security Extensions* (DNSSEC) implementation, and provides information to those interested in implementing it. These reports are designed to introduce case studies to share knowledge and results gained through experiments conducted in 2010 that JPRS carried out in cooperation with Japanese ISPs, equipment vendors, and hosting providers.

APNIC would like to thank JPRS's great initiative and all those involved in the process for making such an important contribution to DNSSEC awareness. APNIC also appreciates JPRS for making the documents available in English for wider distribution. The reports are available for download from:

`http://jprs.jp/dnssec/doc/DNSSEC-testbed-report-fpv1.0-E.pdf`

and

`http://jprs.jp/dnssec/doc/DNSSEC-testbed-report-odv1.0-E.pdf`

RFC Series Editor Appointment

The *Internet Architecture Board* (IAB) is pleased to announce the appointment of Heather Flanagan as the *Request For Comments Series Editor* (RSE). Ms. Flanagan will assume the responsibilities from the Acting RSE, Olaf Kolkman, and begin her tenure on January 1, 2012. The contract negotiated by the *IETF Administrative Oversight Committee* (IAOC) includes an initial term of two years and a presumptive renewal of two years.

Ms. Flanagan was selected by the *RFC Series Oversight Committee* (RSOC) based upon her experience, education, skills and energy she will bring to the position.

Ms. Flanagan is currently the Project Coordinator for the *COmanage* project, an effort funded by a grant from the *National Science Foundation* (NSF) and Internet2 to create a collaboration management platform, prior to that she was Director of Systems Administration, IT Services at Stanford University in Palo Alto, California. Her technical background is complemented by a Masters of Science of Library Science from the University of North Carolina, Chapel Hill that will prove invaluable in the accessing and indexing of RFCs.

Ms. Flanagan brings a high degree of energy and enthusiasm to the position. Her interpersonal skills as a facilitator and good listener will enable her to work well with the capable staff at the RFC Production Center and with the community in reaching consensus on a variety of issues facing the RFC Series.

The RSOC selection followed a lengthy process that included announcing the position inside and outside the community, several rounds of interviews, reference checks, and face-to-face interviews in Taipei at IETF 82. More than thirty-five applications were received, two-thirds of which were from outside the community.

We express our congratulations to Ms. Flanagan. We also want to extend our thanks to Ray Pelletier and the RSOC chaired by Fred Baker for their role in bringing the RSE selection process to a successful conclusion; to Olaf Kolkman for his service to the community as Acting RSE; to Joel Halpern for his ongoing work as editor of the "RFC Editor Model v2" document; and to the RFC Production Center for its customary diligence in the editing and publishing of RFCs this year, likely the second most productive in RFC publication history.

We look forward to working with the new RSE; we wish her well; and know that the community will work with Heather for the betterment of the RFC Series.

—For the IAB
Bernard Aboba, IAB Chair

2011 Global IPv6 Survey Results

On October 20, 2011 the *Number Resource Organization* (NRO) announced the publication of the "Global IPv6 Deployment Monitoring Survey 2011 Results," initially previewed at the *Internet Governance Forum* (IGF) in Nairobi, Kenya, in September.

The findings from the survey drew on data supplied by around 1,600 international respondents, over 350 of which were from the *American Registry for Internet Numbers* (ARIN) region. On behalf of ARIN and GNKS Consulting, we would like to thank all who participated in the survey. Your feedback is crucial to expanding the understanding of where this community is moving, and what can be done to ensure readiness for the widespread adoption of IPv6. We hope you will take this opportunity to review the results at: **http://www.nro.net/wp-content/uploads/ipv6_deployment_survey.pdf**

The Public Switched Telephone Network in Transition

The United States *Federal Communications Commission* (FCC) recently held two workshops to examine the transition from the *Public Switched Telephone Network* (PSTN) to new technologies. Circuit-switched wireline voice technology has created a high standard for reliability, accessibility, and ubiquity. Consumers will continue to expect and demand these qualities, even as they shift from PSTN services to services provided over different networks. The transition away from the PSTN is already occurring, and is likely to accelerate. Through these workshops, the Commission will seek input on the technical, economic, and policy issues that must be addressed to minimize disruption during this transition, and to protect consumers, public safety, competition, and other important interests. For more information, visit: **http://www.fcc.gov/events/public-switched-telephone-network-transition-0**

Upcoming Events

The *North American Network Operators' Group* (NANOG) will meet in San Diegeo, California, February 5–8, 2012. For more information see: **http://nanog.org**

The *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will meet in New Delhi, India, February 21–March 2, 2012. For more information see:
http://www.apricot2012.net/

The *Internet Engineering Task Force* (IETF) will meet in Paris, France, March 25–30, 2012. For more information see:
http://www.ietf.org/meeting/

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in San Jose, Costa Rica, March 11–16, 2012 and in Prague, Czech Republic, June 24–29, 2012. For more information, see: **http://icann.org/**

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

March 2012 Volume 15, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

Internet security continues to receive much attention both in the media and within the *Internet Engineering Task Force* (IETF) and similar organizations that develop technical solutions and standards. Last September, someone managed to break into a trusted *Certification Authority*'s system and subsequently produced numerous fake *digital certificates,* files that comprise part of the architecture for what is generally referred to as "browser security." In our first article, Geoff Huston describes what happened, the implications of this form of attack on the security of web-based services on the Internet, and what can be done to prevent similar attacks in the future.

In our second article, Richard Barnes describes the work of the *DNS-based Authentication of Named Entities* (DANE) working group in the IETF and explains how DANE, when deployed, can help prevent the sort of attack that is described in our first article.

This year I am celebrating 25 years in Internet technical publishing. Prior to launching *The Internet Protocol Journal* (IPJ), I was the editor of *ConneXions—The Interoperability Report,* published from 1987 until 1997 by Interop Company. With the generous support of *The Charles Babbage Institute* at the University of Minnesota, *ConneXions,* which was a paper-only publication, has been scanned and made available online. To mark the 25 combined years of *ConneXions* and IPJ, we asked Geoff Huston to examine the state of computer communications 25 years ago and give us his thoughts on where we have been and where we might be going in this rapidly developing technology landscape.

Please remember to check your subscription expiration date and take the necessary steps if you wish to continue receiving this journal. You will need your subscription ID and the e-mail address you used when you subscribed in order to access your record and renew online. Visit the IPJ website at **www.cisco.com/ipj** and click on the "Subscriber Services" link to get to the login page. The system will send you a URL that allows direct access to your record. If you no longer have access to the e-mail you used when you subscribed or you have forgotten your subscription ID, just send a message to **ipj@cisco.com** and we will make the necessary changes for you.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Hacking Away at Internet Security

by Geoff Huston, APNIC

The front page story of the September 13, 2011, issue of the *International Herald Tribune* said it all: "Iranian activists feel the chill as hacker taps into e-mails." The news story relates how a hacker has "... sneaked into the computer systems of a security firm on the outskirts of Amsterdam" and then "... created credentials that could allow someone to spy on Internet connections that appeared to be secure." According to this news report, the incident punched a hole in an online security mechanism that is trusted by hundreds of millions of Internet users all over the network.

Other news stories took this hyperbole about digital crime and tapping into e-mail conversations on the Internet to new heights, such as *The Guardian's* report on September 5, 2011, which claimed that the "... DigiNotar SSL certificate hack amounts to cyberwar, says expert."[1]

If application-level security is so vulnerable to attack, then this incident surely calls into question the basic mechanisms of trust and security upon which the entire global Internet has been constructed. By implication it also calls into question the trustworthiness of services operated by the major global Internet brands such as Google and Facebook, as much as it raises doubts about the levels of vulnerability for the use of online services such as banking and commercial transactions.

Just how serious is this problem? Are we now at the end of civilization as we know it?

Well, hardly!

Is digital cryptography now broken? Has someone finally managed to devise a computationally viable algorithm to perform prime factorization of massively large numbers, which lies at the heart of much of the cryptography used in the Internet today?

I really don't think so. (At the very least, if someone has managed to achieve this goal, then that person is staying very quiet about it.).

Does this situation represent a systematic failure of security? Do we need to rethink the entire framework of cryptography and security in the Internet?

Not this time.

As far as I can tell, there has been no dramatic failure in the integrity of the digital technology used for security in the Internet today. Yes, some were surprised by this failure, including the Netherlands government, which uses certificates issued by the compromised certification authority, DigiNotar (`http://www.diginotar.com`) as part of its online service infrastructure. But the hacking incident was not based on a successful direct attack on the technology of cryptography by itself, and there is no reason to suppose that the strength of today's encryption algorithms is any weaker today than yesterday.

But in observing that the basic technology tools of the Internet security framework are still operating within acceptable bounds of integrity, and observing that this hacking attack did not create a gaping hole in our commitment to digital cryptography, what cannot be claimed is that the use of these cryptographic tools in today's Internet service environment is similarly trustworthy. The hacking attempt apparently was successful in so far as it provided the capability for third parties to impersonate trusted services and thereby capture users' private data, and evidently some people did indeed do precisely that, and that is not good at all.

Let's look a little more closely at this hacking episode and examine the way in which security is applied to the world of web browsing and the manner in which the vulnerabilities in this security framework were evidently exploited.

Securing a Connection

When I point my browser at my online banking service—or at any other secure website for that matter—a part of the browser navigation bar probably glows a reassuring green, and when I click it I get the message that I am connected to a website run by the Acme Banking corporation, and that my connection to this website has been encrypted to prevent eavesdropping. However, the website *certificate* was issued by some company that I have never even heard of. When I ask for more information, I am told the domain name, the company to whom the certificate for this domain name was issued, the identity of the certificate issuer, and the public key value. I am also reassuringly informed that the message I am viewing was encrypted before being transmitted over the Internet, and that this encryption makes it very difficult for unauthorized people to view information travelling between computers, and it is therefore very unlikely that anyone could read this page as it passes through the network. All very reassuring, and for the most part true, to the extent that we understand the strength of cryptographic algorithms in use today. The connection is using a *Transport-Layer Security* (TLS)[2] connection and the traffic is encrypted using a private session key that should be impenetrable to all potential eavesdroppers.

But that is not the entire truth, unfortunately.

It may well be that your conversation is secure against eavesdropping, but it is only as secure as the ability of the other party to keep its private key a secret. If the other side of the conversation were to openly broadcast the value of its private key, then the entire encryption exercise is somewhat useless. So, obviously, my local bank will go to great lengths to keep its private key value a secret, and I rely on its efforts in order to protect my conversations with the bank.

But even then it is not quite the full story.

Am I really talking to my bank? Or in more general terms, am I really talking to the party with whom I wanted to talk?

The critical weakness in this entire framework of security is that the binding of certificates and keys to *Domain Name System* (DNS) names is not an intrinsic part of the DNS itself. It is not an extension of *Domain Name System Security Extensions* (DNSSEC)[3, 4]. It has been implemented as an add-on module where third parties generate certificates that attest that someone has a particular domain name. Oddly enough, these *Certification Authorities* (CAs) may never have actually issued that particular domain name, because they are often disconnected from the DNS name registration business. Their business is a separate business activity where, after you have paid your money to a domain name registrar and secured your domain name, you then head to a domain name Certification Authority and pay them money (commonly they charge more money than the name registration itself) and receive a domain name certificate.

Certification Authorities

Who gets to be a Certification Authority? Who gets to say who has which domain name and what keys should be associated with that domain name?

Oddly enough the answer is, at a first level of approximation, just about anyone who wants to! I could issue a certificate to state that you have the domain name **`www.example.com`** and that your public key value is some number. The certificate I issue to that effect would not be much different from the certificates issued by everyone else. Yes, my name would be listed as the certificate issuer, but that is about all in terms of the difference between this certificate and the set of certificates you already trust through your browser.

So what is stopping everyone from being a Certification Authority? What is preventing this system from descending into a chaotic environment with thousands of certificate issuers?

For this situation the browser software folks (and other application developers of secure services) have developed a solution. In practice it requires a lot of effort, capability, diligence, and needless to say, some money, to convince a browser to add your Certification Authority public key to its list of trusted Certification Authorities.

You have to convince the browser developers that you are consistently diligent in ensuring that you issue certificates only to the "correct" holders of domain names and that you undertake certificate management practices to the specified level of integrity and trust. In other words, you have to demonstrate that you are trustworthy and perform your role with consistent integrity at all times. You then get listed with all the other trusted Certification Authorities in the browser, and users will implicitly trust the certificates you issue as part of the security framework of the Internet.

How many trusted Certification Authorities are there? How many entities have managed to convince browser manufacturers that they are eminently trustable people? If you are thinking that this role is a special one that only a very select and suitably *small* number of folks who merit such absolute levels of trust should undertake for the global Internet—maybe two or three such people—then, sadly, you are very much mistaken.

Look at your browser in the preferences area for your list of trusted Certification Authorities, and keep your finger near the scroll button, because you will have to scroll through numerous such entities. My browser contains around 80 such entities, including one government ("Japanese Government"), a PC manufacturer ("Dell Inc"), numerous telcos, and a few dedicated certificate issuers, including DigiNotar.

Do I know all these folks that I am meant to trust? Of course not! Can I tell if any of these organizations are issuing rogue certificates, deliberately—or far more likely—inadvertently? Of course not!

The structural weakness in this system is that a client does not know *which* Certification Authority—or even which duly delegated subordinate entity of a Certification Authority—was used to issue the "genuine" DNS certificate. When a client receives a certificate as part of the TLS initialization process, then as long as any one of the listed trusted Certification Authorities is able to validate the presented certificate, even if it is the "wrong" Certification Authority, then the client will proceed with the session with the assumption that the session is being set up with the genuine destination.

In other words the entire certification setup is only as strong—or as weak—as the weakest of the certification authorities. It really does not matter to the system as a whole if any single Certification Authority is "better" at its task than the others, because every certified domain name is protected only to the extent that the "weakest" or most vulnerable trusted Certification Authority is capable of resisting malicious attack and subversion of its function. Indeed, one could argue that there is scant motivation for any trusted Certification Authority to spend significantly more money to be "better" than the others, given that its clients are still as vulnerable as all the other clients of all the other Certification Authorities.

In other words, there is no overt motivation for market differentiation based on functional excellence, so all certificates are only as strong as the weakest of all the Certification Authorities. And therein lies the seed of this particular hacking episode.

The Hack

The hack itself now appears to have been just another instance of an online break-in to a web server. The web server in question was evidently running the service platform for DigiNotar, and the hacker was able to mint some 344 fraudulent certificates, where the subject of the certificate was valid, but the public key was created by the hacker. A full report of the hacking incident was published by Fox-IT[5].

To use these fraudulent certificates in an attack requires a little more than just minting fraudulent certificates. It requires traffic to be redirected to a rogue website that impersonates the webpage that is under attack. This redirection requires collusion with a service provider to redirect client traffic to the rogue site, or a second attack, this time on the Internet routing system, in order to perform the traffic redirection.

So minting the fraudulent certificates is just one part of the attack. Were these fake certificates used to lure victims to fake websites and eavesdrop on conversations between web servers and their clients? Let's look at the client's validation process to see if we can answer this question.

When starting a TLS session, the server presents the client with a certificate that contains the server public key. The client is expected to validate this certificate against the client's locally held set of public keys that are associated with trusted certification authorities. Here is the first vulnerability. The client is looking for any locally cached trusted key to validate this certificate. The client is not looking as to whether a particular public key validates this certificate. Let's say that I have a valid certificate issued by the Trusted Certification Authority Inc. for my domain name, **`www.example.com`**. Let's also say that the server belonging to another Certification Authority, Acme Inc, is compromised, and a fake certificate is minted. If a user is misdirected to a fake instance of **`www.example.com`** and the bad server passes the client this fake certificate, the client will accept this fake certificate as valid because the client has no *presumptive* knowledge that the only key that should validate a certificate for **`www.example.com`** belongs to the Trusted Certification Authority Inc. When the key belonging to Acme Inc validates this certificate and ACME is a trusted entity according to my browser, then that is good enough to proceed.

Actually that is not the full story. What if I wanted to cancel a certificate? How do certificates get removed from the system and how do clients know to discard them as invalid?

A diligent client (and one who may need to check a box in the browser preference pane to include this function) uses a second test for validity of a presented certificate, namely the *Online Certificate Status Protocol* (OCSP)[6]. Clients use this protocol to see if an issued certificate has been subsequently revoked. So after the certificate has been validated against the locally held public key, a diligent application will then establish a secure connection to the certification authority OCSP server and query the status of the certificate.

This secure connection allows for prompt removal of fraudulent certificates from circulation. It assumes of course that clients use OCSP diligently and that the Certification Authority OCSP server has not also been compromised in an attack, but in an imperfect world this step constitutes at least another measure of relative defence.

The OCSP server logs can also provide an indication of whether the fraudulent certificates have been used by impersonating servers, because if the certificate was presented to the client and the client passed it to an OCSP server for validation, then there is a record of use of the certificate. The Fox-IT report contains an interesting graphic that shows the geolocation of the source addresses of clients who passed a bad `*.google.com` certificate to OCSP for validation. The source addresses have a strong correlation to a national geolocation of Iran.

Obviously this attack requires some considerable sophistication and capability, hence the suspicion that the attack may have had some form of state or quasi-state sponsorship, and hence the headlines from *The Guardian,* quoted at the start of this article, that described this attack as an incident of cyberwarfare of one form or another. Whether this incident was a cyber attack launched by one nation state upon another, or whether this was an attack by a national agency on its own citizens is not completely clear, but the available evidence points strongly to the latter supposition.

Plugging the Hole?

This incident is not the first such incident that has created a hole in the security framework of the Internet, and it is my confident guess that it will not be the last. It is also a reasonable guess that the evolution of the sophistication and capability that lie behind these attacks points to a level of resourcing that leads some to the view that various state-sponsored entities may be getting involved in these activities in one way or another.

Can we fix this?

It seems to me that the critical weakness that was exploited here was the level of disconnection between domain name registration and certificate issuance. The holders of the domain names were unaware that fraudulent certificates had been minted and were being presented to users as if they were the real thing. And the users had no additional way of checking the validity of the certificate by referring back to information contained in the DNS that was placed there by the domain name holder.

The end user was unable to refine the search for a trusted Certification Authority that would validate the presented certificate from all locally cached trusted Certification Authorities to the one certification authority that was actually used by the domain name holder to certify the public key value. So is it possible to communicate this additional information to the user in a reliable and robust manner?

The last few years have seen the effort to secure the DNS gather momentum. The root of the DNS is now DNSSEC-signed, and attention is now being focused on extending the interlocking signature chains downward through the DNS hierarchy. The objective is a domain name framework where the end client can validate that the results returned from a DNS query contain authentic information that was entered into the DNS by the delegated authority for that particular DNS zone.

What if we were able to place certificates—or references to certificates—into the DNS and protect them with DNSSEC? The *DNS-based Authentication of Named Entities* (DANE) Working Group of the IETF[0, 7] is considering this area of study. They are considering numerous scenarios at present, and the one of interest here does not replace the framework of Certification Authorities and domain name certificates, but it adds another phase of verification of the presented certificate.

The "Use Cases"[8] document from the DANE working group illustrates the proposed approach. I will quote a few paragraphs from this document. The first paragraph describes the form of attack that was perpetrated in June and July this year on the DigiNotar CA. It is not clear to me if the text predates this attack or not, but they are closely aligned in time:

> "Today, an attacker can successfully authenticate as a given application service domain if he can obtain a 'mis-issued' certificate from one of the widely-used CAs—a certificate containing the victim application service's domain name and a public key whose corresponding private key is held by the attacker. If the attacker can additionally insert himself as a man in the middle between a client and server (for example, through DNS cache poisoning of an A or AAAA record), then the attacker can convince the client that a server of the attacker's choice legitimately represents the victim's application service."[8]

So how can DNSSEC help here?

> "With the advent of DNSSEC [RFC 4033], it is now possible for DNS name resolution to provide its information securely, in the sense that clients can verify that DNS information was provided by the domain holder and not tampered with in transit.

> The goal of technologies for *DNS-based Authentication of Named Entities* (DANE) is to use the DNS and DNSSEC to provide additional information about the cryptographic credentials associated with a domain, so that clients can use this information to increase the level of assurance they receive from the TLS handshake process.
>
> This document describes a set of use cases that capture specific goals for using the DNS in this way, and a set of requirements that the ultimate DANE mechanism should satisfy. Finally, it should be noted that although this document will frequently use HTTPS as an example application service, DANE is intended to apply equally to all applications that make use of TLS to connect to application services named by domain names."[8]

Does DANE represent a comprehensive solution to this security vulnerability?

I would hesitate to be that definitive. As usual with many aspects of security, the objective of the defender is to expend a smaller amount of effort in order to force an attack to spend a far larger amount of effort. From this perspective, the DANE approach appears to offer significant promise because it interlocks numerous security measures and forces a potential attacker to compromise numerous independent systems simultaneously. Within the DANE framework the attacker cannot attack any certification authority, but must compromise a particular certification authority, and the attacker must also attack DNSSEC and compromise the information contained in signed DNS responses for that domain in order to reproduce the effects of the attack described here. This scenario seems to fit the requirement of a small amount of additional defensive effort by the server and the client, creating a significantly larger challenge to the attacker.

But many preconditions must be met here for this approach to be effective:

- DNSSEC needs to be ubiquitously deployed and maintained.
- Issued DNS certificates need to be published in the secure DNS zone using the DANE framework.
- Client DNS resolvers need not only to be DNSSEC-aware, but also to enforce DNSSEC outcomes.
- Applications, including browsers, need to validate the certificate that is being used to form the TLS connection against the information provided by a validated DNS response for the DANE credentials for that DNS zone.

It is probably not perfect, but it is a large step forward along a path of providing more effective security in the Internet.

Unfortunately, this solution does not constitute an instant solution ready for widespread use today—or even tomorrow. We could possibly see this solution in widespread use in a couple of years, but, sadly, it is more likely that securing the DNS for use in the Internet will not receive adequate levels of attention and associated financial resourcing in the coming years. It may take upward of 5 years before we see ubiquitous adoption of DNSSEC and any significant levels of its use by a DANE framework for certificates in the DNS. Until then there is the somewhat worrisome prospect of little change in the framework of Internet security from that used today, and the equally concerning prospect that this particular hacking event will not be the last.

Acknowledgement

I am indebted to Olaf Kolkman of NLnet Labs for a stimulating conversation about this attack and the implications for securing the Internet. NLnet Labs is one of a small number of innovative and highly productive research groups that has developed considerable levels of expertise in this area of security and the DNS.[9]

Postscript

When you lose that essential element of trust, your continued existence as a trusted Certification Authority is evidently a very limited one. On Tuesday September 20, 2011, the Dutch company DigiNotar was officially declared bankrupt in a Haarlem court.

Disclaimer

The views of this article do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

References

[0] Richard L. Barnes, "Let the Names Speak for Themselves: Improving Domain Name Authentication with DNSSEC and DANE," *The Internet Protocol Journal,* Volume 15, No. 2, March 2012.

[1] `http://www.guardian.co.uk/technology/2011/sep/05/digi-notar-certificate-hack-cyberwar`

[2] William Stallings, "SSL: Foundation for Web Security," *The Internet Protocol Journal,* Volume 1, No. 1, June 1998.

[3] Miek Gieben, "DNSSEC: The Protocol, Deployment, and a Bit of Development," *The Internet Protocol Journal,* Volume 7, No. 2, June 2004.

[4] Roy Arends, Rob Austein, Matt Larson, Dan Massey, and Scott Rose, "DNS Security Introduction and Requirements," RFC 4033, March 2005.

[5] Fox IT, "DigiNotar Certificate Authority breach, 'Operation Black Tulip,'"
http://www.rijksoverheid.nl/bestanden/documenten-en-publicaties/rapporten/2011/09/05/diginotar-public-report-version-1/rapport-fox-it-operation-black-tulip-v1-0.pdf

[6] Michael Myers, Rich Ankney, Ambarish Malpani, Slava Galperin, and Carlisle Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP," RFC 2560, June 1999.

[7] **http://datatracker.ietf.org/wg/dane/**

[8] Richard Barnes, "Use Cases and Requirements for DNS-Based Authentication of Named Entities (DANE)," RFC 6394, October 2011.

[9] **http://nlnetlabs.nl/**

[10] On March 26, 2012, at IETF 83 in Paris, France, a Technical Session with the title "Implementation Challenges with Browser Security" was held. The following presentations were given:

- Hannes Tschofenig: "Introduction"
- Eric Rescorla: "How do we get to TLS Everywhere?"
- Tom Lowenthal: "Cryptography Infrastructure"
- Chris Weber: "When Good Standards Go Bad"
- Ian Fette: "Lessons Learned from WebSockets (RFC 6455)"
- Jeff Hodges: "It's Not the End of the World"

All of these presentations are available from:
https://datatracker.ietf.org/meeting/83/materials.html

GEOFF HUSTON B.Sc., M.Sc., is the Chief Scientist at *Asia Pacific Network Information Centre* (APNIC), the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of numerous Internet-related books, was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of Trustees of the Internet Society from 1992 until 2001. E-mail: **gih@apnic.net**

Let the Names Speak for Themselves: Improving Domain Name Authentication with DNSSEC and DANE

by Richard L. Barnes, BBN Technologies

Authentication of domain names is a fundamental function for Internet security. In order for applications to protect information from unauthorized disclosure, they need to make sure that the entity on the far end of a secure connection actually represents the domain that the user intended to connect to. For many years, authentication of domain names has been accomplished by having third-party *Certification Authorities* attest to which entities could represent a domain name. This system of external authorities, however, has recently come under heavy attack, and there have been several high-profile compromises[0]. The *Domain Name System Security Extensions* (DNSSEC) offer an alternative channel for distributing secure information about domain names, through the *Domain Name System* (DNS) itself. The *DNS-based Authentication of Named Entities* (DANE) working group in the *Internet Engineering Task Force* (IETF) has developed a new type of DNS record that allows a domain itself to sign statements about which entities are authorized to represent it. End users' applications can use these records either to augment the existing system of Certification Authorities or to create a new chain of trust, rooted in the DNS.

Authentication

Without authentication, other security services are moot. There is little point in Alice's encrypting information en route to Bob if she has not first verified that she is talking to Bob and not an attacker Eve. In the context of Internet applications, authentication is about ensuring that users know whom they are talking to, and in most cases, that "whom," is represented by a domain name. For example, in the *Hypertext Transfer Protocol* (HTTP), the "authority" section of a *Uniform Resource Identifier* (URI) indicates the domain name of the server that will fulfill requests for that URI. So when an HTTP user agent starts a TCP connection to a remote server, it needs to verify that the server is actually authorized to represent that domain name[1].

The most common security protocol used by Internet applications is the *Transport Layer Security* (TLS) protocol[2]. TLS provides a layer above TCP that facilitates authentication of the remote side of the connection as well as encryption and integrity protection for data. TLS underlies *Secure HTTP* (HTTPS) and secure e-mail[1, 3, 4], and provides hop-by-hop security in real-time multimedia and instant-messaging protocols[5, 6]. In all of these applications, the server that the user ultimately wants to connect to is identified by a DNS domain name[7, 8]. A user might enter `https://example.com` into a web browser or send an e-mail to `alice@example.com`.

One of the main purposes of using TLS in these cases is thus to assure the user that the entity on the other end of the connection actually represents `example.com`; in other words, to authenticate the server as a legitimate representative of the domain name. Note that these comments apply to *Datagram Transport Layer Security* (DTLS) as well, because it provides the same functions as TLS for *User Datagram Protocol* (UDP) packet flows[9].

Today, a server asserts its right to represent a domain by presenting a *Public Key Infrastructure* (PKIX) digital certificate containing that domain[8, 10]. A certificate is an attestation by a Certification Authority of a binding between a public key and a name—the entity holding the corresponding private key is authorized to represent that name. TLS ensures that only the holder of a given private key can read the encrypted data; the certificate ensures that the holder of the key represents the desired name.

Current TLS-based applications maintain a list of Certification Authorities whose certificates they will accept. Unfortunately, over time, these lists have grown very long, with major web browsers trusting nearly 200 Certification Authorities, representing a diverse range of organizations. Because any of these Certification Authorities can vouch for any domain name, a long list creates many points of vulnerability; a compromise at any point allows the attacker to issue certificates for any domain. Several recent attacks have taken advantage of this fact by targeting smaller Certification Authorities as a way to obtain certificates for major domains. For example, an attack through DigiNotar against Google is discussed in this issue[0].

DNSSEC offers an alternative to Certification Authorities. In the DNSSEC system, each domain holder can act as an authority for subordinate domains. The IETF DANE working group has developed a DNS record format for "certificate associations," so that domain holders can sign statements about which certificates can be used to authenticate as that domain. In effect, this scenario allows a domain to speak for itself, instead of through a third-party Certification Authority. DANE associations can be used either as a check on the current model (for example, to limit which Certification Authorities may vouch for a domain) or as an alternative trust path, rooting trust in a DNSSEC authority instead of a Certification Authority. Work on the protocol document is drawing to a close, and several prototype implementations are already in progress.

Background: PKIX and DNSSEC

At one level, the choice of which authentication technology to use is a choice of authorities and scoping. As mentioned previously, authentication is fundamental for security, but it is also very hard to accomplish scalably. For example, a web browser needs to be able to authenticate any website the user chooses to visit. It would clearly not work for each browser vendor to send a human representative to meet every website owner in order to find out what public key should be used for that website.

So instead of relying on having preestablished relationships with every entity we want to authenticate, we rely on centralized authorities to do identity checking. The authorities then create credentials that anyone else can check, so that if the credential is valid and you believe the authority is trustworthy, then the entity holding the credential has the indicated identity.

In a technical sense, an entity holds a credential if it holds the private key corresponding to the public key in the credential. The credential encodes a *binding* between the public key and the identity, asserted by the authority.

Authority is of course not a purely digital concept. If we want to know a person's name in real life we do not just ask them directly, because the person could lie. Instead we look to a credential issued by an authority, such as a driver's license or birth certificate. So the technology question here is how to manage authorities, and how to encode these credentials.

The IETF has defined two major cryptographic authority systems: PKIX, based on digital certificates[10]; and DNSSEC, based on the DNS[11]. Both of these systems allow authorities to associate public keys with identities, and both arrange these authorities hierarchically.

The hierarchy is important because it allows a *relying party* (someone who is verifying identities) to choose whom to trust. In these hierarchical systems, an authority's identity can itself be attested by a credential issued by another authority. When a relying party wants to verify a credential issued by an authority A, he then has to verify that A's credential is valid (under an authority B), and so on until he reaches an authority that he trusts. This sequence of credentials constitutes a logical path through the hierarchy, known as a "certification path" in PKIX terminology (Figure 1).

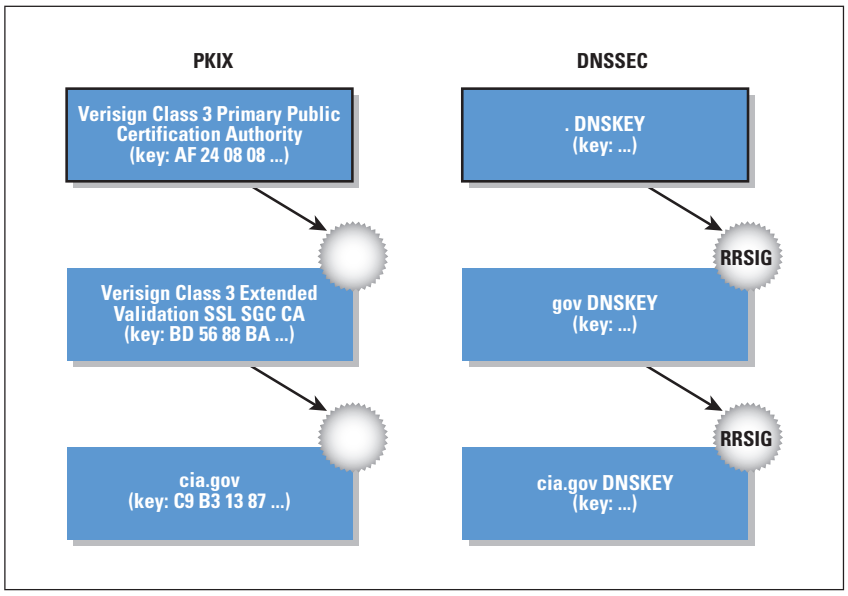


Figure 1: PKIX and DNSSEC Trust Hierarchies

In order to be useful as a given relying party to authenticate someone, a certification path has to end in a *trust anchor,* that is, an authority that the relying party trusts to make assertions. In the DNSSEC context, relying parties can in principle have only one trust anchor, namely the DNS root, although alternatives to the root have been proposed[12]. The PKIX system, on the other hand, does not represent a single globally consistent hierarchy, so in order to be able to validate many certificates, relying parties often have to choose many trust anchors.

Crossing the Streams

Current TLS-based applications rely on PKIX for authentication of domain names, which has facilitated fairly broad deployment, but also created some vulnerabilities. PKIX is based on a very general digital certificate system called X.509, and because of this generality, it has no inherent binding to the DNS. This situation creates two problems when it comes to authenticating domain names.

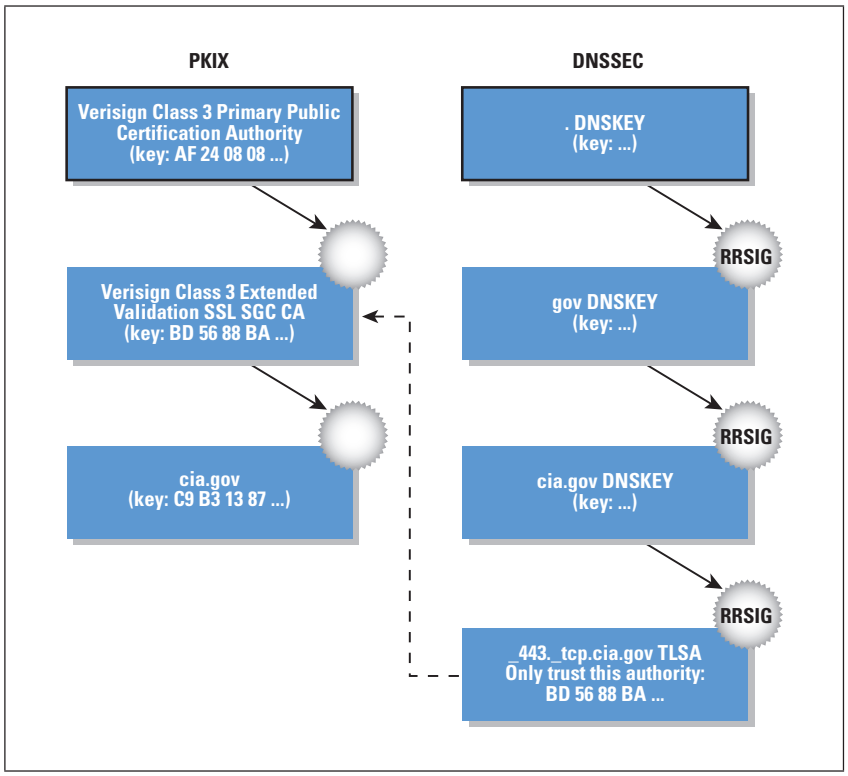
First, unlike the DNS, which has a single global root, there is no single authority under which all PKIX certificates can be verified. Indeed, there is an open marketplace of authorities, where each entity can choose which authority will sign its certificate, leaving relying parties with a choice: Either they must trust every authority that has signed a certificate for an entity it wants to authenticate, or they will be unable to validate the identities of some entities. In general, current software has preferred the former approach of trusting many authorities, to the extent that modern browsers and operating systems will trust up to 200 authorities by default. Users can add to this list, for example, using the "Accept this certificate?" dialogs in their browsers, but it can be very difficult to remove trust anchors from the default list[13].

Second, PKIX authorities today are not constrained in the scope, so they can issue credentials for any name—even those for whom they have no real information (in contrast to the DNS—where each zone can vouch only for sub-domains; only the root can act with impunity). Conversely, there is no real way for a relying party to know what authority should be vouching for a site, so if a rogue authority were to issue a certificate to an unauthorized party, relying parties would have no way to detect it.

Given these vulnerabilities, any of the many authorities trusted within the PKIX system can attack any domain by issuing a false certificate from that domain. This false certificate can then be used to masquerade as the victim domain, for example, to perform a man-in-the-middle attack. Note that the authority itself is not necessarily the bad actor in this attack—it could be an external attacker that can obtain illicit access to the systems that issue certificates. The risks of having broadly trusted Certificatation Authorities have recently become clear, because attackers were able to break into two small Certification Authorities and create fraudulent certificates for Google and Facebook, among others[14, 15].

The goal of DANE is to address some of the vulnerabilities of the current PKIX ecosystem by allowing DNSSEC—to "cross the streams" to allow domains to publish information secured with DNSSEC that can add additional security to PKIX certificates used for TLS. For example, a domain might use DANE to inform relying parties of which authorities can be trusted, as illustrated in Figure 2.

Figure 2: Using a DANE TLS Associations (TLSA) Record to Indicate Which PKIX Authority Should Be Trusted



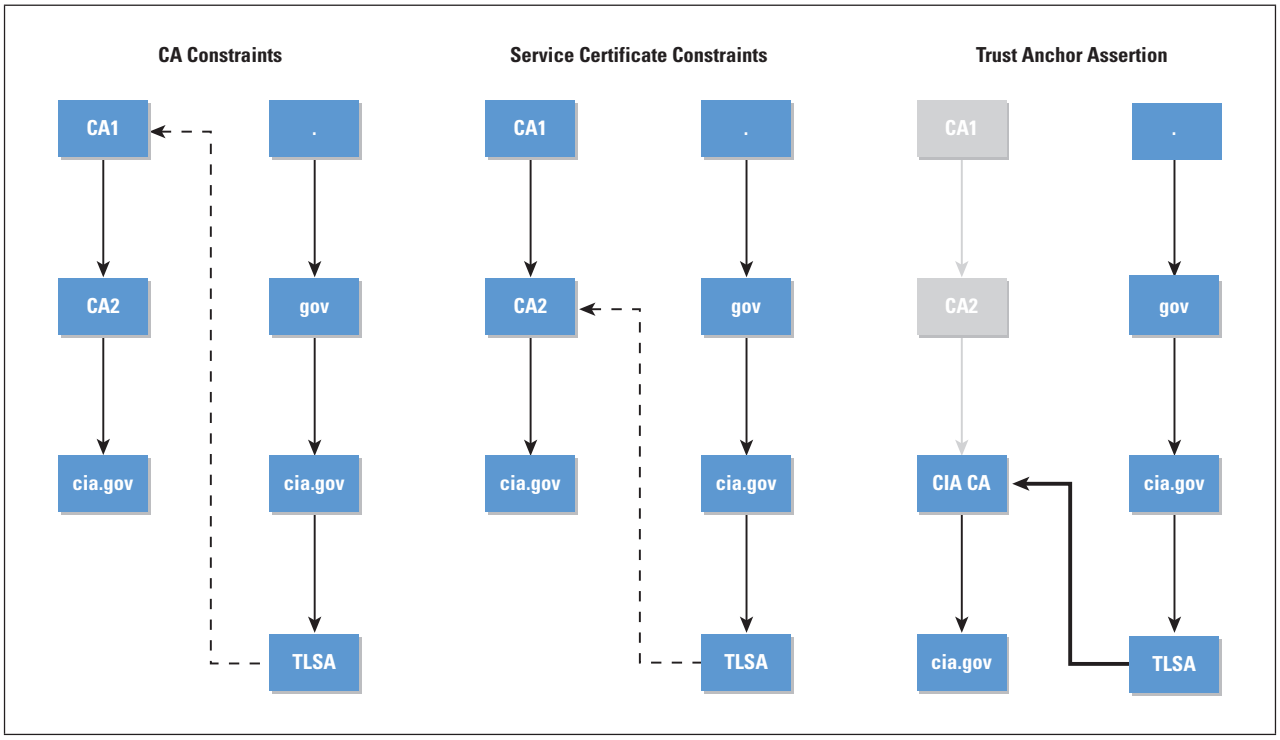
DANE Records

If the goal of DANE is to allow domain operators to make statements about how clients should judge TLS certificates for their domains, then what sorts of statements should DANE allow them to make? The DANE use cases document[16] lays out three major types of statements (Figure 3):

1. *CA Constraints:* The client should accept only certificates issued under a specific Certificatation Authority.
2. *Service Certificate Constraints:* The client should accept only a specific certificate.
3. *Trust Anchor Assertion:* The client should use a domain-provided trust anchor to validate certificates for that domain.

All three of these statements can be viewed as constraining the scope of trust anchors. The first two types limit the scope of existing trust anchors, whereas the third provides the client with a new trust anchor (still within a limited scope). More on these anchors in a moment.

Figure 3: DANE Use Cases



The current draft DANE protocol defines a DNS Resource Record type TLSA for describing *"TLS Associations"*—statements about what certificates are "associated" to a domain[17]. Each TLSA record has three basic fields:

- *Usage:* Which type of statement this record is making
- *Selector/Matching:* How a TLS certificate chain should be matched against this record (for example, by exact match, by public key, or by SHA-1 digest)
- *Certificate for Association:* The actual data against which the TLS certificate chain should be matched

These records are stored under the target domain with a prefix that indicates the transport and port number for the TLS server. So for example, if Alice runs a secure web service at **`example.com`** and wants to tell clients that they should accept only certificates from the Charlie's CA, she could provision a TLSA record under **`_443._tcp.example.com`** with the following contents:

- *Usage:* CA constraint
- *Selector/Matching:* SHA-1 digest
- *Certificate for Association:* SHA-1 digest of Charlie's certificate

When a client Bob wants to connect to **`https://example.com`**, he can find these TLSA records and apply Alice's constraints when he validates the server certificate.

Adding Constraints to PKIX

The major objective of the CA constraints and service certificate constraints is to guard against "mis-issue" of certificates. A certificate is "mis-issued" when a CA issues a certificate to an entity that does not actually represent the domain name in the certificate. Mis-issue can come about in many ways, including through malicious Certification Authorities, compromised Certification Authorities (as in the Comodo and DigiNotar example discussed previously), or Certification Authorities that are simply misled as to the attacker's identity through fraud or other means. Today, mis-issue can be difficult to detect, because there is no standard way for clients to figure out which Certification Authorities are supposed to be issuing certificates for a domain. When an attacker issued false certificates for the Google Gmail service under the DigiNotar Certification Authority, it was noticed only because a vigilant user posted to a Gmail help forum.[18]

By contrast, domain operators know exactly which Certification Authorities they have requested certificates from, and, of course, which specific certificates they have received. With DANE, the domain operator can convey this information to the client. For example, to guard against the DigiNotar attack, Google could have provisioned a TLSA record expressing a Certification Authority constraint with its real Certification Authority (which is not DigiNotar) or a certificate constraint with its actual certificate. Then DANE-aware clients would have been able to immediately see that the DigiNotar certificates were improperly issued and possibly indicative of a man-in-the-middle attack.

Empowering Domain Operators

According to data from the EFF SSL Observatory, which scans the whole IPv4 address space for HTTPS servers and collects their certificates, around 48 percent of all HTTPS servers present self-signed certificates[19]. An unknown number of other servers present certificates issued under Certification Authorities that are not in the major default trust anchor lists. For example, the United States Air Force web portal uses a certificate issued under a Department of Defense Certification Authority that is not trusted by Firefox[20]. In the current environment, most clients cannot authenticate these servers at all; they have to rely on users manually checking certificates, hopefully with some out-of-band information. As a result, these servers and their users are highly vulnerable to man-in-the-middle attacks against their supposedly secure sessions.

DANE Trust Anchor Assertions enable the operators of a domain to advertise a new trust anchor, under which certificates for that domain will be issued. Using these records, clients can dynamically discover what trust anchors they should accept for a given domain, instead of relying on a static list provided by a browser or operating system.

It may seem odd to talk about a domain supplying a client with trust anchors, because trust anchor provisioning is typically a very sensitive activity. If an attacker is able to install a trust anchor into a victim's trust anchor store, then the attacker can masquerade under any name he wants by issuing certificates under that name. The PKIX working group even defined a whole protocol for managing trust anchors[21].

DANE ensures that this trust anchor provisioning is secure by applying scoping and verifying that scoping using DNSSEC. DANE trust anchor assertions are scoped to a particular domain name, so even if an attacker can introduce a false trust anchor, he can use it to spoof only a single name. Furthermore, trust anchor assertions must be DNSSEC-signed, so clients can verify that the entity providing the trust anchor represents the domain in question. Ultimately, the client still has to have a list of trust anchors configured—but they are DNSSEC trust anchors instead of PKIX trust anchors.

Of course, in principle, a client needs only one trust anchor for DNSSEC, the root zone trust anchor. Because control of the DNS root does not change very often, it makes sense for this trust anchor to be statically configured!

The ability of a domain operator to explicitly indicate a trust anchor for a domain is obviously very powerful. It may be tempting to ask whether this case is really the only use case that DANE needs, that is, whether the constraint cases mentioned previously are needed at all. The answer is that the constraint cases are useful as a way to fold in PKIX validation with external Certification Authorities in addition to domain-asserted trust anchors. Most obviously, this feature is useful in transition, when not all clients will be DANE-aware. But even in the longer term, it is possible that Certification Authorities will be able to provide added value over DANE. For example, while DANE is made to bind certificates to domain names, Certification Authorities can vouch for bindings of certificates to other things, such as the legal identity and physical location attested in Extended Validation certificates[22].

Transition Challenges

As described previously, DANE offers some valuable new security properties for TLS authentication. But as with most IETF technologies—especially security technologies—there are some challenges to be overcome and some new potential pitfalls.

The most significant constraint for DANE deployment is DNSSEC deployment. On the server side, this problem is not a significant one because DNSSEC support is spreading fairly rapidly. On the client side, it may be more difficult. Although there are DNS libraries with robust DNSSEC support, many of the major DNS *Application Programming Interfaces* (APIs) that applications use do not provide any information about the DNSSEC status of the results returned.

So in order to implement DANE, application developers may have to re-factor their DNS support in addition to querying for some new record types. If more sites come to rely on DANE, then this process could also draw increasing attention to the various types of intermediaries that cause DNSSEC breakage (for example, home gateways that set DNS flags improperly).

Adding DNSSEC to the TLS connection process can also add significant latency to the TLS connection process. In addition to completing the TLS handshake and certificate validation, the client has to wait for several DNS round trips and then validate the chain of DNSSEC signatures. These combined delays can add up to multiple seconds of latency in connection establishment. Especially for real-time protocols such as HTTPS, *Session Initiation Protocol* (SIP), or *Extensible Messaging and Presence Protocol* (XMPP), such delay is clearly undesirable.

One mechanism proposed to mitigate these delays is to have the server pre-fetch all of the relevant DNSSEC records, namely all of the DS, DNSKEY, and RRSIG records chaining back to the root[27]. Then the server can provide a serialized version of the DNSSEC records in the TLS handshake, saving the client the latency of the required DNS queries. The details of this mechanism, however, are still being worked out among the DANE, TLS, and PKIX working groups[23]. A prototype version is now available in the Google Chrome web browser[24].

Security Considerations

From a security perspective, the major effect of DANE is the new role that DNS operators will play in securing Internet applications. Although DNSSEC has always meant that DNS operators would have more security functions, DANE deployment will give them an explicit effect on application security, acting as arbiters of who can authenticate under a given name in TLS. Especially if services use trust anchor assertions, DNS operators will play an analogous role to the one Certification Authorities play today—a compromise in a DNS operator will allow an attacker to masquerade as a victim domain (albeit for a more limited set of domains because of DANE constraints on names). So DNS operators are likely to inherit many of the security troubles that Certification Authorities experience today and will need to strengthen their security posture accordingly.

Another more subtle risk arises from the fact that the operator of a DNS zone is not always the same as the entity that is authorized to control the contents of the zone, which we will call the "domain holder." We used the phrase "domain operator" previously because DNSSEC protects DNS information only between the operator's name server and the client—it does not say that what is provisioned in the name server is authorized by the domain holder.

When a domain is operated by a third party, that third party is a point of vulnerability between the client and the holder of the domain. If the domain operator provides false DANE information through malice or compromise, then a client will not be able to distinguish it from genuine DANE information. To some extent, this risk is not really new; because many current Certification Authorities authenticate requests for domain certificates based on information that is under the control of the domain operator, domain operators can already influence the credentialing process. With DANE, however, the vulnerability is much easier to exploit, for example, because the DNS operator does not have to trick a third party. This vulnerability is also fundamental to protocols that rely on DNSSEC for security, and the implications for DANE are discussed in detail in the DANE use cases document[16]. The main mitigation is simply increased care on the part of domain holders to ensure that domain operators are not behaving badly.

Conclusions

For many years now, Internet applications have relied on assertions by third-party PKIX Certification Authorities to ensure that a server holding a particular private key was authorized to represent a domain. The promise of DANE is a more direct interaction between clients and the domains they interact with, secured by DNSSEC. In the short run, DANE can be deployed as an adjunct to the current system of certificates and authorities, adding constraints to better protect domains. In the long run, DANE will also allow domain operators to vouch for their own names.

The transition and security problems that face DANE are largely the growing pains of DNSSEC. It is not that DANE is causing these problems itself; rather, the problems arise because DANE is the first real application of DNSSEC that is expected to be widely deployed. So although it may be difficult to mitigate some of the security problems that DANE raises, and to enable more robust DNSSEC support in applications and gateways, these changes will ultimately make it simpler for applications to use DNSSEC for other purposes.

The DANE working group is making consistent progress on its deliverables, and there are already some prototype deployment tools. Their use cases document has been published as RFC 6394[16], and the corresponding document defining the TLSA record type is starting to mature[17]. As of this writing, it is in Working Group Last Call. On the client side, a variant of DANE has already been implemented in Google Chrome; on the server side, prototype tools are available to generate DANE records and to generate "DNSSEC-stapled" certificates based on DANE records[24, 25]. There is also an early-stage command-line tool for generating and verifying TLSA records[26].

References

[0] Geoff Huston, "Hacking Away at Internet Security," *The Internet Protocol Journal,* Volume 15, No. 1, March 2012.

[1] Eric Rescorla, "HTTP Over TLS," RFC 2818, May 2000.

[2] Tim Dierks and Eric Rescorla, Editors, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, August 2008.

[3] Chris Newman, "Using TLS with IMAP, POP3 and ACAP," RFC 2595, June 1999.

[4] Paul Hoffman, "SMTP Service Extension for Secure SMTP over Transport Layer Security," RFC 3207, February 2002.

[5] Jonathan Rosenberg, Henning Schulzrinne, Gonzalo Camarillo, Alan Johnston, Jon Peterson, Robert Sparks, Mark Handley, and Eve Schooler, "SIP: Session Initiation Protocol," RFC 3261, June 2002.

[6] Peter Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core," RFC 6120, March 2011.

[7] Paul Mockapetris, "Domain Names – Concepts and Facilities," RFC 1034, November 1987.

[8] Peter Saint-Andre and Jeff Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)," RFC 6125, March 2011

[9] Eric Rescorla and Nagendra Modadugu, "Datagram Transport Layer Security Version 1.2," RFC 6347, January 2012.

[10] David Cooper, Stefan Santesson, Stephen Farrell, Sharon Boeyen, Russell Housley, and Tim Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC 5280, May 2008.

[11] Roy Arends, Rob Austein, Matt Larson, Dan Massey, and Scott Rose, "DNS Security Introduction and Requirements," RFC 4033, March 2005.

[12] `https://dlv.isc.org/`

[13] `http://arstechnica.com/apple/news/2011/09/safari-users-still-susceptible-to-attacks-using-fake-diginotar-certs.ars`

[14] **http://blogs.comodo.com/it-security/data-security/the-recent-ra-compromise/**

[15] **http://www.theregister.co.uk/2011/09/06/diginotar_audit_damning_fail/**

[16] Richard Barnes, "Use Cases and Requirements for DNS-Based Authentication of Named Entities (DANE)," RFC 6394, October 2011.

[17] Paul Hoffman and Jakob Schlyter, "Using Secure DNS to Associate Certificates with Domain Names for TLS," Internet Draft, work in progress, **draft-ietf-dane-protocol-16**, February 2012.

[18] **http://www.google.co.uk/support/forum/p/gmail/thread?tid=2da6158b094b225a&hl=en**

[19] **http://www.eff.org/observatory**

[20] **https://www.my.af.mil/**

[21] Russ Housley, Sam Ashmore, and Carl Wallace, "Trust Anchor Management Protocol (TAMP)," RFC 5934, August 2010.

[22] **http://cabforum.org/Guidelines_v1_2.pdf**

[23] Adam Langley, "Serializing DNS Records with DNSSEC Authentication," Internet Draft, work in progress, July 2011, **draft-agl-dane-serializechain**.

[24] **http://www.imperialviolet.org/2011/06/16/dnssec-chrome.html**

[25] **https://dane.xelerance.com/**

[26] **https://github.com/pieterlexis/swede**

[27] Wikipedia, "List of DNS record types," **http://en.wikipedia.org/wiki/List_of_DNS_record_types**

RICHARD BARNES has been with BBN Technologies since 2005. Richard is a member of BBN's Internet standards security team. In that role, he currently leads BBN's IETF standards efforts in the areas of geolocation, presence, and emergency calling. He is chair of the IETF GEOPRIV working group and a member of the IETF Security Area Directorate (SECDIR), and he is one of the program chairs of the Emergency Services Workshop. Prior to joining BBN, he was a student at the University of Virginia (United States), from which he received a B.A. and M.S. in Mathematics, with research focused on biologically based neural networks, quantum informatics, and network security. E-mail: **rbarnes@bbn.com**

A Retrospective: Twenty-Five Years Ago

by Geoff Huston, APNIC

The Information Technology business is one that rarely pauses for breath. Gordon Moore noted in 1965 that the number of components in integrated circuits had doubled every year from 1958 to 1965, and confidently predicted that this doubling would continue "for at least 10 years." This feature has been a continuing feature of the silicon industry for the past 50 years now, and its constancy has transformed this prediction into *Moore's Law*. The implications of this constant impetus for innovation in this industry have resulted in an industry that is incapable of remaining in stasis, and what we have instead is an industry that completely reinvents itself in cycles as short as a decade.

Looking back over the past 25 years, we have traversed an enormous distance in terms of technical capability. The leading silicon innovations of the late 1980s were in the Intel 80486 chip, which contained 1 million transistors on a single silicon chip with a clock speed of 50 MHz, and a similarly capable Motorola 68040 processor. Twenty-five years later the state of the art is a multicore processor chip that contains just under 3 billion individual transistors and clock speeds approaching 4 GHz. And where has all that processing power gone? In the same period we have managed to build extremely sophisticated programmed environments that have produced such products as Apple's *Siri* iPhone application, which combines voice recognition with a powerful information manipulation system, and we have packaged all of this computing capability into a device that fits comfortably in your pocket with room to spare!

Given that the last 25 years in IT has been so active, to look back over this period and contemplate all that has happened is a daunting task, and I am pretty sure that any effort to identify the innovative highlights in that period would necessarily be highly idiosyncratic. So instead of trying to plot the entire story that took us from then to now, I would like instead just to look at "then." In this article, to celebrate 25 combined years of *The Internet Protocol Journal* (IPJ)[2, 3] and its predecessor *ConneXions—The Interoperability Report*[0], I would like to look at the networking environment of the late 1980s and see what, if anything, was around then that was formative in shaping what we are doing today, and how it might influence our tomorrow.

The Computing Landscape of the Late 1980s

The computing environment of the late 1980s now seems to be quite an alien environment. Obviously there were no pocket-sized computers then. Indeed there were no pocket-sized mobile phones then. (I recall a visit from a salesman at the time who sported the very latest in mobile telephony—a radio setup that was the size of a briefcase!)

In 1987 the IT world was still fixated with the mainframe computer, which was basking in its last couple of years of viability in the market. IBM enjoyed the dominant position in this marketplace, and *Digital Equipment Corporation* (DEC) was competing with IBM with its VAX/VMS systems. These systems were intended to take the place of the earlier DEC-10 architectures, as well as offering an upgrade path for the hugely successful PDP-11 minicomputer line. The typical architecture of the computing environment was still highly centralized, with a large multiuser system at its core, and an attendant network or peripheral devices. These peripheral devices were traditionally video terminals, which were a simple ASCII keyboard and screen, and the interaction with the mainframe was through simple serial line character-based protocols.

Although it may not have been universally accepted at the time, this period at the end of the 1980s marked the end of the custom-designed mainframe environment, where large-scale computer systems were designed as a set of component subsystems, placed into a rack of some sort and interconnected through a bus or blackplane. Like many other human efforts, as far as the mainframe computer sector was concerned its final achievements were its greatest.

While the mainframe sector was inexorably winding down, at the other end of the market things were moving very quickly. The Zylogics Z80 processor of the mid-1970s had been displaced by the Intel 8080 chip, which evolved rapidly into 16-bit, then 32-bit processor versions. By 1987 the latest chip was the Intel 80386, which could operate with a clock speed up to 33 MHz. The bus was 32 bits wide, and the chip supported a 32-bit address field. This chip contained some 275,000 transistors, and was perhaps the transformative chip that shifted the personal computer from the periphery of the IT environment to the mainstream. This chip took on the mainframe computer and won. The evolving architecture of the late 1980s was shifting from a central processing center and a cluster of basic peripheral devices to one of a cluster of personal desktop computers.

The desktop personal computer environment enabled computing power to be treated as an abundant commodity, and with the desktop computer came numerous interface systems that allowed users to treat their computer screens in a manner that was analogous to a desktop. Information was organized in ways that had a visual counterpart, and applications interacted with the users in ways that were strongly visual. The approach pioneered by the Xerox Star workstation in the late 1970s and brought to the consumer market through the Apple Lisa and Macintosh systems were then carried across into the emerging "mainstream" of the desktop environment with Windows 2.0 in the late 1980s.

The state of the art of portability was still in the category of "luggable" rather than truly portable, and the best example of what was around at the time is the ill-fated Macintosh Portable, which like its counterpart in the portable phone space was the size of a briefcase and incredibly heavy.

Oddly enough, while the industry press was in raptures when it was released in 1989, it was a complete failure in the consumer market. The age of the laptop was yet to come.

One major by-product in this shift in the computing environment to a distributed architecture was a major shift in the attention to networking, and at the same time as there was a large-scale shift in the industry from mainframes to personal computers, there were also numerous major changes in the networked environment.

The Networking Environment of the Late 1980s

A networking engineer in the late 1980s was probably highly conversant in how to network serial terminals to mainframes. The pin-outs in the DB-25 plug used by the RS-232 protocol was probably one of the basic ABCs of computer networking. At that time much of the conventional networked environment was concerned with connecting these terminal devices to mainframes, statistical multiplexors, and terminal switches, and serial switch suppliers such as Gandalf and Micom were still important in many large-scale computing environments.

At the same time, another networking technology was emerging—initially fostered by the need to couple high-end workstations with mainframes—and that was *Ethernet*. Compared to the kilobits per second typically obtained by running serial line protocols over twisted pairs of copper wires, the 10-Mbps throughput of Ethernet was blisteringly fast. In addition, Ethernet could span environments with a diameter of around 1500 meters, and with a certain amount of tweaking or with the judicious use of Ethernet bridges and fibre-optic repeaters this distance could be stretched out to 10 km or more.

Ethernet heralded a major change in the networked environment. No longer were networks hub-and-spoke affairs with the mainframe system at the center. Ethernet supplied a common bus architecture that supported any-to-any communications. Ethernet was also an open standard, and many vendors were producing equipment with Ethernet interfaces. In theory, these interfaces all interoperated, at least at the level of passing Ethernet frames across the network (aside from a rather nasty incompatibility between this original Digital-Intel-Xerox specification and the IEEE 802.3 "standardized" specification!).

However, above the basic data framing protocol the networked environment was still somewhat chaotic. I recall the early versions of the multiprotocol routers produced by Proteon and Cisco supported more than 20 networking protocols! There was *DECnet,* a proprietary network protocol suite from the Digital Equipment Corporation, which at around 1987 had just released Phase IV, and was looking toward a Phase V release that was to interoperate with the International Organization for Standardization's *Open Systems Interconnection* (OSI) protocol suite[1] (more on this subject a bit later).

There was IBM's *Systems Network Architecture* (SNA), which was a hierarchical network that supported a generic architecture of remote job entry systems clustered around a central service mainframe. There was the *Xerox Network Services* (XNS) protocol used by Xerox workstations. Then there were Apollo's *Network Computing Architecture* (NCA) and Apple's *AppleTalk*. And also in this protocol mix was the *Transmission Control Protocol/Internet Protocol* (TCP/IP) protocol suite, used at that time predominately on UNIX systems, although implementations of TCP/IP for Digital's VAX/VMS system were very popular at the time. A campus Ethernet network of the late 1980s would probably see all of these protocols, and more, being used concurrently.

And there was the ISO-OSI protocol suite, which existed more as a future protocol suite than as a working reality at the time.

The ISO-OSI and TCP/IP protocol suites were somewhat different from the others that were around at the time because both were deliberate efforts to answer a growing need for a vendor-independent networking solution. At the time the IT environment was undergoing a transition from the monoculture of a single vendor's comprehensive IT environment—which bundled the hardware of the mainframe, network, peripherals, terminals, and the software of the operating system, applications, and network all into the one bundle—into a piecemeal environment that included a diverse collection of personal workstations, desktop computers, peripherals, and various larger minicomputers and mainframe computers in one environment. What was needed was a networking technology that was universally supported on all these various IT assets. What we had instead was a more piecemeal environment. Yes, it was possible to connect most of these systems into a common Ethernet substrate, but making A talk to B was still a challenge, and various forms of protocol translation units were also quite commonplace at the time. What the industry needed was a vendor-independent networking protocol, and there were two major contenders for this role.

ISO-OSI and TCP/IP

The ISO-OSI protocol suite was first aired in 1980. It was intended to be an all-embracing protocol suite that embraced both the IEEE 802.3 Ethernet protocols and the X.25 packet switching protocols that were favoured by many telephony operators as their preferred wide-area data services solution. The ISO-OSI network layer included many approaches, including the telephony sector's *Integrated Service Digital Network* (ISDN), a *Connection-Oriented Network Service* (CONS), a virtual circuit function based largely on X.75 that was essentially the "call-connection" function for X.25, and a *Connectionless Network Service* (CLNS), based loosely on the IP protocol with the use of the *End System-to-Intermediate System Routing Exchange Protocol* (ES-IS) routing protocol.

Above the network layer were numerous end-to-end transport protocols, notably *Transport Protocol Class 4* (TP4), a reliable connection-oriented transport service, and *Transport Protocol Class 0* (TP0), a connectionless packet datagram service. Above this layer was a Session Layer, X.215, used by the TP4 CONS services, and a Presentation Layer, defined using the *Abstract Syntax Notation One* (ASN.1) syntax.

ISO-OSI included numerous application-level services, including *Virtual Terminal Protocol* (VTP) for virtual terminal support, *File Transfer Access And Management* (FTAM) for file transfer, *Job Transfer And Management* (JTAM) for batch job submission, *Message Handling System* (MHS, also known as X.400) for electronic mail, and the X.500 Directory service. ISO-OSI also included a *Common Management Information Protocol* (CMIP). ISO-OSI attempted to be everything to everybody, as evidenced by the "kitchen sink" approach adopted by many of the OSI standardization committees at the time.

When confronted by many technology choices, the committees apparently avoided making a critical decision by incorporating both approaches into the standard. The most critical decision in this protocol suite was the inclusion of both connection-oriented and connectionless networking protocols. They also used session and presentation layer protocols, whose precise role was a mystery to many! ISO-OSI was a work-in-progress at the time, and the backing of the telephone sector, coupled with the support of numerous major IT vendors, gave this protocol an aura of inevitability within the industry. Whatever else was going to happen, there was the confident expectation that the 1990s would see all computer networks move inevitably to use the ISO-OSI protocol suite as a common, open, vendor-neutral network substrate.

If the ISO-OSI had a mantra of inevitably, the other open protocol suite of the day, the TCP/IP protocol suite, actively disclaimed any such future ambitions. TCP/IP was thought of at the time as an experiment in networking protocol design and architecture that ultimately would go the way of all other experiments, and be discarded in favor of a larger and more deliberately engineered approach. Compared to the ISO-OSI protocols, TCP/IP was extremely "minimalist" in its approach. Perhaps the most radical element in its design was to eschew the conventional approach at the time of building the network upon a reliable data link protocol. For example, in DECnet Phase IV, the data link protocol, *Digital Data Communications Message Protocol* (DDCMP), performed packet integrity checks and flow control at the data link level. TCP/IP gracefully avoided this problem by allowing packets to be silently dropped by intermediate data switches, or corrupted while in flight. It did not even stipulate that successive packets within the same end-to-end conversation follow identical paths through the network.

Thus the packet switching role was radically simplified because now the packet switch did not need to hold a copy of transmitted packets, nor did it need to operate a complex data link protocol to track packet transmission integrity and packet flow control. When a switch received a packet, it forwarded the packet based on a simple lookup of the destination address contained in the packet into a locally managed forwarding table. Or it discarded the packet.

The second radical simplification in TCP/IP was the use of real-time packet *fragmentation*. Previously, digital networks were constructed in a "vertically integrated" manner, where the properties of the lower layers were crafted to meet the intended application of the network. Little wonder that the telephone industry put its support behind X.25, which was a reliable unsynchronized digital stream protocol. If you wanted low levels of jitter, you used a network with smaller packet sizes, whereas higher packet sizes improved the carriage efficiency. Ethernet attempted to meet this wide variance in an agnostic fashion by allowing packets of between 64 and 1500 octets, but even so there were critics who said that for remote terminal access the smallest packets were too large, and for large-scale bulk data movement the largest packets were too small. *Fiber Distributed Data Interface* (FDDI), the 100-Mbps packet ring that was emerging at the time as the "next thing" as commodity high-speed networking used a maximum size of 4000 octets packets in an effort to improve carriage efficiency, whereas the *Asynchronous Transfer Mode* (ATM) committee tried to throw a single-packet-size dart at the design board and managed to get the rather odd value of 53 octets!

IP addressed this problem by trying to avoid it completely. Packets could be up to 64,000 octets long, and if a packet switch attempted to force a large packet through an interface that could not accept it, the switch was allowed to divide the packet into appropriately sized autonomous fragments. The fragments were not reassembled in real time: that was the role of the ultimate receiver of the packets.

As an exercise in protocol design, IP certainly showed the elegance of restraint. IP assumed so little in terms of the transmission properties of the underlying networks that every packet was indeed an adventure! But IP was not meant to be the protocol to support the prolific world of communicating silicon in the coming years. This protocol and the IP networks that were emerging in the late 1980s were intended to be experiments in networking. There was a common view that the lessons learned with experience of operating high-speed local networks and wide-area networks using the TCP/IP protocol suite would inform the larger industry efforts. The inclusion of IP-based technologies in the ISO-OSI protocol suite[4] was a visible instantiation of this proposed evolutionary approach.

While these two protocol suites vied with each other for industry attention at the time, there was one critical difference: It was a popular story at the time that the ISO-OSI protocol suite was a stack of paper some 6 feet high, which cost many hundreds of dollars to obtain, with no fully functional implementations, whereas the TCP/IP protocol suite was an open-sourced and openly available free software suite without any documentation at all. Many a jibe at the time characterized the ponderous approach of the ISO-OSI approach as "vapourware about paperware," while the IP effort, which was forming around the newly formed *Internet Engineering Task Force* (IETF), proclaimed itself to work on the principle of "rough consensus and running code."

Local- and Wide-Area Networking

The rise of Ethernet networks on campuses and in the corporate world in the late 1980s also brought into stark visibility the distinction between local- and wide-area networking.

In the local-area network, Ethernet created a new environment of "seamless connectivity." Any device on the network could provide services to any other device, and the common asset of a 10-Mbps network opened up a whole new set of computing possibilities. Data storage could be thought of as a networked resource, so desktop computers could access a common storage area and complement it with local storage, and do so in a way that the distinction between local resources and shared networkwide resources was generally invisible. The rich computing environment of visualizing the application, popularized by both the Macintosh and Windows 2.0, complemented a rich networked environment where rather than bringing a user into the location that had both the data and the computing resources, the model was invested, and the user was able to exclusively use the local environment and access the remote shared resources through networking capabilities integrated into the application environment. Local-area networking was now an abundant resource, and the industry wasted no time on exploiting this new-found capability.

But as soon as you wanted to venture further than your *Local-Area Network* (LAN), the picture changed dramatically. The wide-area networking world was provisioned on the margins of oversupply of the voice industry, and the services offered reflected the underlying substrate of a digital voice circuit. The basic unit of a voice circuit was a 64-kbps channel, which was "groomed" into a digital circuit of either 56 or 48 kbps, depending on the particular technology approach used by the voice carrier. Higher capacities (such as 256 or 512 kbps) were obtained by multiplexing individual circuits together. Even high-capacity circuits were obtained by using a voice trunk circuit, which was either 1.5 (T1) or 2.048 Mbps (E1), again depending on the digital technology used by the voice carrier. Whereas the LANs were now supporting an any-to-any mode of connection, these *Wide-Area Networks* (WANs) were constructed using point-to-point technologies that were either statically provisioned or implemented as a form of "on-demand" virtual circuit (X.25).

In the late 1980s users' patience was running thin over having to use an entirely different protocol suite for the wide area as distinct from the local area. Often the wide area required the use of different applications with different naming and addressing conventions. One approach used by many Ethernet switch vendors was to introduce the concept of an *Ethernet Serial Bridge*. This technology allowed a logical IEEE 802.3 Ethernet to encompass much larger geographic domains, but at the same time protocols that worked extremely efficiently in the local area encountered significant problems when passed through such supposedly "transparent" Ethernet serial bridges.

However, these bridge units allowed significantly larger and more complex networks to be built using Ethernet as the substrate. The Ethernet *Spanning Tree Algorithm* gained traction in order to allow arbitrary topologies of interconnected LANs to self-organize into coherent topologies that eliminated loops and allowed for failover resilience in the network.

What has changed, and what has stayed the same?

So what have we learned from this time?

In the intervening period ISO-OSI waned and eventually disappeared, without ever having enjoyed widespread deployment and use. Its legacy exists in numerous technologies, including the X.500 Directory Service, which is the substrate for today's *Lightweight Directory Access Protocol* (LDAP) Directory Services. Perhaps the most enduring legacy of the ISO-OSI work is the use of the "layered stack" conceptual model of network architectures. These days we refer to "Layer 2 Virtual LANs (VLANs)" and "Layer 3 Virtual Private Networks (VPNs)" perhaps without appreciating the innate reference to this layered stack model.

Of course the ISO-OSI protocol suite was not the only casualty of time. DECnet is now effectively an historic protocol, and Novell's *NetWare* has also shifted out of the mainstream of networking protocols. Perhaps it may be more instructive to look at those technologies that existed at the time that have persisted and flourished so that they now sit in the mainstream of today's networked world.

Ethernet has persisted, but today's Ethernet networks share little with the technology of the original IEEE 802.3 *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD) 10-Mbps common bus network. The entire common bus architecture has been replaced by switched networks, and the notion of self-clocking packets was discarded when we moved into supporting Gbps Ethernets. What has persisted is the IEEE 802.3 packet frame format, and the persistence of the 1500-octet packet as the now universal lowest common factor for packet quantization on today's network. Why did Ethernet survive while other framing formats, such as *High-Level Data Link Control* (HDLC), did not?

I could suggest that it was a triumph of open standards, but HDLC was also an open standard. I would like to think that the use of a massive address space in the Ethernet frame, the 48-bit *Media Access Control* (MAC) address, and the use since its inception of a MAC address registry that attempted to ensure the uniqueness of each Ethernet device were the most critical elements of the longevity of Ethernet.

Indeed not only has UNIX persisted, it has proliferated to the extent that it is ubiquitous, because it now forms the foundation of the Apple and Android products. Of the plethora of operating systems that still existed in the late 1980s, it appears that all that have survived are UNIX and Windows, although it is unclear how much of Windows 2.0 still exists in today's Windows 7, if anything.

And perhaps surprisingly TCP/IP has persisted. For a protocol that was designed in the late 1970s, in a world where megabits per second was considered to be extremely high speed, and for a protocol that was ostensibly experimental, TCP/IP has proved to be extremely persistent. Why? One clue is in the restrained design of the protocol, where, as we have noted, TCP/IP did not attempt to solve every problem or attempt to be all things for all possible applications. I suspect that there are two other aspects of TCP/IP design that contributed to its longevity.

The first was a deliberate approach of modularity in design. TCP/IP deliberately pushed large modules of functions into distinct subsystems, which evolved along distinct paths. The routing protocols we use today have evolved along their own paths. Also the name space and the mapping system to support name resolution has evolved along its own path. Perhaps even more surprisingly, we have had the rate control algorithms used by TCP, the workhorse of the protocol suite, evolve along its own path.

The second aspect is use of what was at the time a massively sized 32-bit address space, and an associated address registry that allowed each network to use its own unique address space. Like the Ethernet 48-bit MAC address registry, the IP address registry was, in my view, a critical and unique aspect of the TCP/IP protocol suite.

Failures

What can we learn from the various failures and misadventures we have experienced along the way?

Asynchronous Transfer Mode (ATM) was a technology that despite considerable interest from the telephone operators proved to be too little too late, and was ultimately swept aside in the quest for ever larger and ever cheaper network transmission systems. ATM appeared to me to be perhaps the last significant effort to invest value into the network through allowing the network to adapt to the various differing characteristics of applications.

The underlying assumption behind this form of adaptive networking is that attached devices are simply incapable of understanding and adapting to the current state of the network, and it is up to the network to contain sufficient richness of capability to present consistent characteristics to each application. However, our experience has been quite the opposite, where the attached devices are increasingly capable of undertaking the entire role of service management, and complex adaptive networks are increasingly seen as at best meaningless duplication of functions, and at worst as an anomalous network behavior that the end device needs to work around. So ATM failed to resonate with the world of data networking, and as a technology it has waned. In the same way subsequent efforts to equip IP networks with *Quality of Service* (QoS) responses, or the much-hyped more recent *Next-Generation Networking* (NGN) networking efforts have been failures, for much the same basic reasons.

Fiber Distributed Data Interface (FDDI) also came and went. Rings are notoriously difficult to engineer, particularly in terms of managing a coherent clock across all attached devices that preserves the circumference of the ring, as measured in bits on the wire. From its earlier lower-speed antecedents in the 4-Mbps token, the 100-Mbps FDDI ring attracted considerable interest in the early 1990s. However, it was in effect a dead end in terms of longer-term evolution—the efforts to increase the clock speed required either the physical diameter of the ring to shrink to unusable small diameters or the clock signal to be locked at extraordinarily high levels of stability that made the cost of the network prohibitive. This industry appears to have a strong desire for absolute simplicity in its networks, and even rings have proved to be a case of making the networks too complex.

Interestingly, and despite all the evidence in their favor, the industry is still undecided about open technologies. TCP/IP, UNIX, and the Apache web platform are all in their own way significant and highly persuasive testaments to the power of open-source technologies in this industry, and a wide panoply of open technologies forms the entire foundation of today's networked environment. Yet, in spite of all this accumulated experience, we still see major efforts to promote closed, vendor-specific technologies into the marketplace. Skype is a case in point, and it is possible to see the iPhone and the Kindle in a similar light, where critical parts of the technology are deliberately obscured and aspects of the device behavior are deliberately sealed up or occluded from third-party interception.

The Next Twenty-Five Years

In wondering about the next 25 years, it may be interesting to look back ever further, to the early 1960s, and see what, if anything, has proved to be enduring from the perspective of the past 50 years. Interestingly, it appears that very little of that time, except for the annoying persistence of Fortran, and the ASCII keyboard as the ubiquitous input device, is still a part of today's networked environment. So over a 50-year time period much has changed in our environment.

But, interestingly, when we par down the period to the past 25 years, there is still much that has survived in the computing and networking environment. A Macintosh computer of the late 1980s looks eerily familiar, and although today's systems are faster, lighter, and a lot less clunky, there is actually very little that has changed in terms of the basic interface with the user. A Macintosh of that time could be connected to an Ethernet network, and it supported TCP/IP, and I suspect that if one were to resurrect a Mac system from 1988 loaded with *MacTCP* and connect it to the Internet today it would be frustratingly, achingly slow, but I would like to think that it would still work! And the applications that ran on that device have counterparts today that continue to use the same mechanisms of interaction with the user.

So if much of today's world was visible 25 years ago, then where are the aspects of change? Are we just touching up the fine-point details of a collection of very well established technologies? Or are there some basic and quite fundamental shifts underway in our environment?

It seems to me that the biggest change is typified in today's tablet and mobile phone computers, and in these devices it is evident that the metaphors of computing and interaction with applications are changing. The promise from 1968 in the film *2001: A Space Odyssey* of a computer that was able to converse with humans is now, finally, within reach of commodity computing and consumer products. But it is more than merely the novelty of a computer that can "talk." The constant search for computing devices that are smaller and more ubiquitous now means that the old paradigm of a computer as a "clever" but ultimately bulky typewriter is fading away. Today we are seeing modes of interaction that use gestures and voice, so that the form factor of a computer can become smaller while still supporting a functional and efficient form of interaction with the human user.

It is also evident that the pendulum of distribution and centralization of computing capability is swinging back, and the rise of the heavily hyped *Cloud*[5, 6] with its attendant collection of data centers and content distribution networks, and the simultaneous shrinking of the end device back to a "terminal" that allows the user to interact with views into a larger centrally managed data store held in this cloud, appears to be back in vogue once more.

It is an open question whether these aspects of today's environment will be a powerful and persistent theme for the next 25 years, or whether we will see other aspects of our environment seize industry momentum, so they are very much just a couple of personal guesses. Moore's Law has proved to be truly prodigious over the past 50 years. It has allowed us to pack what was a truly unbelievable computing capability and storage into astonishingly small packages and then launch them into the consumer market with pricing each year that appears to be consistently lower than the previous year.

If this property of packaging ever greater numbers of transistors into silicon chips continues for the next 25 years at the same rate, then it is likely that whatever happens in the next 25 years, the only limitation may well be our imagination rather than any intrinsic limitations of the technology itself.

For Further Reading

[0] The Charles Babbage Institute at the University of Minnesota has scanned the complete collection of *ConneXions—The Interoperability Report,* and it is available at this URL: **http://www.cbi.umn.edu/hostedpublications/Connexions/index.html**

[1] Starting in April 1989 (Volume 3, No. 4), *ConneXions* published a long-running series of articles under the general heading "Components of OSI," which described almost every aspect of this protocol suite. The same journal also published articles on many of the other technologies mentioned in this article, including FDDI, AppleTalk, and ATM.

[2] Vint Cerf, "A Decade of Internet Evolution," *The Internet Protocol Journal,* Volume 11, No. 2, June 2008.

[3] Geoff Huston, "A Decade in the Life of the Internet," *The Internet Protocol Journal,* Volume 11, No. 2, June 2008.

[4] International Organization for Standardization, "Final text of DIS 8473, Protocol for Providing the Connectionless-mode Network Service," RFC 994, March 1986.

[5] T. Sridhar, "Cloud Computing—A Primer Part 1: Models and Technologies," *The Internet Protocol Journal,* Volume 12, No. 3, September 2009.

[6] T. Sridhar, "Cloud Computing—A Primer Part 2: Infrastructure and Implementation Topics," *The Internet Protocol Journal,* Volume 12, No. 4, December 2009.

GEOFF HUSTON B.Sc., M.Sc., is the Chief Scientist at *Asia Pacific Network Information Centre* (APNIC), the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of numerous Internet-related books, was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of Trustees of the Internet Society from 1992 until 2001. E-mail: **gih@apnic.net**

Letter to the Editor

Dear Editor,

Who knew? Twenty-five years ago I started a tiny company that grew into Interop to spread the technical word about this funny thing we called *The Internet* and this really obscure thing called "TCP/IP." Back in the '70s, when the basic protocols were being created and experimented with, you were a high school kid in Norway and I was running a tiny group at SRI International and I let you use my machine across the ocean by using the ARPANET, the precursor to the Internet, because you seemed both smart and polite. Fifteen years later I decided to hire you to start a newsletter, *ConneXions—The Interoperability Report,* about the burgeoning Internet because of those properties and the perceived need to communicate monthly about the ins and outs of these simple but far-reaching technical protocols. You had the technical knowledge and good sense to enlist the brains of the real engineers in the field with real experience to further the knowledge of "all things Internet."

Who knew this would be still going on 25 years later? Your combination of passion and patience has produced an amazing record of ongoing expertise for the whole world to enjoy.

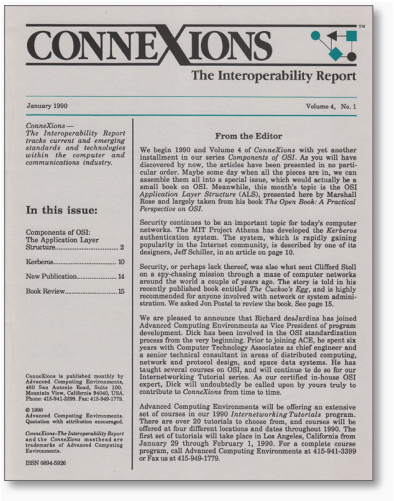
Thank you for being Ole!

—Dan Lynch
Founder of Interop
dan@lynch.com

Thank you, Dan!

I appreciate your very kind words. I also want to take this opportunity to thank all of the contributors to this journal. We could not do this without you!

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com


CONNEXIONS
The Interoperability Report
January 1990 Volume 4, No. 1
ConneXions—
The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.
In this issue:
Components of OSI: The Application Layer Structure 2
Kerberos 10
New Publication 14
Book Review 15
From the Editor
We begin 1990 and Volume 4 of ConneXions with yet another installment in our series Components of OSI. As you will have discovered by now, the articles have been presented in no particular order. Maybe some day when all the pieces are in, we can assemble them all into a special issue, which would actually be a small book on OSI. Meanwhile, this month's topic is the OSI Application Layer Structure (ALS), presented here by Marshall Rose and largely taken from his book The Open Book: A Practical Perspective on OSI.
Security continues to be an important topic for today's computer networks. The MIT Project Athena has developed the Kerberos authentication system. The system, which is rapidly gaining popularity in the Internet community, is described by one of its designers, Jeff Schiller, in an article on page 10.
Security, or perhaps lack thereof, was also what sent Clifford Stoll on a spy-chasing mission through a maze of computer networks around the world a couple of years ago. The story is told in his recently published book entitled The Cuckoo's Egg, and is highly recommended for anyone involved with network or system administration. We asked Jon Postel to review the book. See page 15.
We are pleased to announce that Richard desJardins has joined Advanced Computing Environments as Vice President of program development. Dick has been involved in the OSI standardization process from the very beginning. Prior to joining ACE, he spent six years with Computer Technology Associates as chief engineer and a senior technical consultant in areas of distributed computing, network and protocol design, and space data systems. He has taught several courses on OSI, and will continue to do so for our Internetworking Tutorial series. As our certified in-house OSI expert, Dick will undoubtedly be called upon by yours truly to contribute to ConneXions from time to time.
Advanced Computing Environments will be offering an extensive set of courses in our 1990 Internetworking Tutorials program. There are over 20 tutorials to choose from, and courses will be offered at four different locations and dates throughout 1990. The first set of tutorials will take place in Los Angeles, California from January 29 through February 1, 1990. For a complete course program, call Advanced Computing Environments at 415-941-3399 or Fax us at 415-949-1779.
ConneXions is published monthly by Advanced Computing Environments, 480 San Antonio Road, Suite 100, Mountain View, California 94040, USA. Phone: 415-941-3399. Fax: 415-949-1779.
© 1990 Advanced Computing Environments. Quotation with attribution encouraged.
ConneXions-The Interoperability Report and the ConneXions masthead are trademarks of Advanced Computing Environments.
ISSN 0894-5926


The Internet Protocol Journal
December 2011 Volume 14, Number 4
A Quarterly Technical Publication for Internet and Intranet Professionals
In This Issue
From the Editor 1
Port Control Protocol 2
Challenges to DNS Scaling 9
Networking @ Home 15
IETF Tools 21
Fragments 25
Call for Papers 31
From The Editor
Depletion of the IPv4 address space and the transition to IPv6 has been a "hot topic" for several years. In 2011, interest in this topic grew considerably when the Asia Pacific Network Information Centre (APNIC) became the first Regional Internet Registry (RIR) to start allocating addresses from its final /8 IPv4 address pool. Although depletion dates are difficult to predict accurately, there is no question that the day will come when it will no longer be possible to obtain IPv4 space from the RIRs. News stories about IP addresses being sold for considerable sums of money are becoming more common.
Numerous organizations have been working diligently to promote, test, and deploy IPv6 through efforts such as the World IPv6 Day, while the Internet Engineering Task Force (IETF) continues to develop solutions to aid in the transition. One such effort, the Port Control Protocol (PCP), is described in our first article by Dan Wing.
The Internet Corporation for Assigned Names and Numbers (ICANN) will soon begin accepting applications for new Top-Level Domains (TLDs). It is not yet known how many new TLDs will eventually be deployed, but the plans have prompted several studies focused on the resiliency and scalability of the Domain Name System (DNS). Bill Manning discusses some of the technical challenges associated with a vastly expanded TLD space.
The IETF Homenet Working Group "...focuses on the evolving networking technology within and among relatively small 'residential home' networks. For example, an obvious trend in home networking is the proliferation of networking technology in an increasingly broad range and number of devices. This evolution in scale and diversity sets some requirements on IETF protocols." Geoff Huston gives an overview of some of the challenges facing this Working Group.
The product of the IETF is a set of documents, mainly protocol specifications and related material. These documents start life as Internet Drafts and proceed through a series of iterative refinements toward eventual publication as Request For Comments (RFCs). Over time, several tools have been developed to aid in the document development process, and they are now organized at the IETF Tools webpage. We asked Robert Sparks to give us an overview of some of the most important tools and the process involved in their development.
—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com
You can download IPJ back issues and find subscription information at: www.cisco.com/ipj
ISSN 1944-1134

Fragments

NIXI to Run New NIR in India

March saw the launch of a new *National Internet Registry* (NIR) for India, following the successful conclusion of talks between the *Asia Pacific Network Information Centre* (APNIC) and the Government of India.

The *Indian Registry For Internet Names And Numbers* (IRINN) will be run by the *National Internet Exchange of India* (NIXI) and serve ISPs within the country that wish to sign up. It is the result of a long collaboration between APNIC and NIXI, with APNIC staff sharing their expertise with NIXI, and NIXI officials putting together an impressive technical installation in preparation for the launch. The new registry was announced on the final day of APNIC 33, a technical conference conducted in conjunction with the *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT 2012).

APNIC Executive Council Chairman, Akinori Maemura said of the announcement, "We are extremely happy that this process is heading towards a positive conclusion; which, on the other hand, is also a commencement of a new relationship. I would like to thank the NIXI team for their support and the hard work they have demonstrated in making this a reality."

Director General of APNIC, Paul Wilson commented, "We welcome the new National Internet Registry in India to the APNIC community. The Internet is a global community and IRINN, as the NIR is being called, should be part of that. I hope that many new Internet Services Providers will be formed in India, and they will always be able to choose between IRINN and APNIC for IP addresses. The market here is big enough and that kind of diversity will ensure better services and lower prices for all Indians."

APNIC has over 300 members locally, mostly Internet Services Providers and Telecommunication Communications companies, and over 6 million *Internet Protocol version 4* (IPv4) addresses were allocated in 2011. There are already 6 National Internet Registries in Asia in South Korea: (KISA KRNIC), Japan (JPNIC), China (CNNIC), Indonesia (IDNIC), Vietnam (VNNIC) and Taiwan (TWINIC). This is out of 56 economies in the Asia Pacific region.

"It's really about what is a better fit for the individual organizations. Typically we tend to see larger organizations prefer a regional service, especially those who operate in multiple economies to maintain an account with APNIC," said Paul Wilson.

NIXI is a not-for-profit organization, set up for peering of ISPs among themselves for the purpose of routing domestic traffic within India, instead of routing it through international peering points, thereby resulting in reduced latency and reduced bandwidth charges for ISPs. NIXI is managed and operated on a neutral basis, in line with the best practices for such initiatives globally.

Internet Hall of Fame Advisory Board Named

The *Internet Society* (ISOC) recently announced that in conjunction with its 20th anniversary celebration, it is establishing an annual *Internet Hall of Fame* program to honor leaders and luminaries who have made significant contributions to the development and advancement of the global Internet.

Inaugural inductees will be announced at an Awards Gala during the ISOC's *Global INET 2012* conference in Geneva, Switzerland, April 22–24, 2012, **www.internetsociety.org/globalinet**

"There are extraordinary people around the world who have helped to make the Internet a global platform for innovation and communication, spurring economic development and social progress," noted ISOC CEO Lynn St. Amour. "This program will honor individuals who have pushed the boundaries to bring the benefits of a global Internet to life and to make it an essential resource used by billions of people. We look forward to recognizing the achievements of these outstanding leaders."

ISOC has convened an Advisory Board to vote on the inductees for the 2012 Internet Hall of Fame inauguration. The Advisory Board is a highly-qualified, diverse, international committee that spans multiple industry segments and backgrounds. This year's Advisory Board members include:

- Dr. Lishan Adam, ICT Development Researcher, Ethiopia
- Chris Anderson, Editor-in-Chief, *WIRED Magazine*
- Alex Corenthin, Directeur des Systemes d'Information, University Cheikh Anta Diop of Dakar/Chair, Internet Society Senegal Chapter
- William Dutton, Professor of Internet Studies, Oxford Internet Institute
- Joichi Ito, Director, MIT Media Lab
- Mike Jensen, Independent ICT Consultant, South Africa
- Aleks Krotoski, Technology Academic/Journalist/Author
- Loic Le Meur, Founder & CEO, LeWeb
- Mark Mahaney, Internet Analyst, Citigroup
- Dr. Alejandro Pisanty, Professor at National University of Mexico/ Chair of Internet Society Mexico Chapter
- Lee Rainie, Director, Pew Research Center's Internet & American Life Project
- Jimmy Wales, Co-founder, Wikipedia

"We are extremely grateful to our distinguished Advisory Board members who have donated their time, energy, and expertise to this program," St. Amour added. "The breadth of their experiences and the diversity of their perspectives are invaluable, and we truly appreciate their participation."

The Internet Society is the trusted independent source for Internet information and thought leadership from around the world. With its principled vision and substantial technological foundation, the Internet Society promotes open dialogue on Internet policy, technology, and future development among users, companies, governments, and foundations. Working with its members and Chapters around the world, the Internet Society enables the continued evolution and growth of the Internet for everyone.

For more information, see: **http://www.internetsociety.org**

IETF Journal Now Available by Subscription

The *IETF Journal* provides anyone with an interest in Internet standards an overview of the topics being debated by the *Internet Engineering Task Force* (IETF), and also helps facilitate participation in IETF activities for newcomers.

The *IETF Journal* aims to provide an easily understandable overview of what is happening in the world of Internet standards, with a particular focus on the activities of the IETF Working Groups. Each issue highlights hot issues being discussed in IETF meetings and on the IETF mailing lists.

Visit *The IETF Journal* on the Web at **www.internetsociety.org/ietfjournal** to see the latest edition, or to subscribe to the e-mail edition or have it delivered as a hardcopy, visit:
http://www.internetsociety.org/ietfjournal-subscribe

The *IETF Journal* is an Internet Society publication produced in cooperation with the IETF.

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

June 2012 Volume 15, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: www.cisco.com/ipj

ISSN 1944-1134

From The Editor

Deployment of IPv6 took another step forward on June 6, 2012, when numerous website operators, *Internet Service Providers* (ISPs), and home router vendors participated in the *World IPv6 Launch*. Organized by the Internet Society, the event attracted significant media attention as the participants enabled IPv6 permanently and rendered it "on by default." More information about the event is available from www.worldipv6launch.org

Migration to IPv6 is not a simple task, as outlined in many previous editions of this journal. Various tools and techniques have been developed, one being the use of so-called *Carrier-Grade NATs* whereby the end customers connect to the Internet using private (RFC 1918) addresses and the ISP provides translation for both public IPv4 and IPv6 addresses. In April of this year, the *Internet Engineering Task Force* (IETF) approved and the *Internet Assigned Numbers Authority* (IANA) allocated a new IPv4 address block (100.64.0.0/10), designated for use as "Shared Transition Space" in support of the IPv6 transition. We asked Wesley George to describe the rationale behind the use of this additional private address space and discuss the debate that resulted from this allocation.

The world of telecommunications has changed dramatically as a result of the rapid expansion of the Internet. Traditional telephone lines are being replaced by *Voice over IP* (VoIP) systems for both private and business use. These changes represent big challenges for traditional telephone carriers, and even for some countries whose income used to depend largely on telephone "settlement charges" for international phone calls. The *World Conference on International Telecommunications* (WCIT) will take place this coming December in Dubai. Geoff Huston discusses some of the proposed changes to the *International Telecommunication Regulations* that could affect the Internet in various ways and will be discussed at WCIT.

The IETF is concerned not only with IPv4-to-IPv6 migration, but also with recovery upon router or link failure. In our final article, Russ White describes *IP Fast Reroute,* a technique for providing fast traffic recovery when these failures occur.

As always, your feedback about anything you read in this journal is most appreciated. Please contact us at ipj@cisco.com and don't forget to renew your subscription and provide us with any postal or e-mail changes.

—Ole J. Jacobsen, Editor and Publisher

ole@cisco.com

Shared Transition Space: Is it necessary?

by Wesley George, Time Warner Cable

Recently, the *Internet Engineering Task Force* (IETF) approved[1] and the *Internet Assigned Numbers Authority* (IANA) allocated[2] a new IPv4 address block (`100.64.0.0/10`) designated for use as "Shared Transition Space" in support of the IPv6 transition. This decision was highly controversial within the different standards and policy bodies that discussed the idea. The author would like to note that people have been debating this topic for years, and nearly everyone within the broad stakeholder community seems to have a strong opinion on the matter, including me. Despite the best of intentions, some of my opinions and biases may appear within the article. I did not intend this article to be a definitive conclusion on the matter, but rather a summary of the recent discussion. Whether the standards bodies involved came to the "right" or "wrong" conclusion—as well as the veracity of the arguments on both sides—is an exercise for you, the reader.

Internet Service Providers (ISPs) and users have significant investments in equipment and applications that must be updated to support IPv6. Progress is accelerating with regard to IPv6 availability in hardware, software, and access, though broad availability remains a long-term problem. In the interim, IPv4 will continue to be an important capability for providing users with access to Internet resources. As a consequence, considerable effort has been expended in conserving the increasingly scarce IPv4 resources while maintaining "business as usual." This conservation has taken the form of policies for address allocation and management[3], as well as new protocols and technologies. It is likewise important to note that ISPs must manage IPv4 exhaustion in a way that is least disruptive to users while undertaking full IPv6 deployment—two completely different and parallel activities. Any business that relies entirely on efforts to extend the useful life of IPv4 without executing on an IPv6 deployment plan is merely delaying the inevitable effects on their customers and ultimately their profitability.

IPv4 "life extension" is an area that remains controversial. Some believe that any effort to extend the useful life of IPv4 and allow the IPv4 Internet to keep growing beyond its original design limitations will seriously affect the timeliness of reaching critical mass with IPv6. The idea that many opponents of the "life-extensions" methods are supporting is that IPv4 exhaustion and the resulting transition from IPv4 to IPv6 is going to be disruptive to customers and operations no matter when it actually occurs. From this perspective it is preferable to have a brief—but significant—disruption and transition completely to IPv6. This plan is akin to the idea that it is better to just rip the bandage off and have a moment of pain than removing it slowly in an attempt to reduce the pain.

The counterpoint to this argument is that we must look at the situation pragmatically with the goal of maintaining business continuity, growth, and customer satisfaction.

IPv4 Exhaustion

The impending IPv4 address exhaustion[4] and the problems it will create has been the topic of much discussion in many different areas of the Internet community. The need to deploy IPv6 has figured prominently in the discussion, because it is the proper long-term solution. However, the unfortunate reality is that deploying IPv6 is a parallel activity to any work that provides continuity to the existing IPv4 network in order to keep it operational and able to grow to meet demand. As an Internet community, we are not where we need to be in terms of critical mass of our IPv6 deployments, in terms of either available, deployed equipment that supports IPv6 fully or applications that are able to use IPv6 when it is available.

IPv6 deployment is a requirement, but most ISPs do not have control over all variables affecting IPv6 deployment, and they have limited influence on progress outside of their network boundaries. This reality is especially true with residential services, where customers often purchase IP-enabled hardware directly from retailers to connect to their home networks. Consumers generally do not care about whether a device supports IPv4 or IPv6, so they do not make purchasing decisions based on such features. Customers should not be required to be technology experts in order to get their devices to work properly for their intended use. Customers generally are not interested in their ISP dictating the equipment that they may use in their home, and they do not like being told that they must replace "obsolete" gear, especially if they purchased it recently. The service provider sells "Internet" service, so customers expect their "Internet" devices to work—period. As a result, if an ISP wants to continue to grow, that ISP must continue to offer IPv4 services until the existing equipment without IPv6 support ages out of the network and is replaced.

The IETF recently released a *Best Current Practice* (BCP) document[5] that provides some guidance for implementers that support for IPv6 on "IP-capable" devices is going to be a necessity, and the *Consumer Electronics Association* (CEA) now has a working group on IPv6 Transition[6]. In conjunction with events like *World IPv6 Launch*[7], there are near-constant improvements in the availability of IPv6-capable hardware, software, access, and services. The result of this situation should be that critical mass of IPv6 deployment will happen soon and reduce reliance on IPv4 and IPv4 life-extension technologies.

Because of the costs, operational complexities, performance concerns, and effects on customers that most IPv4 life-extension technologies create, service providers should focus on reaching IPv6 critical mass in essential areas.

When IPv6 has become sufficiently ubiquitous, the need for IPv4 life-extension technologies will be reduced along with the scale of deployments. Because a lot of the costs of deploying IPv4 life-extension technologies are initial costs, there is some truth to the argument that after they are deployed they are unlikely to disappear anytime soon. Why would a carrier invest significant time and money in deploying something only to pull it back out a short time later? Therefore the best method to reduce the cost of *Carrier-Grade NAT* (CGN) deployment is to work to deploy less of it.

ISPs are different when it comes to their expectations for growth, and their IPv4 addressing reserves or consumption rates differ accordingly. Some have areas of their internal network where they can make changes and reclaim globally unique IPv4 addresses for reuse to support customers, some have addresses that can be reclaimed via auditing and improved efficiency of allocation, and still others have already undertaken many of these projects and do not have much address space left to reclaim. Further, new IPv4 address availability as a combination of policies and demand may be different for each *Regional Internet Registry* (RIR). To summarize, the need for IPv4 address life-extension technologies is different on each network. The costs of deploying, the complexity of supporting, and the growth rate all figure into how widely service providers will have to deploy one or more technologies to extend their remaining IPv4 resources.

NAT444

Network Address Translation (NAT)[28, 30] is already widely used for translating one IPv4 address to another, usually to provide separation or address sharing between a private network with multiple hosts and a public network or the Internet. In the context of IPv4 and IPv6 transition, these types of NAT are commonly referred to as *NAT44,* because they translate between IPv4 and IPv4 (vs. IPv4 to IPv6, IPv6 to IPv6, etc.). There is a proposed extension to NAT intended to preserve even more IPv4 resources. This proposal is called *Carrier-Grade NAT* (CGN)[8]. The "Carrier Grade" in the name originates from the position of the NAT within the topology. Instead of NAT between a private and public network at the edge of a single network such as a home or business office, CGN is implemented inside of an ISP's network and serves many customers simultaneously. These CGN implementations are typically scaled to handle thousands of simultaneous customer endpoints, often resulting in millions of simultaneous sessions. The RFC[8] does not advocate the use of CGN; it describes how an ISP forced to deploy CGN can use it during IPv6 transition.

This sort of implementation addresses the need for an individual, globally unique IPv4 address for each of the ISP's customers by allowing the ISPs to allocate each customer an IPv4 address that may not be globally unique and employ NAT to give them access to resources on the IPv4 Internet.

This sharing often allows ISPs to see oversubscription of public IPv4 addresses anywhere from 2:1 to more than 10,000:1 based on the type of applications behind the NAT and their simultaneous application layer port allocations and session counts. Most commonly, a CGN is used in conjunction with a local NAT on the customer's home network, creating two layers of NAT to traverse between the home network and the Internet. This model is commonly referred to as *NAT444,* because there is a translation layer between three sets of IPv4 addresses end to end.

A known problem with NAT is that it makes end-to-end communication and visibility between hosts more difficult, because it essentially hides hosts behind address translation. Because NAT is so common (nearly every home network and many commercial networks use NAT), networking applications have adapted so that they can discover the presence of a NAT and then change their behavior in order to maintain communications in the presence of NATs. However, the addition of this second layer of NAT often interferes with those workarounds, and undesirable or unpredictable results may occur[9].

Over time it is likely that applications will again adapt to the impediments created by multiple layers of NAT, but it is not possible to anticipate and correct every potential problem that may be generated by adding this second layer of NAT. This reality should serve as a warning to those who provide services over an Internet connection: IPv6 support is extremely important. IPv6 is important because CGN means that ISP-controlled equipment will be actively involved in the path between content or application providers and their end users, making that relationship reliant on the service provider and the service provider's CGN vendor to an extent that was not necessary in the past. If the CGN implementation breaks something, it not only reflects on the CGN vendor and the service provider, it also reflects poorly on the relationship between the end customer and the service that that customer is using—and may cause that customer to form a negative opinion of the brand itself.

In other words, if a consumer uses an Internet-enabled application on a new Brand X smart TV and it does not work well, regardless of whether it is a problem with the CGN, the service provider, or something else entirely, the consumer may form the opinion and share via an online review that, "Brand X's TVs are ok, unless you try to use any of their fancy new features. I would not buy one if I were you, because Company X clearly does not know what it is doing." CGN represents a potentially significant increase in the amount of testing that must be done, especially in implementations that are uncommon, such as small, corner-case deployments, and closed architectures. Although using IPv6 is dependent on support at the client, the content or application provider, and the ISPs in between, if this support is present, it allows the content or application provider and client to bypass the service provider's CGN machinery—as well as any IPv4 NAT that may be present—and have a true end-to-end connection. This scenario restores control over the user experience back to the brand, and allows the ISP to resume supplying bit carriage.

IPv4 Addressing Requirements

Independent of the potential connectivity problems that NAT444 may create, it generates additional problems for the implementing ISP because of its need for IPv4 addresses. Because the CGN requires two sets of addresses—one for the inside (private) network and one for the outside (public) network—the ISP must identify address ranges to use for both. In order for its customers to be able to reach the Internet, the external pool must use globally unique IPv4 addresses. The number of addresses required will depend on the implementation of CGN, its scale profile, the topology of the network (how many hosts are behind each CGN instance), and the usage profile of the customer traffic. If the service provider has few or no available globally unique IPv4 addresses, it will have to either make changes in its network in order to reclaim addresses from elsewhere or make a request for a new allocation from its RIR[29].

However, depending on the number of addresses that the RIR has available and its policies for justification, it may not be possible to obtain sufficient address space with this method. For example, in the Asia-Pacific region, the austerity policies in place mean that no matter how many IPv4 addresses they might have been able to justify using previous rules, most requesters are eligible for only a few hundred IPv4 addresses as their final allocation ever[10]. This situation then requires the ISP to source IPv4 addresses via the IPv4 address transfer market[11], adding additional cost to an already expensive deployment. In fact, if the service provider must source addresses via the transfer market, it may be more cost-effective to simply obtain more addresses and continue with business as usual without deploying CGN at all.

Internal Pool: Private Addressing Alternatives

When addresses are sourced for the public address pool, the service provider must also identify a pool of private addresses that is large enough for the provider to allocate one to each customer behind the CGN. Depending on the size and scale of the CGN, and how much the service provider is willing to segment and separate different sections of its network, this number could be a large block of addresses, perhaps even a `/8` or more.

The most obvious choice might be to simply use address ranges reserved for private network use[12], because there is a `/8`, a `/12`, and a `/16` available for this purpose. However, this address space has some drawbacks. First, because of the prevalence of RFC 1918 addressing within most enterprise networks, there is a significant chance that the chosen address blocks may conflict with existing use of RFC 1918 space for management systems and other internal resources. Depending on the size of the CGN implementation, it may be necessary to instantiate multiple segments of the network where the entirety of RFC 1918 space is used, and in order for those segments to talk to one another or to talk to devices with conflicting numbering, significant additional complexity is required.

On the customer side, remote workers could experience problems where the address that they have been assigned is in a block that is already in use on their company's enterprise network, meaning that it may cause problems connecting to those hosts via a *Virtual Private Network* (VPN), or problems accessing some of the resources from the remote network. It may be possible to change the address assigned to the end user in an attempt to eliminate this conflict, but this approach is not necessarily scalable because it likely requires manual intervention in an automated address-assignment system, and there are limits to the number of times that a change of address can "fix" this problem without creating a problem for another user.

The other problem with the use of RFC 1918 space in the CGN is that it may conflict with the address space used by the customer's local network and NAT. For example, if a customer has a local network numbered out of `192.168.1.0/24` and the customer's router is allocated the external address of `192.168.1.85`, the router may fail to function properly because it has the same address range on both the internal and external interface. It may be possible through analysis to identify and carefully allocate addresses so that the portions of RFC 1918 commonly used by default in home gateway devices are not allocated. However, anecdotal evidence[13] suggests that because of the wide variety of devices and implementations available—plus the fact that many users reconfigure their networks to use a different IP address range than the default configuration of the device—there simply may not be enough RFC 1918 addresses not in use to make this option viable.

"Squat" Space

Another alternative is to unofficially reuse one or more portions of the existing range of allocated globally unique IPv4 addresses as private addresses. In a network that does not talk directly to the Internet, such as a private network or VPN, the existing allocations of IPv4 space do not have any meaning, and so it is not strictly necessary to stick to RFC 1918 address space for numbering resources that are only internally accessible. Reuse of allocated IPv4 addresses has the benefit of not conflicting with in-use RFC 1918 addresses, but comes with its own set of problems. If the provider's own space is reused, the provider must carefully separate the private use from the public use to avoid conflicts, and managing this overlap may require additional complexity such as the use of VPNs as a method to separate the networks. The more common method is to reuse a block of addresses that is not currently allocated to the network using them; in other words, squatting on "someone else's" address space. Usually providers select space to use in this manner based on a low likelihood that either the owner will begin announcing the space on the global Internet or the users behind that network will need to connect to the users behind the ISP's NAT.

This method requires extreme care. The service provider must ensure that the routes for those prefixes are not inadvertently leaked to the global Internet, because such a leak could potentially cause a route-hijack denial-of-service attack, albeit an unintentional one. This method is even more risky if the ISP has one or more partners who have connections into the private portion of its network, because it may not have complete control of the announcement boundaries. Certainly there are safeguards such as tagging the announcements with *Border Gateway Protocol* (BGP) communities such as no-advertise or no-export[14], but these solutions are not always practical, and they are not completely fail-safe. Depending on the chosen address space, the effects could be significant based on the true owner of that space—no service provider really wants to risk a public relations nightmare because it inadvertently caused an outage affecting the critical infrastructure of a large government agency or multinational corporation whose space it "borrowed" and then leaked to the Internet.

As a result of the IPv4 transfer market, it is quite likely that some of the address blocks that are not visible on the global Internet today and that some consider "safer" to squat on may end up being transferred to another party who plans to begin using them on the public Internet, and potentially requiring those squatting on the space to renumber to a different address block. ISPs can mitigate this risk somewhat by selecting multiple candidate blocks that are all preconfigured in the network such that it is relatively straightforward to make a rapid change from one block to another if the current block in use suddenly becomes unacceptable. Many ISPs use this method today, but because of the risks, it cannot be considered a real solution to the problem. Further, because it essentially encourages large service providers to violate the spirit—if not the letter—of the very policies that govern IP address allocation and use, standards bodies such as the IETF or policy organizations like RIRs cannot officially recommend such a solution.

Class E Addresses

A final alternative is to repurpose the reserved space in `240/4`[2] and make it available for this use. There have been several failed attempts to repurpose this reserved space within the IETF in the past few years[15, 16]. The primary challenge with this alternative is that because the Class E space has been reserved for many years, many networking implementations are explicitly configured to reject this address space as invalid. Getting this problem fixed in software, and more importantly, getting those software upgrades deployed widely, may require a similar level of effort to that which is required to deploy IPv6, and deploying IPv6 would be a more effective use of the resources required to implement software and hardware changes.

Even in situations like a CGN where more of the implementation is under central control, this solution would be attractive only to a service provider that owns and operates the *Customer Premises Equipment* (CPE) routers for all of its customers such that it could work with a small number of vendors to get software patches to enable use of this space. Therefore this solution is also too limited in applicability to be seen as a general solution that a body like the IETF could recommend.

Shared Addresses

Although the solutions previously discussed may be acceptable in some applications, the risks and deficiencies make it necessary for other applications to find another source for the IP address blocks to be used on the private side of a CGN. It is possible to use "public" (globally unique) IPv4 addresses on the private side as well, but the challenges to obtaining additional public IPv4 addresses that were discussed previously are exacerbated by the even larger number of addresses required, so this solution is far from practical. Additionally, expecting each service provider that implements CGN to obtain its own address space for its inside pools would end up using a significant amount of the remaining IPv4 resources in a way that does not necessarily require globally unique addresses. However, because each service provider has different needs, growth rates, and applications, it is unclear that simply expecting each service provider to request space from the RIRs for its internal CGN pools would create a doomsday scenario where a few networks would use up all of the remaining available IPv4 space in a short time. Because CGN creates additional costs and complexity to implement and support, and could be viewed as "second-class" IPv4 service, most service providers are not likely to implement it across the entire network and all tiers of customers, instead preferring to implement it only as widely as absolutely necessary.

Service providers could choose to implement it only for net new customers (that is, growth above turnover); they could choose to implement it only in certain markets or for certain types of service where it is less likely to cause support problems and adversely affect the service. All of these things reduce the number of addresses that may be needed for the interior CGN address pool. Nevertheless, using globally unique addresses in an application that does not require unique addresses is not a good use of a very limited resource. That is why the idea of having a shared and reserved block of addresses specifically for use as an interior (private) pool on a CGN keeps resurfacing.

One alternative to formally reserving a shared transition space was to have a third party request a block of sufficient size from one or more of the RIRs and then make it available for use as a shared block by anyone who wishes to do so.

Given the "last /8" policies in effect at each of the RIRs, it would likely be quite difficult to justify sufficient space to be useful, and the cost involved in receiving and maintaining such a delegation would likely be prohibitive. There would also be challenges addressing potential abuse concerns.

Reserving a block via the standard IETF/IANA process meant that IETF would have a chance to document the problems and recommend best practices that must be considered when implementing something that uses this shared space. This policy would help to ensure that service providers and implementers are aware of these guidelines and recommendations. For example, many implementations make certain assumptions about address scope based on the address itself, such as assuming that RFC 1918 addresses are locally scoped, and then adapt their behavior accordingly. With things like squat space or an unofficially shared CGN space, implementers would not know that this space should be treated in a specific way, and the result may be more network breakage. The officially declared shared space must still wait for implementers to make changes to their products, and that may not always happen, but the chances are still better than if it had been done in an unofficial manner.

As you can probably see, this problem does not have a clear-cut and straightforward solution, and this situation has led to vigorous discussion within the standards and policy bodies that have discussed it. The next section gives a brief history of the activity in those bodies that ultimately led to the space being allocated.

Some History

Shared transition space proposals have been controversial each time a variant of the idea has come up for discussion. As IPv4 exhaustion became a reality and IPv6 deployment continued to lag, more people realized that IPv4 life-extension technologies such as CGN may be a necessary evil. When people saw CGN as a likely response to the gap between IPv4 exhaustion and wide IPv6 support, they began to understand the need for the shared transition space, and thus support for allocating that space has gradually grown.

Although variants of this discussion may be much older than the items discussed in the following paragraphs, this article focuses specifically on the history of the idea to allocate shared address space specifically for CGN. There was an unsuccessful proposal in 2005[17] to update RFC 1918 with an additional three /8s, but this proposal was not specifically focused on CGNs, unlike some of the other proposals. The most recent set of proposals regarding shared CGN space first came up in the APNIC Policy *Special Interest Group* (SIG) in early 2008, where Policy Proposal 058 was discussed. APNIC members abandoned the proposal and recommended that the authors take the idea to the IETF, because that is the body that typically directs IANA to reserve IP address blocks for special uses such as this one[18].

This recommendation resulted in a pair of Internet drafts[19, 20], hereafter referred to as **shirasaki** in late 2008. The draft originally requested four /8s, with a minimum size of a /12, but subsequent revisions of the draft revised the request to only one /10. The draft never gained much traction within the IETF, but the authors continued to update it to keep the discussion going. In mid-2010, a second IETF draft[21] was published, requesting that a full /8 be reserved for this purpose. It contained references to the **shirasaki** drafts, but provided additional justification and noted that a /10 may not be enough addresses for many of the large service providers.

The draft went through several revisions in the following months, eventually being replaced by a different draft[22], hereafter referred to as **draft-weil**, which reduced the /8 requested down to a /10. Attendees of the IETF 79 meeting in Beijing, China, discussed the draft across two different working groups. People expressed strong opinions both in support of and in opposition to the idea, but the draft did not achieve clear consensus. With the future of the draft unclear, one of its authors submitted policy proposal 127 to the *American Registry for Internet Numbers* (ARIN)[23]. The ARIN *Advisory Council* (AC) accepted this policy proposal as draft policy 2011-5[24] in early 2011, and vigorously discussed it with participants at the ARIN XXVII public policy meeting and with members of the mailing list. At the conclusion of the discussion, the ARIN AC recommended the policy to the ARIN board for adoption.

This discussion took on additional urgency because during this time the IANA officially announced that it had exhausted the free pool of IPv4 addresses and delegated the last of the /8s to the RIRs in accordance with policy[4]. The side effect of this exhaustion meant that it was no longer possible for IETF to direct IANA to reserve space unless IANA was directed to repurpose an existing reservation, because it had no unreserved address blocks of sufficient size to meet the request. Therefore, the IETF and one or more of the RIRs would have to work in concert to make a suitable IPv4 address block available, instead of it being solely under IETF's purview. ARIN staff reached out to the IETF's *Internet Architecture Board* (IAB) for guidance, because by strict interpretation[25], ARIN was not authorized to make this allocation by itself. IAB reaffirmed this interpretation, and recommended that the matter be brought back to the IETF for (re)consideration[26]. With this guidance, the authors revised **draft-weil-shared-transition-space-request** and reintroduced it for discussion. For a period of time, the document was split into two, with most of the long-form discussion of pros and cons being moved to a second draft[27].

As of the publication date of this article, the secondary draft has expired without progressing, but most of the important information contained there was incorporated back into **draft-weil**. The document was not adopted by any IETF Working Group. Instead, an IETF Area Director sponsored it as an individual submission.

It went through its first IETF "Last Call" to gauge consensus and receive comments in August 2011. The subsequent discussion, revisions, and secondary last calls (October 2011 and January 2012) generated hundreds of messages on the IETF discussion list and a total of 12 versions of the document before it was approved for publication in February 2012.

The reason why the debate on this shared transition space was so spirited can be traced to a few critical concerns. First, although consensus-based RFCs documenting CGN[8] were already approved, this draft allocating space specifically to facilitate its deployment became a referendum within the IETF on whether NAT444/CGN should even be used. If you believed that NAT444 and CGN were bad ideas, it was likely that you would also be against a shared transition space. From that perspective, shared transition address space provided a more complete solution to a problem that had been created by a "Bad Idea" that should not have been allowed to proceed in the first place. There was also resistance to what was deemed "waste" of the limited remaining blocks of IPv4 addresses to solve a problem that not everyone agreed was a real or important problem. Also, although IETF participants do not speak for their companies per se, this proposal had consistent support from numerous individuals employed by large residential broadband providers. As a result, some saw it as those service providers looking for a way to bail themselves out of a problem that they created by not deploying IPv6 rapidly enough to avoid having to use CGN. On the converse side of the argument, those in favor saw CGN as a largely foregone conclusion, and saw this proposal as simply a practical solution to a real problem.

The *Internet Engineering Steering Group* (IESG) ultimately sent a note to the IETF discussion list acknowledging the difficulty of coming to a decision on this matter and noting that some explanatory text would be added to RFC 6598:

> "Colleagues,
>
> The IESG has observed very rough consensus in favor of the allocation proposed in **`draft-weil-shared-transition-space-request`**. Therefore, the IESG will approve the draft. In order to acknowledge dissenting opinions and clarify the IETF position regarding IPv6, the IESG will attach the following note:
>
> > "A number of operators have expressed a need for the special purpose IPv4 address allocation described by this document. During deliberations, the IETF community demonstrated very rough consensus in favor of the allocation.
> >
> > While operational expedients, including the special purpose address allocation described in this document, may help solve a short-term operational problem, the IESG and the IETF remain committed to the deployment of IPv6."

In many ways, the final decision came down to the difference between theory and practice in the IETF's desire to make the Internet work better. Theoretically, making a CGN easier to implement has the potential to make the Internet work much more poorly, and could be seen as rewarding bad behavior (failing to deploy and support IPv6 in a timely fashion). However, in practice, making CGN harder to implement causes unnecessary pain and effort for operators and potentially for users, while having little or no effect on IPv6 deployment. Approving this shared transition space avoids the appearance that IETF is trying to punish operators or users for perceived past "sins" and helps to reinforce the idea that IETF is responsive to operational concerns and therefore still relevant to the operator community. It is unlikely that the result of this decision will have much bearing on an operator's plan for how widely, when, where, or even if it will deploy CGNs, and this article makes no such recommendations. However, I will reiterate that IPv6 is the long-term solution, and that the smallest CGN deployment possible will make for a less complex and less expensive network for the continued support of traditional IPv4 devices.

Acknowledgements

Special thanks to Ole Jacobsen for suggesting that I write this article, to Kirk Erichsen and Jason Weil for their review and comments, and to all involved in the discussion of the referenced IETF drafts and ARIN policy for giving me plenty to write about!

References

[1] Victor Kuarsingh, Chris Donley, Jason Weil, Marla Azinger, and Christopher Liljenstolpe, "IANA-Reserved IPv4 Prefix for Shared Address Space," RFC 6598, April 2012.

[2] IANA Address Assignments:
http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.txt

[3] RIR Policies triggered by IPv4 Depletion:

ARIN:
https://www.arin.net/resources/request/ipv4_countdown.html

RIPE:
http://www.ripe.net/internet-coordination/ipv4-exhaustion/reaching-the-last-8

APNIC:
http://www.apnic.net/community/ipv4-exhaustion/ipv4-exhaustion-details

LACNIC:
http://www.lacnic.net/en/politicas/manual11.html

AFRINIC:
http://www.afrinic.net/docs/policies/AFPUB-2010-v4-005-draft-05.htm

[4] Number Resource Organization (NRO), "Free Pool of IPv4 Address Space Depleted," February 2011, **http://www.nro.net/news/ipv4-free-pool-depleted**

[5] Chris Donley, Christopher Liljenstolpe, Wesley George, and Lee Howard, "IPv6 Support Required for All IP-Capable Nodes," RFC 6540, April 2012.

[6] Consumer Electronics Association IPv6 Working Group, **http://www.ce.org/Press/CurrentNews/press_release_detail.asp?id=12139**

[7] World IPv6 Launch, **http://www.worldipv6launch.org/**

[8] Sheng Jiang, Brian Carpenter, and Dayong Guo, "An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition," RFC 6264, June 2011.

[9] Chris Donley, Lee Howard, and Victor Kuarsingh, "Assessing the Impact of Carrier-Grade NAT on Network Applications," Internet Draft, work in progress, November 2011, **draft-donley-nat444-impacts-03**

[10] APNIC, "Policies for IPv4 address space management in the Asia Pacific region," **http://www.apnic.net/policy/add-manage-policy#9.10**

[11] ARIN, "Understanding the IPv4 Transfer Market," **https://www.arin.net/resources/transfers/transfer_market.html**

[12] Daniel Karrenberg, Yakov Rekhter, Eliot Lear, and Geert Jan de Groot, "Address Allocation for Private Internets," RFC 1918, February 1996.

[13] Akiro Kato, A sampling of RFC 1918 IP address usage in Japan, **http://www.ietf.org/mail-archive/web/v6ops/current/msg06187.html**

[14] Paul Traina, "BGP Communities Attribute," RFC 1997, August 1996.

[15] Vince Fuller, "Reclassifying 240/4 as usable unicast address space," Internet Draft, work in progress, March 2008, **draft-fuller-240space-02**

[16] Paul Wilson, George Michaelson, and Geoff Huston, "Redesignation of 240/4 from 'Future Use' to 'Private Use,'" Internet Draft, work in progress, September 2008, **draft-wilson-class-e-02**

[17] Tony Hain, "Expanded Address Allocation for Private Internets," Internet Draft, work in progress, February 2005, **draft-hain-1918bis-01**

[18] Shirou Niinobe, Takeshi Tomochika, Jiro Yamaguchi, Dai Nishino, Hiroyuki Ashida, Akira Nakagawa, and Toshiyuki Hosaka, "Proposal to create IPv4 shared use address space among LIRs," prop-058, January 2008, **http://www.apnic.net/policy/proposals/prop-058**

[19] Jiro Yamaguchi, Yasuhiro Shirasaki, Shin Miyakawa, Akira Nakagawa, and Hiroyuki Ashida, "NAT444 addressing models," Internet Draft, work in progress, January 2012, **draft-shirasaki-nat444-isp-shared-addr-07**

[20] Ikuhei Yamagata, Shin Miyakawa, Akira Nakagawa, Jiro Yamaguchi, and Hiroyuki Ashida, "ISP Shared Address," Internet Draft, work in progress, January 2012, **draft-shirasaki-isp-shared-addr-07**

[21] Jason Weil, Victor Kuarsingh, and Chris Donley, "IANA Reserved IPv4 Prefix for IPv6 Transition," Internet Draft, work in progress, September 2010, **draft-weil-opsawg-provider-address-space-02**

[22] Victor Kuarsingh, Chris Donley, Jason Weil, Marla Azinger, and Christopher Liljenstolpe, "IANA-Reserved IPv4 Prefix for Shared Address Space," Internet Draft, work in progress, February 2012. (Became RFC 6598[1]), **draft-weil-shared-transition-space-request-15**

[23] ARIN Public Policy Mailing List (PPML), "Shared Transition Space for IPv4 Address Extension," ARIN-prop-127, **http://lists.arin.net/pipermail/arin-ppml/2011-January/019278.html**

[24] "Shared Transition Space for IPv4 Address Extension," ARIN policy 2011-5, **https://www.arin.net/policy/proposals/2011_5.html**

[25] Brian Carpenter, Fred Baker, and Michael Roberts, "Memorandum of Understanding Concerning the Technical Work of the Internet Assigned Numbers Authority," RFC 2860, June 2000.

[26] Internet Architecture Board, "Response to ARIN's request for guidance regarding Draft Policy ARIN-2011-5," **http://www.iab.org/documents/correspondence-reports-documents/2011-2/response-to-arins-request-for-guidance-regarding-draft-policy-arin-2011-5/**

[27] Stan Barber, Owen Delong, Chris Grundemann, Victor Kuarsingh, and Benson Schliesser, "ARIN Draft Policy 2011-5: Shared Transition Space," Internet Draft, work in progress, September 2011, **draft-bdgks-arin-shared-transition-space-03**

[28] Geoff Huston, "Anatomy: Inside Network Address Translators," *The Internet Protocol Journal,* Volume 7, No. 3, September 2004.

[29] Daniel Karrenberg, Gerard Ross, Paul Wilson, and Leslie Nobile, "Development of the Regional Internet Registry System," *The Internet Protocol Journal,* Volume 4, No. 4, December 2001.

[30] Geoff Huston, "NAT++: Address Sharing in IPv4," *The Internet Protocol Journal,* Volume 13, No. 2, June 2010.

[31] *The Internet Protocol Journal,* Volume 14, No. 1, March 2011. This issue of IPJ is entirely devoted to the topic of IPv4 address depletion and IPv6 transition.

[32] Michelle Cotton and Leo Vegoda, "Special Use IPv4 Addresses," May 2012, **draft-vegoda-cotton-rfc5735bis-02**

WESLEY GEORGE has been working in IP networking for approximately 13 years, across operations, engineering and capacity planning, architecture, and design in large wired and wireless networks. He has been heavily involved in IPv6 evangelism and deployment for a surprisingly long time. He has been an active participant in IETF for 5 years, including serving as former co-chair of the IPv6 Renumbering (6renum) working group and current co-chair of the sunset4 working group. He was active in ARIN's policy development process during the time that the policy discussed in this article was being addressed. He currently works for Time Warner Cable, but this article represents his views alone, and should not be mistaken for his current employer's official stance on anything. Wes can be reached via twitter (**@wesgeorge**) or via **wesley.george@twcable.com**

December in Dubai: Number Misuse, WCIT, and ITRs

by Geoff Huston, APNIC

In November 1988, telephone companies from 178 nations sent their respective government representatives to the *World Administrative Telegraph and Telephone Conference* (WATTC) in Melbourne, Australia. At the time the generally cosy relationships between governments and their monopoly telephone companies often made it extremely difficult to see the difference between the government's representatives and those of the telephone company. The group resolved to agree to the rather grandly titled *International Telecommunication Regulations* (ITRs)[1].

At this meeting the companies' national representatives agreed to a set of additional regulations that supplemented the binding regulations of the *International Telecommunication Convention*. The goals of these regulations were rather grand; they aspired to promote the "harmonious development and efficient operation of technical facilities, as well as the efficiency, usefulness and availability to the public of international telecommunication services." More practically, these ITRs defined the general principles for the provision and operation of international telephony services among signatories to the ITRs.

At that time the Internet was little more than a somewhat obscure experiment in advanced data communication protocols undertaken by a small number of researchers in North America and to a far smaller extent in Europe. However, since 1988 the Internet—and the world in which the Internet has flourished—has changed dramatically. If we view the rise of the Internet over the past 25 years as a product of an appropriately liberalized international regulatory regime as much as it was a product of the titanic shifts in computing and communications technologies that also occurred over this period, then we can make the case that the Internet of today is a product of these ITRs. And what a prodigious product it has been!

In Dubai, between the 3rd and 14th of December 2012, the nations of the world will convene at the 2012 *World Conference on International Telecommunications* (WCIT)[2], and they intend to use this conference to review these 25-year-old ITRs and consider some proposed changes to this regulatory framework that underlie international telecommunications.

At the moment the international meeting cycle is ramping up to consider what aspects of the ITRs should be altered, what should stay the same, and what should be dropped. After all, much has happened in the past 25 years, and an argument could be made that the ITRs should be amended to better reflect today's world.

But the world is not exactly aligned at the moment about what should and what should not be folded into a new set of international regulatory obligations.

Some countries appear to be advocating for some quite specific measures to be added to the ITR to address what for them are characterized as otherwise unresolvable operational problems. Others are advocating a more general approach to have the ITRs explicitly embrace the Internet and fold references to the Internet in every place where specific carriage and service delivery technologies are referenced in the ITRs. It is when these two approaches intersect that the situation gets interesting.

In order to illustrate some of the underlying tensions that exist in this activity, I would like to take a specific example of a proposed amendment to the ITRs and consider in in terms of the broader context of telephony and the Internet.

The proposal I want to examine here concerns the topic that has been called "number misuse." In telephony this term referred to an operating practice where a call to a dialled number is not routed to the destination subscriber who is located at that called number, but instead the call is re-routed to a different destination.

What we see in the "Number Misuse" proposal for a revision of the ITRs is an attempt to fold the concepts of "number misuse" and the Internet together, with a result that some countries want the ITRs to explicitly take on the concept of "IP Address and Routing Misuse" within the framework of national obligations through common regulatory action within the same scope as the telephony called number misuse. If successful, this effort would result in a regulatory obligation for governments to take necessary actions to investigate and prosecute such instances of so-called "number misuse." The intended scope of such enforcement of such obligations would encompass not only the telephone network but also the Internet. Surely we all desire a global public communications network that operates with integrity, and surely we would want to see countries take the necessary actions to ensure that it happens. So why is this idea not exactly the best idea to appear in the ITR negotiation process so far?

Let's look at the motivations behind number misuse in the world of telephone carriers and telephone services, and then look at how it could conceivably map in to the world of the Internet.

To understand the telephone world and where this problem of number misuse is coming from, it may be useful to understand a little of how money circulates in the phone world.

Telephony: Sender Pays

In many ways the telephone leaned heavily on the telegraph service for its service model, which, in turn, leaned on the postal service, establishing a provenance for the telephone service model that stretched back over some centuries to at least the 1680s and London's Penny Post, if not earlier.

The postal service model that gained ascendency over the preceding centuries was one in which the original sender of the letter paid for the entire service of letter delivery. If the postal service that received the letter in the first place needed to use the services of a different postal service to complete the delivery, neither the sender nor the intended recipient were aware of it. The postal services were meant to divide the money received from the sender to deliver the letter, and apportion it between themselves to compensate each service provider for undertaking its part in the delivery of the letter.

The telephone service, for the most part, operates in a very similar fashion. The caller pays for the entire cost of the call, and the called party pays nothing.

When both the caller and the called party are connected to the same carrier, the process is straightforward. The carrier charges the caller for the cost of the call and, presumably, some small (often not so small) margin for profit.

However, when we apply the same model to, say, international phone calls, the model is not so simple. The common desire on the part of the telephone operators was to preserve the same simple model: the caller pays. Now in this case the caller pays the presumably higher price of establishing a voice circuit from a carrier in one country in one part of the world to another carrier in another country in another part of the world. But now the caller's carrier should not keep all the revenue associated with the call. The other end, the *terminating carrier,* has also incurred costs in servicing this call. The arrangement that the telephone industry developed was the concept of "intercarrier call accounting financial settlements."

To explain this concept it may be useful to introduce the unit of a *call minute,* which is commonly used as a means of measuring a telephone call. What carriers establish between themselves on a bilateral basis is the intercarrier settlement cost per call minute of a telephone call that originates in one carrier and is terminated by the other carrier.

Now if both carriers can establish a value of a call-minute settlement rate where in both directions the call-minute termination costs roughly equate to the call-minute settlement rate, then in theory, at any rate, neither party is relatively advantaged over the other, irrespective of whether the callers are predominately located in one carrier or in the other carrier. In theory, such an arrangement should be financially neutral to both carriers.

However, although in theory practice and theory should align, in practice it rarely happens. What happened in the telephone case was that we saw some carriers set a call-minute call-termination settlement rate that was well above cost, while at the same time set its international call tariffs such that outbound calls were prohibitively expensive for local subscribers.

The result was that the local customers of these carriers found it cheaper to request that the other party call them—the desired outcome. The local carrier then generated income not by charging local subscribers but by revenue generated as an outcome of the call accounting settlement payments that were generated by the net imbalance of called versus calling call minutes.

Carriers all over the world played this game. For example, in France in the early 1990s it was some 5–10 times more expensive to call a U.S. number from France than it was to make a call between the same two numbers in the other direction. If you add in a further consideration, namely that in the 1980s many carriers were part of the public administration and were in effect government-operated national monopolies whose profits contributed to national revenue, then you get an outcome that is described in *Opinion No. 1* of the 1989 ITRs, under the heading "Special Telecommunication Arrangements," namely: "...considering further that, for many Members, revenues from international telecommunications are vital for their administrations."

Telephony Special Services and Number Misuse

It is often said that the only really major innovation in more than a century of the telephone service was the fax. Perhaps that is a little too unkind, but innovations in the delivered services industry were few and far between. However, there were many innovations that are important to this story, and the ones that are relevant here are *number redirect* and the so-called *premium* services.

The premium services attracted a higher call cost, and the carrier conventionally split the revenue from the service with the called service. These services traditionally included weather forecasts, sports results, new headlines (until the Internet became all but completely ubiquitous and decimated these services!), and so on. They also attracted the sex industry. However, in many countries such services were not permitted, so a conventional premium service was not an option for this industry.

As ever, we are naturally inventive, and some folks came up with a clever solution to use number redirect to redirect the call to this otherwise not-permitted premium service to another country. As part of this redirection, the premium service provider needed to reach an agreement with the new home carrier of the call-termination point to divide the international call accounting revenue provided by callers to this service between the carrier and the service provider. Not only did this arrangement effectively circumvent local regulations relating to locally provided premium services, it also leveraged off the international call accounting arrangements to the benefit of the premium service provider as well as the terminating carrier.

We may be inventive, but all too often we are greedy as well. The next step was to circumvent any arrangement with the destination carrier and redirect the call to an entirely different carrier.

One of the side effects of deregulation of the telephone industry in many countries was that in place of a single carrier that would receive all incoming international calls for a given country code there were numerous carriers that were ostensibly competing for the these incoming calls. Instead of routing calls based solely on the dialled country code, carriers now could route calls based on number blocks within the country code, and use different transit routes based on number-block rules. What if a premium service provider took a number block from a country code and specified that all incoming calls were to be routed by a third-party carrier? That all sounds innocent enough, but what if this third party did not actually route the calls through to the country in question, but instead terminated the calls and still charged the calling carrier the international call accounting settlement rate? No doubt the service provider has gotten a better deal, so the service provider is happy, and the carrier that terminates the call is receiving a portion of the call settlement rate, so the terminating carrier is happy. But happiness is not universal here. The carrier in the called country code is getting nothing from this arrangement, even though its country call code is being used for these premium service calls. From the carrier's perspective it is being defrauded of what it might claim is legitimate international call accounting revenue through the "misuse" of the number block drawn from its country code.

If the country-code carrier could discover this unauthorized number-block diversion, then presumably it could withdraw the number block and stop the international call diversion. Unfortunately it does not always work. The carrier can withdraw the number block, but at times—and under perhaps somewhat shady circumstances—the premium service provider, and potentially the transit carriers, might still be able to convince local carriers that the number-block diversion is still legitimate. Although the country-code carrier might see the problem, the carrier's ability to enforce carriers in other countries to respect its authority regarding the use of number blocks drawn from its country code is not always clear. At times the carrier is effectively powerless to enforce a remedy.

And the scheme can be further refined. Why even enter into any form of discussion with the international carrier for a number block? Why not pick one or more of the more obscure national country codes, generate some number blocks from these codes, and then get a cooperative transit carrier to enter a number-block diversion request into the local carrier? The number block is perhaps drawn from a country code that already makes extensive use of third-party transit arrangements, the local carrier may not question the request, and the carriers in the countries from which the number blocks have been drawn may not have the resources to even detect that this event has occurred.

At this point we have arrived at the situation that is motivating some of the proposals to augment the ITRs in this round of negotiation. The position of the nations that have been highlighting this problem as being an important problem in the world of international telephony is that the unauthorized use of phone numbers drawn from their E.164[3] telephone number block is, in their eyes, a case of "number misuse."

The reason why they want to identify this situation and write it into the ITRs at this time is that they would like to involve governments in the role of enforcers of conformance with the conventions of management of telephone country codes. It appears that they would like to obligate governments to adopt a policy, as a common convention, that calls made to a country's country code be directed such that the call request is sent to an authorized carrier located in the country, and to ensure that all authorized carriers essentially honor the integrity of the country codes of all other countries that use the E.164 country-code number plan.

It is also reasonable to ascribe the motivation for this measure as one that is intended to ameliorate the inexorable revenue leakage of the former rich money tap of international call accounting settlement payments. I am not sure that the various antics of the international premium service market are the true intended target of this measure. I suspect that the intended targets of this proposed regulatory measure are those carriers that have devised other methods to honor the intentions of their callers when they make an international phone call, and make the phone of the dialled number ring, yet at the same time bypass the traditional call accounting arrangements. Already *Voice over IP* (VoIP) trunking is commonplace, where the call is mapped into a VoIP call, and one way to bypass the conventional call accounting measures is to use a VoIP trunk to enter the dialled country, and then pass the call back into the *Public Switched Telephone Network* (PSTN) as a locally originated call, terminating it on the originally dialled number. The call is then subject to domestic intercarrier call-termination tariffs, which are generally far lower than their international counterparts.

The Internet and services such as Skype are exerting massive downward pressure on what carriers can charge for conventional phone services without encouraging all remaining customers to use Internet-based services. In an effort to retain some level of market share, it is now evidently more commonplace for carriers themselves to embrace IP-based approaches and bypass these imposed intercarrier international settlement charges. For many countries in the developing world, however, this shift represents a twofold financial blow. Not only are they seeing their foreign-sourced revenue stream disappear at the same rate as the call-termination minutes of conventional telephony vaporise, but they are also seeing this revenue stream being replaced by growing IP traffic volumes that represent a net cost to the national economy.

It should come as no surprise to see some countries attempt to advocate an international regulatory response that is intended to reverse this development, and restore the role of the international telephone network as a means of structural flow of monies from the business sector from the richer economies to the consolidated revenue stream of those poorer economies.

Internet Number Misuse

In and of itself, the previous discussion is by no means a novel discussion for the telephone world, and the tensions exposed by the continual erosion of the traditional telephone business through the onslaught of new technology is not at all surprising.

What is perhaps a bit surprising are the recent moves within the ITR preparatory activities that see numerous national delegations advocating pulling Internet addressing and routing into the same category of telephone-number regulation and also fold these factors into this matter of number misuse in a manner that would apply to both E.164 numbers and IP addresses.

Now some things do not readily translate from telephony to the Internet: there is no "National IP Address Plan" as a counterpart to the E.164 number plan, because the IP address plan is aligned to networks, as distinct from countries. However, you could take a broad view and find some form of mapping from the proposed recommendations regarding the use of E.164 networks to IP addresses. It would appear that the application of the proposals regarding number misuse would see a regulation to the effect that IP packets should be routed to the destination address specified in the packet, and not rerouted and terminated elsewhere. Surely this scenario describes part of the way the Internet works in any case. For the network to actually function, packets need to be passed to their addressed destination. Or so you would think.

And that is indeed what happens much of the time within the Internet. But by no means all of the time. As part of the normal course of operation of IP networks, many operators deploy equipment that intercepts packets and forms a synthetic response using the address of the intended destination. And many national administrations either operate—or mandate the operation of—equipment that inspects packets in transit and discards packets addressed to certain number blocks.

What is going on? Why do network operators regularly "misuse" IP addresses by deliberately intercepting packets and generating a synthetic response?

Packet Diversion

The most prevalent reason is the use of proxies, and, in particular, web proxies. These devices sit "on the wire" and intercept web fetches and cache the downloaded data.

When another user requests the same URL, the proxy uses the cached version of the content rather than forwarding the request on to the original site. This caching is by no means unusual: it is typical for web browsers to cache the most recently visited webpages and when the user returns to the page, the local cached copy is used rather than re-performing the download. For the browser and the network operator the rationale for this form of "address misuse" is the same: it is both a desire to improve performance for the end user and a desire to increase the efficiency of the network by reducing the data volumes being shifted across the transit links. So the outcomes are, on the whole, positive outcomes; users see improved performance and potentially lower costs for the service, using an interception technique that is generally transparent.

Is the deployment of a web proxy an instance of fraud?

Here is where another critical difference between the Internet and the telephone world comes into play. In the Internet the sender does not "pay all the way" to get a packet from its source to its intended destination. In general, every IP packet could be thought of as being partially funded by both the sender and the receiver.

The user who generated the packet pays for an *Internet Service Provider* (ISP) service, and the ISP may, in turn, purchase transit services from another ISP, and so on for sequenced transit services. However, at a peering exchange point, or within a provider network, the sender's money runs out. The packet is not unfunded, however, for at this point the receiver's services take over, and the packet transits a path that is funded by the receiver's ISP's transit services, and there to the receiver's ISP and there to the receiver.

If a packet is diverted to a proxy, then who wins and who loses? Can we make the case that a party in this situation is being cheated?

As long as the proxy is a faithful proxy, then the user wins, insofar as the user experiences improved performance and the benefits of a more efficient network while still seeing precisely the same content. And the content provider wins, insofar as the content is delivered to the user without the incremental cost of packet handling at the content site. And the network service providers win, in so far as the amount of network traffic is reduced while the revenue levels remain constant. In this case there is no end-to-end service payment on the part of the user that would trigger an intercarrier settlement payment, so it is difficult to make the case that this action necessarily damages any party involved in the network transaction.

Given the widespread deployment of these proxy caching devices across the entire Internet, the beneficial outcomes of improved performance and network efficiency, and the option for content providers to use techniques that in effect mark content as not cacheable, it is extremely challenging to sustain a case that the use of proxies is a case of address misuse.

So the use of traffic diversion and intercepting proxies in the Internet is not generally regarded as an example of intentional fraud or even an accepted case of address misuse. It is just what we do today in the Internet.

Packet Interception

What about the deliberate interception and discarding of packets in flight? Surely this case is one of "misuse" of IP addresses?

That is a very hard case to make when you consider that such actions are exactly how firewalls work, and almost every network uses firewalls in some manner or other. The action of a firewall is to intercept all packets, and discard those that match some predetermined set of rules relating to acceptable and unacceptable packets.

Many users run firewalls that deliberately block all incoming connection requests unless they match quite specific rules.

Many ISPs run firewalls that deliberately block access to ISPs' services from users who are not direct customers of the ISP.

Many countries have content regulations that block access to certain content, enforced either through government-operated facilities or through obligations imposed through the conditions associated with the carrier license within that country. The country I live in, Australia, imposes such constraints on its carriers for certain types of content, as does China through its much-reported national firewall facilities.

Users, service providers and carriers, and governments all use various forms of packet interception. Are we all guilty of number misuse? Should we support changes to the ITRs to obligate governments to stop this practice completely?

Aside from many other motivations for firewalls, security is a continuing concern in the Internet, and there is little doubt that although firewalls have not eradicated all forms of toxic traffic and associated abuse and attack, they are an important part of a larger story about securing the Internet. Irrespective of the various views that are expressed at a national level about censorship, intellectual property rights, and the position of common carriers and users, it seems counterintuitive to me that we would want to obligate governments to pull down our firewalls and filters as a necessary consequence of a revised set of ITRs.

Number "Misuse"

What this example illustrates is that the two networks—the traditional telephone network and the Internet—operate in very distinct and different ways. It not only encompasses differences between circuit and packet switching, but also reaches into the differences in the concepts of a network transaction, differences in the tariff structures, and, critically, differences in the way in which financial settlements are undertaken between service providers on the Internet.

Consider what could readily be acknowledged as an operating practice that defrauds operators in the world of telephony and negatively affects the services provided to telephone subscriber—that same practice in the Internet can result in positive outcomes used to enhance performance, reduce costs, and improve the operational efficiency of the service delivered to end users.

This case of attempting to regulate "number misuse" illustrates the fact that to take a stance of "one size fits all" when considering the topic of international regulation of telecommunications is a stance that has considerable risks of generating outcomes that are entirely inappropriate when translating a particular situation from telephony to the Internet.

WCIT and the ITRs—Where to Go from Here?

The international call accounting arrangements used by the telephone world, and the use of structurally embedded imbalances in call accounting settlement rates, are still major factors in the ITR discussions. This accounting imbalance is sanctioned in the resolutions of the 1988 World Administrative Telegraph and Telephone Conference, where *Resolution 3,* concerning the apportionment of revenue, provided for structural cross-subsidization of the developing world through asymmetric fixing of call accounting rates between the so-called developed and developing economies.

But in an increasing commercial world of telecommunications, where it is no longer a relatively exclusive collection of publicly funded monopolies that were an integral part of public utility service providers that in effect were an instrument of national governments, pushing the onus of an international developmental agenda onto an increasingly privatized commercial activity has been a less-than-comfortable fit. Private operators see this situation in a more dispassionate light as a business cost input, and seek to find ways to minimize this cost in order to improve the competitive positions of their businesses.

However, the changes in this industry over the past 25 years are so much larger than even this significant broad-scale shift in the onus of capital injection and operation from the public to the private sector. At the same time, we are seeing an even more fundamental shift in technology foundations, from circuits to packets with the introduction of the Internet into the picture. This shift has brought about profound shifts in the engineering of communications infrastructure and, as we have seen, it also has triggered profound shifts in the pricing of the consumer service, shifting from transactional pricing to a "connection rental" model where packet transit costs are bundled into the service. This bundling, in turn, has led to profound shifts in the manner in which money moves between the network operators themselves.

And perhaps of even greater and more lasting significance in this industry is the decoupling of carriage and content. We have now seen the rise of highly valuable content-centric enterprises that have business models that rely on a ubiquitous and abundant underlying communications infrastructure but are not financially beholden to the infrastructure operators. They have been able to forge direct relationships with consumers without having to deal with any form of mediation or brokerage imposed by carriage providers. The current values of these content enterprises dwarf the residual value of the carriage service sector, and the outlook for this sector is one of continuing shift in value away from carriage service providers and into the areas of content-based services.

Given the sheer scale of these changes in this industry over the past quarter century, it seems to me that the view that you can simply fold the Internet transparently into the current framework of the ITRs by the prolific insertion of "and the Internet" into the text of the regulations is simply not viable.

Packets are not circuits, and the mechanisms used to engineer packet networks are entirely different from those used with the circuit switches that supported traditional telephony services. This difference encompasses far more than engineering. The ways in which users pay for services differ, and this shift in the retail tariff structure of the Internet service implies a forced change in the way in which carriers interact to support a cohesive framework of network interconnection. The concept of a "call" really has no direct counterpart in the Internet. To extend this thought further into the area of "call accounting" and "caller pays" is again an extension that does not clearly map into the Internet. So when the existing ITRs refer to intercarrier call accounting financial settlements, there is no clear translation of such a concept into the Internet. When we extend this intercarrier interconnection framework into structural imbalances in call accounting settlement rates, and extend this framework further into the concepts of number misuse, all forms of connection between traditional telephony and the Internet are completely lost.

However, this conclusion should not imply that the ITRs are now an historic relic, completely overtaken by comprehensive shifts in both the technology and service models of today's global communications network. Irrespective of the fine level of detail in these 25-year-old documents, the ideals behind the ITRs are indeed worthy ideals, and they should not be discarded lightly.

Ultimately, what we are dealing with here is the role of individual nation states with respect to a public communications service for the entire world. In setting forth a framework for supporting an efficient, effective, and capable global communications system, the obligations stated in the current ITRs relating to the promotion of international telecommunications services, and the endeavours to make such services generally available to the public, all remain thoroughly worthwhile objectives.

The concept that widely respected technology standards are critical to worldwide technical interoperability of any telecommunications service is also an important aspect, and again the recognition of this factor in the ITRs is a worthwhile consideration.

But, as we both review the changes of the past quarter century and try to peer into what may emerge over the next quarter century, perhaps less is best in this area of regulatory measures.

Rather than seeking to explicitly add various regulations that attempt to address specific incidents of number misuse, and instead of making rather clumsy efforts to include the Internet into the already detailed provisions relating to intercarrier settlement models of the increasingly historic traditional telephone network, perhaps the best set of ITRs we could have for tomorrow's world are national obligations that support a lightweight common regulatory framework.

This framework should be both more minimal with respect to describing or relying on particular technologies and service frameworks and more encompassing in scope in stating the overall objectives and common aspirations all nations share in supporting this unique, incredibly valuable common resource of a common communications service that truly embraces the entire world.

Postscript: "It's All Just Telecoms"

I received a comment soon after I wrote an early draft article that I thought would provide some further insight to the WCIT process, so here is the comment and some further thoughts on the topic:

The comment was in the form of a report from a preparatory meeting for WCIT earlier in 2012. Evidently there is a mood within certain parts of the ITR drafting process to simply say: "The ITRs should apply to the Internet in full, because the Internet is nothing more than a telecom service and should be treated that way."

In one sense it is true that the Internet is nothing more than a telecommunications service, but in the same way that the post, radio, television, and of course the telephone are also all just telecommunications services. But the nature of the particular service has many consequences, and the attempt to lump telephony and the Internet into the same form of regulatory handling is at best a somewhat misguided effort.

I truly wonder if, more than a century ago, the counterparts of today's government delegates, in a meeting of that august body, the *Universal Postal Union* (UPU), would have argued that a telephone conversation was just an exchange of letters without the artifice of paper, and that the telephone was indeed just a part of the postal service, because it is just "a communications service."

Indeed I am pretty sure their counterparts did precisely that, and for the next 80 years or more in many countries the Postmaster General operated the telephone service, and operated the wireless spectrum administration and regulated radio and television broadcasts, as well as operating the national postal service, the telegraph service, and telex services, all because "it's all just communications."

But, ultimately we changed this paradigm. We created distinct entities to administer different communications media and services because it is actually not "all just communications"—nor is it "all just telecoms." Effective regulatory handling of these different communications mechanisms, using distinct forms of investment and finances, and at times entirely distinct regulatory frameworks and often distinct organizations and associated participatory arrangements, allows us to realize the true potential of these various services and do so efficiently and effectively. This recognition of a need for distinction in the regulatory frameworks for various services avoids the unfortunate situation of the stultifying dead hand of history misapplying one form of regulation on an entirely distinct and very different medium.

I suspect the best thing the postal folks, in the form of the UPU, ever did was to tell the telephone folks "hail and farewell" and let them get on with their role using an organization specifically designed to meet their collective needs in supporting telephony.

It may be well and truly time for the telephone folks, in the form of the *International Telecommunications Union* (ITU), to come to a similar arrangement in its dealings with the Internet!

Disclaimer

These views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

Further Reading:

[1] The current International Telecommunication Regulations (1988):
http://www.itu.int/dms_pub/itu-t/oth/3F/01/T3F010000010001PDFE.pdf

[2] World Conference on International Telecommunications (WCIT-12),
http://www.itu.int/en/wcit-12/Pages/default.aspx

[3] Geoff Huston, "ENUM—Mapping the E.164 Number Space into the DNS," *The Internet Protocol Journal,* Volume 5, No. 2, June 2002.

GEOFF HUSTON, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of numerous Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005. He served on the Board of Trustees of the Internet Society from 1992 until 2001.
E-mail: **gih@apnic.net**

Behind the Curtain: IP Fast Reroute

by Russ White, Verisign

The field of network and protocol engineering has three watchwords: *faster, bigger,* and *cheaper.* Although we all know the joke about choosing two out of the three, the reality of networking is that we have been doing all three for years—and it doesn't look like there is any time on the horizon when we will not be doing all three.

In that spirit, *IP Fast Reroute* addresses all three of these watchwords. Fast—you are probably thinking—is obvious, but what about bigger and cheaper? Fast Reroute provides the network designer with some trade-offs in the space of redundancy through additional backup links against deploying protocol changes, and network stretch against the size of a failure domain, so you can—in theory—build larger, less-redundant failure domains with Fast Reroute than without.

But to understand these effects, we need to go behind the curtain, understanding Fast Reroute as more than a few configuration options. This article first looks at the motivation behind IP Fast Reroute, and then discusses four different techniques, or stages, in the Fast Reroute story.

What Is Your Motivation?

To really discuss network speed, we need to be able to define how fast "fast" really is. In the 1980s, a network was fast if it could converge in 90 seconds or less (the longest time the *Routing Information Protocol* [RIP] could take to converge). As we moved into more advanced Distance-Vector and Link State protocols (*Enhanced Interior Gateway Routing Protocol* [EIGRP], *Open Shortest Path First* [OSPF], and *Intermediate System-to-Intermediate System* [IS-IS]), 5-second convergence became the norm. We learned to tweak timers to get to convergence times faster than 1 second.

But what if we need convergence that is faster than less than 1 second? What if we need to converge so fast that the only packets lost are either in flight or in a buffer waiting to be serialized onto the link? And what if we need to be able to handle a large number of prefixes with minimal network disruption due to link or device failures?

IP Fast Reroute techniques come into play in this situation.

Preinstalled Backup Paths

Although it is often sold as a Fast Reroute technique, *preinstalled backup paths* really are not; rather they support other Fast Reroute techniques at the protocol level. If the protocol has calculated a loop-free path that is an alternate to the current best path, this alternate path can be installed in the forwarding table so it is readily available for use in case the best path fails.

This solution does provide immediate failover at the hardware level, but the alternate path must be calculated to be installed. How is this alternate path computed?

Loop-Free Alternates

The first mechanism available for calculating an alternate path is with *Loop-Free Alternates*. To understand this mechanism, we must make a short detour into graph theory (or geometry, if you prefer). Use the following network as an example:

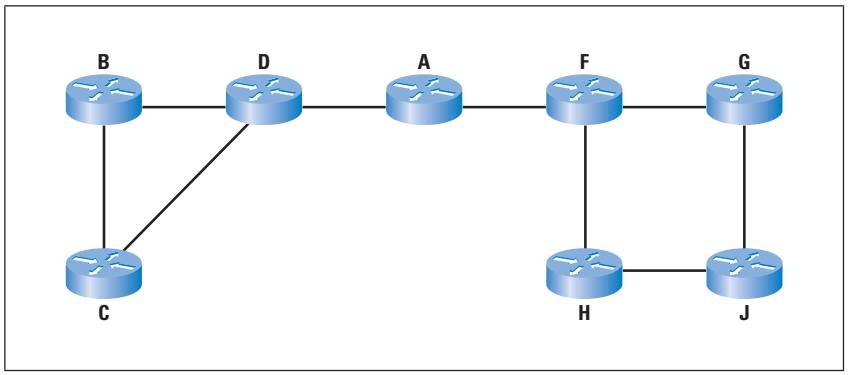


Figure 1: Network for Loop-Free Alternates

Assume:

- A is the destination.
- B's best path is through D to A.
- G's best path is through F to A.

What is the key to allowing B to forward traffic through C toward A if the B → D link fails? B must know the traffic it forwards to C (for A) will not be forwarded back to B itself. How can B know C will forward the traffic to D, rather than to B itself? By examining the metric at C toward A.

In EIGRP, B knows C's metric toward A because the routing protocol includes this information in the update. In a link state protocol (OSPF or IS-IS), B can calculate C's cost to A directly by running *Shortest Path First* from C's perspective (given B and C share the same link state database).

Loop-free alternates are simply calculating whether any given neighbor will forward traffic to any particular destination back to you, or on toward the destination. If a neighbor would forward the traffic on toward the destination, then it is a loop-free alternate.

Under what conditions would C forward traffic sent from B back to B? *If C is using B as its best path (or one of its best paths) toward A.*

What about G? If it forwards traffic to J toward A, will J return the traffic to G itself? In this four-hop ring, there are two possible configurations:

- J is using H as its best path. In this case, traffic forwarded by G to A through J will be correctly forwarded. Note, however, that in this case H cannot use J as an alternate path toward A, because any traffic H sends to A through A will loop back to H itself.
- J is using G as its best path. In this case, J can use G as a loop-free alternate, but G cannot use J as a loop-free alternate.

No matter how you work the metrics in the four-hop ring case, there will always be at least one device that does not have a loop-free alternate path to A.

Split Horizon and Loop-Free Alternates

If the concept of loop-free alternates is difficult to understand by considering the problem in this way, another useful way to look at the problem is through the distance-vector idea of *split horizon*. To review, the split horizon rule states:

> *Do not advertise a route to a destination toward a neighbor you are using to forward traffic to that same destination.*

If C is forwarding traffic toward A to B, then C will not advertise A to B, meaning B will not even know about this alternate path, preventing a loop even if B's best path to A fails. If you always consider where a distance-vector protocol will split horizon, you will always be able to see where loop-free alternates will fail to provide an alternate path to any given destination.

Getting Around the Loops

If we want to design a system that will find every possible alternate path toward a given destination, rather than just finding those that are not normally taken out by split horizon anyway, what must we do? We need to find a way to route through a neighbor to some distant next hop without that neighbor actually forwarding the traffic back to the originating router.

To put this concept in more concrete terms, examine the following network as an example:

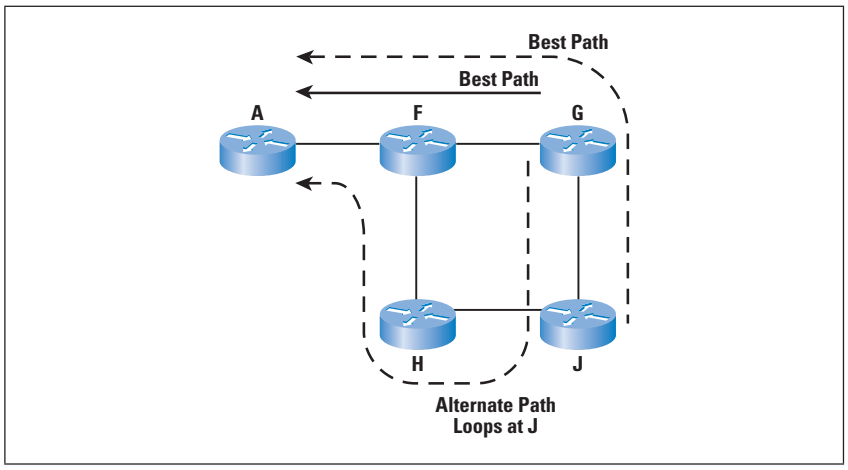


Figure 2: Alternate Path Loops

If G wants to use the path through J as an alternate path, then it must somehow figure out how to forward traffic to J without J returning the traffic to G itself. How can this process be done? G can tunnel the traffic through J to some device somewhere beyond J; therefore, every mechanism beyond loop-free alternates must use some form of tunneling to resolve the Fast Reroute problem. Calculating the point to which G needs to tunnel is the topic of the remaining mechanisms.

Not-Via

Even though we might be working with a link state protocol, it is easiest to understand *Not-via* in terms of a distance-vector protocol and split horizon. Not-via essentially begins with the observation that G does not have an alternate path to A through J in this case because J will not advertise such a route. *J is, in fact, using G as its best path toward A, so the path from G through J to A cannot be viable.*

The solution is not just simply having J advertise the route to A because traffic forwarded by G toward A through J will simply be looped back to G itself. So what is the solution?

In the case of Not-via, F advertises a route to itself through H only (not through G). This route will be advertised through H, then J, and finally to G. When G receives this route, it can determine that this path is an alternate path to A because its best path to A is normally through F. Any path that can reach F not through (or not via) its best path to F must, necessarily, be a loop-free alternate path to F. To reach A through F, however, G must tunnel to F directly, thereby avoiding the problem of J returning traffic destined to A back to G.

The address F advertises through H only is called "F Not-via G," and that is why this system is called "Not-via." This mechanism works in every topology (so long as an alternate path exists). The one downside to Not-via is that for each protected link or node, a new advertisement must be built and advertised through the network.

Disjoint Topologies

The problem of finding a next hop that passes over the split-horizon point can also be solved using the ability to form multiple disjoint topologies—multiple topologies that do not share the same links (or nodes, in some cases) to reach the same set of destinations. If this information sounds complex, that is because it is complex; a lot of hours and thought have gone into various systems to build and use multiple disjoint topologies within a single physical network. But there is a moderately simple way, referring back to Figure 2. In this network, G can take the following steps:

1. Remove the G → F link from its local database temporarily (just for this calculation).
2. Calculate the best path to F.
3. If an alternate path to F exists, mark this alternate path as a second topology.

4. If its path to F fails, place all traffic that would normally pass across G → F on this alternate topology.

It might not be obvious from this set of actions, but these actions will actually cause G to discover that it is, in fact, on a ring, and that it can place traffic on the opposite direction on this ring to get traffic to the same destination. Placing the traffic it would normally send to F via G → F on a separate topology overcomes the forwarding table at J, a process that would loop the traffic back to G itself. You could use a tunnel to F instead of a separate topology; tunnels are, in effect, a disjoint topology seen in a different way.

Conclusion

What advantage does IP Fast Reroute provide the network designer? The ability to reduce the amount of physical redundancy while maintaining the same actual level of redundancy in the network. Moving to Not-via or disjoint topology solutions removes the need to manually manage link costs as well, while adding only moderate complexity at the protocol level.

IP Fast Reroute is an interesting technology just on the edge of adoption that will be useful in campus, data center (through Layer 2 routing), and standard Layer 3 network designs.

For Further Reading

Work is currently active on the disjoint topology mechanism within the research community and the IETF; in particular, the following drafts will be of interest to anyone who wants to learn more:

[1] Alia Atlas, Robert Kebler, Maciek Konstantynowicz, Andras Csaszar, Russ White, and Mike Shand, "An Architecture for IP/LDP Fast-Reroute Using Maximally Redundant Trees," Internet Draft, work in progress, October 2011, **draft-atlas-rtgwg-mrt-frr-architecture-01**

[2] Alia Atlas, Gabor Envedi, and Andras Csaszar, "Algorithms for Computing Maximally Redundant Trees for IP/LDP Fast-Reroute," Internet Draft, work in progress, March 2012, **draft-enyedi-rtgwg-mrt-frr-algorithm-01**

[3] Stefano Previdi, Mike Shand, and Stewart Bryant, "IP Fast Reroute Using Not-via Addresses," Internet Draft, work in progress, December 2011, **draft-ietf-rtgwg-ipfrr-notvia-addresses-08**

[4] Clarence Filsfils and Pierre Francois, "LFA applicability in SP networks," Internet Draft, work in progress, January 2012, **draft-ietf-rtgwg-lfa-applicability-06**

RUSS WHITE is a Principle Research Engineer at Verisign. He has co-authored numerous technical books, RFCs, and software patents. He focuses primarily on network complexity, network design, the space where routing and naming intersect, control-plane security, protocol design, protocol operation, and software-defined networks. E-mail: **riwhite@verisign.com**

Letters to the Editor

Ed.: We received several letters in response to the article "A Retrospective: Twenty-Five Years Ago," by Geoff Huston, published in the previous issue of this journal. Here is some of the feedback:

Hi Geoff,

Just wanted to show my appreciation for your nice article. As an ex-DEC who moved to WorldCom after my MSc in Computer Engineering & Telecoms with a Master's project on IP signaling over ATM, I can certainly relate to a large part (not all ;–) of what you wrote.

I normally don't read such long articles, but had to make an exception as I kept interested until the end!

Thank you!

—*Pedro Paiva, Etoy, Switzerland*
pedro.paiva@a3.epfl.ch

Greetings Geoff,

I just wanted to let you know that I really enjoyed your recent article, "A Retrospective: Twenty-Five Years Ago," published in *The Internet Protocol Journal*. I lived through most of the history that you talked about as I came up through the telecom industry and then finished off my career at Cisco.

It certainly is interesting to reflect back on all the past controversy around network infrastructure design and how competing ideas and philosophies played out. (Talk about losers, remember *Switched Multi-megabit Data Service* (SMDS) driven by the *Regional Bell Operating Companies* (RBOCs)? While at Nortel, I remember once in a design review meeting that one of our BNR geeks put up a slide (overhead foil back then) that showed various network evolution scenarios. The last one was an "oh-by-the-way, there's this theory that the Internet could take over the world" (of network infrastructure). All the room snickered. Who's laughing now?

There was as much energy, maybe more, put into defending architectures based on market control as there was on technological elegance. Still, it is a fascinating and dynamic industry full of extremely smart people with clever ideas, and I enjoyed every minute of it.

I started at "the phone company" in the late 1960s and it has been quite a journey from relay-driven switches controlling tip and ring loops to the current *Multiprotocol Label Switching* (MPLS) backbone networks, terabit switching, and hitching rides on photons.

Thanks for your insight and for your well-written article.

Best regards,

—Marc Williams
willimarc@gmail.com

The author responds:

Hi Marc,

Thanks for your note and your recollections from some 25 years ago.

I recall SMDS as well. If I recall correctly, this was an invention coming out of a university in Western Australia. Elsewhere in the world it was marketed as a 34-Mbps product. In Australia it was marketed in 2-Mbps and 10-Mbps forms (evidently the telco thought that we primitive Aussies were not "ready" for any higher speed!). I was a customer of their 10-Mbps product, and experienced some disappointment when it became evident that 10 Mbps was a theoretical peak that was simply unachievable because the inline PCs that were used for packet accounting slowed the throughput of any SMDS link down to just 3 Mbps! So in Australia SMDS was largely killed by the telco and it was never really used for high-speed digital trunk services.

I experienced a similar reaction to the Internet in the late 1980s as you have observed, when, in response to suggesting that the universities were about to build a national IP network, many of the telco managers did the polite snicker performance and then suggested that we should "get with the times," sign up as customers of their national ATM network, and leave the engineering to them. I'm glad the universities saw through it and supported me in persisting along the path to a national IP network. It was a strange moment some 6 years later when the same telco came knocking on our door to make an offer to buy the network from the universities because their own efforts to construct an IP product were simply getting nowhere at the time.

It has indeed been quite a journey, and I too have enjoyed every bit of it!

Kind regards,

—Geoff, Chief Scientist, APNIC
gih@apnic.net

Hello Geoff!

I haven't chuckled that much in years; what great memories. A few of my strong memories:

- Lack of documentation for new functions in software required an off-net test network and a Sniffer. The amount of hours spent figuring exactly what the function was doing or wasn't doing could fill an ocean. Absolutely my favorite activity and still is.
- I inherited a stat-mux system that was transporting ASCII terminals back to a centralized DEC terminal server arrangement. Hated it with a passion. One day, after a couple of beers, a light bulb came on that Ethernet is a stat-mux, so I bought a couple of Cisco AGS units, remotely installed a terminal server and an AGS, hauled it back to the other AGS in the central location and danced a jig, and then I started ripping out the old WAN stat-mux the following week.
- Anything relying on a token for timing is pure evil. You never know when you've engineered a TTL exhaust until it happens, and that can be based on Distance + Nodes or pure application coincidence. Ring resets are the devil's work. Token-based systems are not stat-muxs, but Ethernets are; that's why Ethernet survived and is the "last man standing."
- I totally agree with your comments surrounding the "cloud." I can remember that the distributed-versus-centralized fad has occurred at least four times over the past 25 years ...
- Z80: I built my first PC with a Z80; thank goodness for the peek-and-poke function!
- OEM would claim anything was portable as long as it had a carrying handle attached, even if it took two people to carry it.
- I fell in love with TCP/IP very early for the simple reason that it has the best of both worlds: a tightly coupled connection and connectionless protocol. It is much faster to troubleshoot or modify because IP requires a different expertise than TCP, and when you run across individuals who can work across the layers, hire them!

So, a lot of fond memories. I started out as a telemetry engineer on the Apollo project and I thought that was challenging and fulfilling. But, it doesn't hold a candle to the 1984–1995 period.

Oh, one other thing; I take umbrage to "...the annoying persistence of FORTRAN." That's the first language I learned back in the late '60s and I still have an active compiler on an old laptop that I still program on ... LOL!!

Keep attacking the certificate situation! The current situation is a disgrace, and I fully support the concept presented by Barnes: let's hurry it up!

Regards,

—Paul Dover
pdover@centeriem.com

The author responds:

Hi Paul,

Thanks for those recollections. I too spent a massive amount of time starting as a protocol analyzer, trying to make an IBM PC look enough like a Uniscope to allow file transfer between the PC and the Univac mainframe—no doubt it was a character-forming experience, but all I can say now is thank goodness for *tcpdump* and *wireshark*!

Thanks for your note—I truly appreciate the feedback!

Warm regards,

—Geoff, Chief Scientist, APNIC
gih@apnic.net

Dear Ole,

Congratulations on your 25-year anniversary!

You can tell how well people enjoy their professions by how great their products are, and yours is in the "excellent" category.

Regards,

—Paul Dover
pdover@centeriem.com

Ole,

Congratulations on your reaching a major milestone: 25 years of technology publishing! We are glad that you are continuing this service through *The Internet Protocol Journal* and look forward to many more years in this field.

Best,

—T. Sridhar
tsridhar@ieee.org

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

September 2012 Volume 15, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: www.cisco.com/ipj

ISSN 1944-1134

From The Editor

Internet devices use various forms of timers and timestamps to determine everything from when a given e-mail message arrives to the number of seconds since a particular device was rebooted. Most systems use the *Network Time Protocol* (NTP) to obtain the current time from a large network of Internet time servers. NTP will be the subject of a future article in this journal. This time we will focus our attention on the *Leap Second,* which is occasionally applied to *Coordinated Universal Time* (UTC) in order to keep its time of day close to the *Mean Solar Time*. Geoff Huston explains the mechanism and describes what happened to some Internet systems on July 1, 2012, as a result of a leap second addition.

The Internet of Things (IoT) is a phrase used to describe networks where not only computers, smartphones, and tablets are Internet-aware, but also autonomous sensors, control systems, light switches, and thousands of other embedded devices. In our second article, David Lake, Ammar Rayes, and Monique Morrow give an overview of this emerging field which already has its own conferences and journals.

The *World Wide Web* became a reality in the early 1990s, thanks mostly to the efforts of Tim Berners Lee and Robert Cailliau. The web has been a wonderful breeding ground for new protocols and technologies associated with access to and presentation of all kinds of media. The phrase *Web 2.0,* coined in 1999, has, per Wikipedia, "...been used to describe web sites that use technology beyond the static pages of earlier web sites." David Strom argues that the term is no longer appropriate and that we have moved on to a new phase of the web, dominated by mobile devices and Social Networking.

The last few years have seen great advances in Internet-based collaboration tools. Sometimes referred to as *Telepresence,* these systems allow not only high-quality audio and videoconferencing, but also the use of shared whiteboards and other presentation material. In our final article, Pat Jensen describes one important component of such systems, namely the *Binary Floor Control Protocol* (BFCP), which the IETF's XCON Centralized Conferencing working group has developed.

As always we welcome your feedback on anything you read in this journal. Contact us by e-mail at ipj@cisco.com

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Leaping Seconds

by Geoff Huston, APNIC

The tabloid press is never lost for a good headline, but in July 2012 this one in particular caught my eye: "Global Chaos as Moment in Time Kills the Interwebs."[1] I am pretty sure that "global chaos" is somewhat "over the top," but a problem did happen on July 1 this year, and yes, it affected the Internet in various ways, as well as affecting many other enterprises that rely on IT systems. And yes, the problem had a lot to do with time and how we measure it. In this article I will examine the cause of this problem in a little more detail.

What Is a Second?

I would like to start with a rather innocent question: What exactly is a *second?* Obviously it is a unit of time, but what defines a second? Well, there are 60 seconds in a minute, 60 minutes in an hour, and 24 hours in a day. That information would infer that a "second" is 1/86,400 of a day, or 1/86,400 of the length of time it takes for the Earth to rotate about its own axis. Yes?

Almost, but this definition is still a little imprecise. What is the frame of reference that defines a unit of rotation of the Earth? As was established in the work a century ago in attempting to establish a frame of reference for the measurement of the speed of light, these frame-of-reference questions can be quite tricky!

What is the frame of reference to calibrate the Earth's rotation about its own axis? A set of distant stars? The Sun? These days we use the Sun, a choice that seems logical in the first instance. But cosmology is far from perfect, and far from being a stable measurement, the length of time it takes for the Earth to rotate once about its axis relative to the Sun varies month by month by up to some 30 seconds from its mean value. This variation in the Earth's rotational period is an outcome of both the Earth's elliptical orbit around the Sun and the Earth's axial tilt. These variations mean that by the time of the March equinox the *Solar Day* is some 18 seconds shorter than the mean, at the time of the June solstice it is some 13 seconds longer, at the September equinox it is some 21 seconds shorter, and in December it is some 29 seconds longer.

This variation in the rotational period of the Earth is unhelpful if you are looking for a stable way to measure time. To keep this unit of time at a constant value, then the definition of a second is based on an ideal version of the Earth's rotational period, and we have chosen to base the unit of measurement of time on *Mean Solar Time*. This mean solar time is the average time for the Earth to rotate about its own axis, relative to the Sun.

This value is relatively constant, because the variations in solar time work to cancel out each other in the course of a full year. So a second is defined as 1/86,400 of mean solar time, or in other words 1/86,400 of the average time it takes for the Earth to rotate on its axis. And how do we measure this mean solar time? Well, in our search for precision and accuracy the measurement of mean solar time is not, in fact, based on measurements of the sun, but instead is derived from baseline interferometry from numerous distant radio sources. However, the measurement still reflects the average duration of the Earth's rotation about its own axis relative to the Sun.

So now we have a second as a unit of the measurement of time, based on the Earth's rotation about its own axis, and this definition allows us not only to construct a uniform time system to measure intervals of time, but also to all agree on a uniform value of absolute time. From this analysis we can make calendars that are not only "stable," in that the calendar does not drift forward or backward in time from year to year, but also accurate in that we can agree on absolute time down to units of minute fractions of a second. Well, so one would have thought, but the imperfections of cosmology intrude once again.

The Earth has the Moon, and the Earth generates a tidal acceleration of the Moon, and, in turn the Moon decelerates the Earth's rotational speed. In addition to this long-term factor arising from the gravitational interaction between the Earth and the Moon, the Earth's rotational period is affected by climatic and geological events that occur on and within the Earth[2]. Thus it is possible for the Earth's rotation to both slow down and speed up at times. So the two requirements of a second—namely that it is a constant unit of time and it is defined as 1/86,400 of the mean time taken for the Earth to rotate on its axis—cannot be maintained. Either one or the other has to go.

In 1955 we went down the route of a standard definition of a second, which was defined by the *International Astronomical Union* as 1⁄31,556,925.9747 of the 1900.0 *Mean Tropical Year.* This definition was also adopted in 1956 by the *International Committee for Weights and Measures* and in 1960 by the *General Conference on Weights and Measures,* becoming a part of the *International System of Units* (SI). This definition addressed the problem of the drift in the value of the mean solar year by specifying a particular year as the baseline for the definition.

However, by the mid-1960s this definition was also found to be inadequate for precise time measure-ments, so in 1967 the SI second was again redefined, this time in experimental terms as a repeatable measurement. The new definition of a second was 9,192,631,770 periods of the radiation emitted by a Caesium-133 atom in the transition between the two hyperfine levels of its ground state.

Leaping Seconds

So we have the concept of a second as a fixed unit of time, but how does this relate to the astronomical measurement of time? For the past several centuries the length of the *Mean Solar Day* has been increasing by an average of some 1.7 milliseconds per century. Given that the solar day was fixed on the Mean Solar Day of the year 1900, by 1961 it was around a millisecond longer than 86,400 SI seconds. Therefore, absolute time standards that change the date after precisely 86,400 SI seconds, such as the *International Atomic Time* (TAI), get increasingly ahead of the time standards that are rigorously tied to the Mean Solar Day, such as *Greenwich Mean Time* (GMT).

When the *Coordinated Universal Time* (UTC) standard was instituted in 1961, based on atomic clocks, it was felt necessary that this time standard maintain agreement with the GMT time of day, which until then had been the reference for broadcast time services. Thus, from 1961 to 1971 the rate of broadcast time from the UTC atomic clock source had to be constantly slowed to remain synchronized with GMT. During that period, therefore, the "seconds" of broadcast services were actually slightly longer than the SI second and closer to the GMT seconds.

In 1972 the *Leap Second* system was introduced, so that the broadcast UTC seconds could be made exactly equal to the standard SI second, while still maintaining the UTC time of day and changes of UTC date synchronized with those of *UT1* (the solar time standard that superseded GMT). Reassuringly, a second is now a SI second in both the UTC and TAI standards, and the precise time when time transitions from one second to the next is synchronized in both of these reference frameworks. But this fixing of the two time standards to a common unit of exactly 1 second means that for the standard second to also track the time of day it is necessary to periodically add or remove entire standard seconds from the UTC time-of-day clock. Hence the use of so-called leap seconds. By 1972 the UTC clock was already 10 seconds behind TAI, which had been synchronized with UT1 in 1958 but had been counting true SI seconds since then. After 1972, both clocks have been ticking in SI seconds, so the difference between their readouts at any time is 10 seconds plus the total number of leap seconds that have been applied to UTC.

Since January 1, 1988, the role of coordinating the insertion of these leap-second corrections to the UTC time of day has been the responsibility of the *International Earth Rotation and Reference Systems Service* (IERS). IERS usually decides to apply a leap second whenever the difference between UTC and UT1 approaches 0.6 second in order to keep the absolute difference between UTC and the mean solar UT1 broadcast time from exceeding 0.9 second.

The UTC standard allows leap seconds to be applied at the end of any UTC month, but since 1972 all of these leap seconds have been inserted either at the end of June 30 or December 31, making the final minute of the month in UTC, either 1 second longer or 1 second shorter when the leap second is applied. IERS publishes announcements in its *Bulletin C* every 6 months as to whether leap seconds are to occur or not. Such announcements are typically published well in advance of each possible leap-second date—usually in early January for a June 30 scheduled leap second and in early July for a December 31 leap second. Greater levels of advance notice are not possible because of the degree of uncertainty in predicting the precise value of the cumulative effect of fluctuations of the deviation of the Earth's rotational period from the value of the Mean Solar Day. Or, in other words, the Earth is unpredictably wobbly!

Between 1972 and 2012 some 25 leap seconds have been added to UTC. On average this number implies that a leap second has been inserted about every 19 months. However, the spacing of these leap seconds is quite irregular: there were no leap seconds in the 7-year interval between January 1, 1999, and December 31, 2005, but there were 9 leap seconds in the 13 years between 1985 and 1997, as shown in Figure 1. Since December 31, 1998, there have been only 3 leap seconds, on December 31, 2005, December 31, 2008, and June 30, 2012, each of which has added 1 second to that final minute of the month, at the UTC time of day.

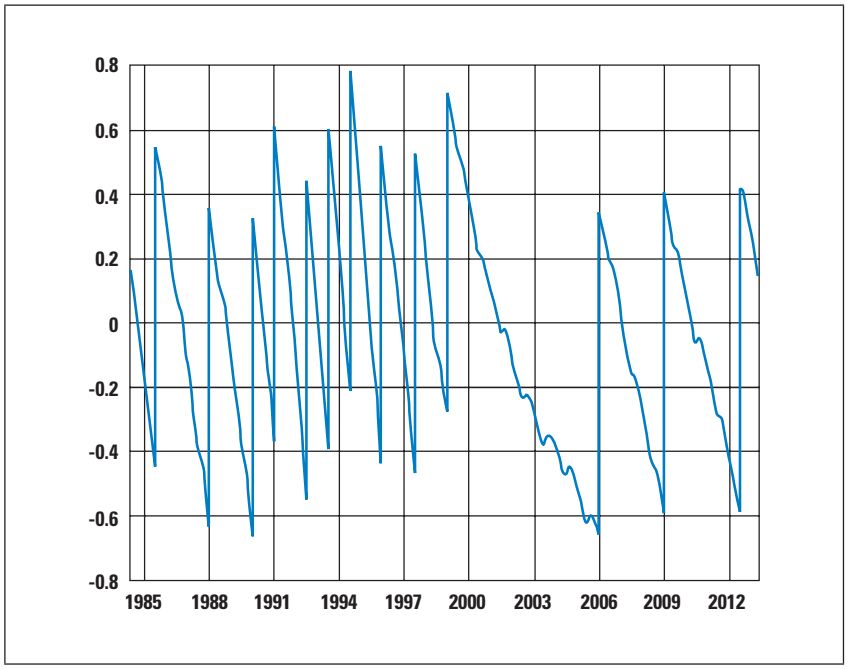


Figure 1: The difference between UT1 and UTC 1984–2012

Leaping Seconds and Computer Systems

The June 30, 2012 leap second did not pass without a hitch, as reported by the tabloid press. The side effect of this particular leap second appeared to include computer system outages and crashes—an outcome that was unexpected and surprising. This leap second managed to crash some servers used in the Amadeus airline management system, throwing the Qantas airline into a flurry of confusion on Sunday morning on July 1 in Australia. But not just the airlines were affected, because LinkedIn, Foursquare, Yelp, and Opera were among numerous online service operators that had their servers stumble in some fashion. This event managed to also affect some *Internet Service Providers* and data center operators. One Australian service provider has reported that a large number of its Ethernet switches seized up over a 2-hour period following the leap second.

It appears that one common element here was the use of the Linux operating system. But Linux is not exactly a new operating system, and the use of the *Leap Second Option* in the *Network Time Protocol* (NTP) [7–10] is not exactly novel either. Why didn't we see the same problems in early 2009, following the leap second that occurred on December 31, 2008?

Ah, but there *were* problems then, but perhaps they were blotted out in the post new year celebratory hangover! Some folks noticed something wrong with their servers on January 1, 2009. Problems with the leap second were recorded with Red Hat Linux following the December 2008 leap second, where kernel versions of the system prior to Version 2.6.9 could encounter a deadlock condition in the kernel while processing the leap second.[3]

> "[...] the leap second code is called from the timer interrupt handler, which holds *xtime_lock*. The leap second code does a *printk* to notify about the leap second. The *printk* code tries to wake up *klogd* (I assume to prioritize kernel messages), and (under some conditions), the scheduler attempts to get the current time, which tries to get *xtime_lock => deadlock*."[4]

The advice in January 2009 to sysadmins was to upgrade the systems to Version 2.6.9 or later, which contained a patch that avoided this kernel-level deadlock. This time it is a different problem, where the server CPU encountered a 100-percent usage level:

> "The problem is caused by a bug in the kernel code for high resolution timers *(hrtimers)*. Since they are configured using the CONFIG_HIGH_RES_TIMERS option and most systems manufactured in recent years include the *High Precision Event Timers* (HPET) supported by this code, these timers are active in the kernels in many recent distributions.

> "The kernel bug means that the *hrtimer* code fails to set the system time when the leap second is added. The result is that the *hrtimer* representation of the time taken from the kernel is a second ahead of the system time. If an application then calls a kernel function with a timeout of less than a second, the kernel assumes that the timeout has elapsed immediately after setting the timer, and so returns to the program code immediately. In the event of a timeout, many programs simply repeat the requested operation and immediately set a new timer. This results in an endless loop, leading to 100% CPU utilisation."[5]

Leap Smearing

Following a close monitoring of its systems in the earlier 2005 leap second, Google engineers were aware of problems in their operating system when processing this leap second. They had noticed that some clustered systems stopped accepting work during the leap second of December 31, 2005, and they wanted to ensure that this situation did not recur in 2008. Their approach was subtly different to that used by the Linux kernel maintainers.

Rather than attempt to hunt for bugs in the time management code streams in the system kernel, they noted that the intentional side effect of NTP was to continually perform slight time adjustments in the systems that are synchronizing their time according to the NTP signal. If the quantum of an entire second in a single time update was a problem to their systems, then what about an approach that allowed the 1-second time adjustment to be smeared across numerous minutes or even many hours? That way the leap second would be represented as a larger number of very small time adjustments that, in NTP terms, was nothing exceptional. The result of these changes was that NTP itself would start slowing down the time-of-day clock on these systems some time in advance of the leap second by very slight amounts, so that at the time of the applied leap second, at 23:59:59 UTC, the adjusted NTP time would have already been wound back to 23:59:58. The leap second, which would normally be recorded as 23:59:60 was now a "normal" time of 23:59:59, and whatever bugs that remained in the leap second time code of the system were not exercised.[6]

More Leaping?

The topic of leap seconds remains a contentious one. In 2005 the United States made a proposal to the *ITU Radiocommunication Sector* (ITU-R) Study Group 7's Working Party 7-A to eliminate leap seconds. It is not entirely clear whether these leap seconds would be replaced by a less frequent *Leap Hour,* or whether the entire concept of attempting to link UTC and the Mean Solar Day would be allowed to drift, and over time we would see UTC time shifting away from the UT1 concept of solar day time.

This proposal was most recently considered by the ITU-R in January 2012, and there was evidently no clear consensus on this topic. France, Italy, Japan, Mexico, and the United States were reported to be in favor of abandoning leap seconds, whereas Canada, China, Germany, and the United Kingdom were reportedly against these changes to UTC. At present a decision on this topic, or at the least a discussion on this topic, is scheduled for the 2015 *World Radio Conference*.

Although these computing problems with processing leap seconds are annoying and for some folks extremely frustrating and sometimes expensive, I am not sure this factor alone should affect the decision process about whether to drop leap seconds from the UTC time framework. With our increasing dependence on highly available systems, and the criticality of accurate time-of-day clocks as part of the basic mechanisms of system security and integrity, it would be good to think that we have managed to debug this processing of leap seconds.

It is often the case in systems maintenance that the more a bug is exercised the more likely it is that the bug will be isolated and corrected. However, with leap seconds, this task is a tough one because the occurrence of leap seconds is not easily predicted. The next time we have to leap a second in time, about the best we can do is hope that we are ready for it.

For Further Reading

The story of calendars, time, time of day, and time reference standards is a fascinating one. It includes ancient stellar observatories, the medieval quest to predict the date of Easter, the quest to construct an accurate clock that would allow the calculation of longitude, and the current constellations of time and location reference satellites. These days much of this material can be found on the Internet.

[0] Wikipedia, "Leap Second,"
http://en.wikipedia.org/wiki/Leap_second

[1] Herald Sun online,
http://www.heraldsun.com.au/news/leap-second-crashes-qantas-and-leaves-passengers-stranded/story-e6frf7jo-1226413961235

[2] "The deviation of the Mean Solar Day from the SI-based day, 1962–2010," graph in the Wikipedia article referenced earlier[0],
http://upload.wikimedia.org/wikipedia/commons/thumb/2/28/Deviation_of_day_length_from_SI_day_.svg/1000px-Deviation_of_day_length_from_SI_day_.svg.png

[3] Red Hat Bugzilla - Bug 479765, "Leap second message can hang the kernel,"
https://bugzilla.redhat.com/show_bug.cgi?id=479765

[4] "Re: Bug: Status/Summary of slashdot leap-second crash on new years 2008–2009,"
http://lkml.org/lkml/2009/1/2/373

[5] "Leap second bug in Linux wastes electricity," *The H Open,* July 3, 2012,
http://www.h-online.com/open/news/item/Leap-second-bug-in-Linux-wastes-electricity-1631462.html

[6] "Time, technology and leaping seconds," Google Official Blog, September 15, 2011,
http://googleblog.blogspot.de/2011/09/time-technology-and-leaping-seconds.html

[7] Burbank, J., Kasch, W., and D. Mills, "Network Time Protocol Version 4: Protocol and Algorithms Specification," RFC 5905, June 2010.

[8] Mills, D. and B. Haberman, "Network Time Protocol Version 4: Autokey Specification," RFC 5906, June 2010.

[9] Elliott, C., Haberman, B., and H. Gerstung, "Definitions of Managed Objects for Network Time Protocol Version 4 (NTPv4)," RFC 5907, June 2010.

[10] Lourdelet, B. and R. Gayraud, "Network Time Protocol (NTP) Server Option for DHCPv6," RFC 5908, June 2010.

Disclaimer

The views expressed are the author's and not those of APNIC, unless APNIC is specifically identified as the author of the communication. APNIC will not be legally responsible in contract, tort, or otherwise for any statement made in this publication.

GEOFF HUSTON, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of numerous Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005. He served on the Board of Trustees of the Internet Society from 1992 until 2001.
E-mail: **gih@apnic.net**

The Internet of Things

by David Lake, Ammar Rayes, and Monique Morrow, Cisco Systems

Until a point in time around 2008 or 2009, there were more human beings in the world than devices connected to the Internet. That is no longer the case.

In 2010, the global average of connected devices per person was 1.84. Taking only those people who use the Internet (around 2 billion in 2010), that figure becomes 6 devices per person.[1] Chip makers such as ARM have targeted developments of low-power CPUs and predicts up to 50 billion devices connected by 2020.[2]

Today, most of these devices are entities that the user interacts directly with—a PC or Mac, smartphone, tablet, etc. But what is changing is that other devices used every day to orchestrate and manage the world we live in are becoming connected entities in their own right.

They consist not just of users interacting with the end devices—the source and treatment of the information garnered will now occur autonomously, potentially linking to other networks of similarly interconnected entities.

Growing to an estimated 25 billion connected devices by 2015, the rapid explosion of devices on the Internet presents some new and interesting challenges.[3]

A Definition of the Internet of Things

The *Internet of Things* (IoT) consists of networks of sensors attached to objects and communications devices, providing data that can be analyzed and used to initiate automated actions. The attributes of this world of things may be characterized by low energy consumption, auto-configuration, embeddable objects, etc. The data also generates vital intelligence for planning, management, policy, and decision making. In essence, the five properties that characterize the Internet of Things are as follows:

- *A Unique Internet Address* by which each connected physical object and device will be identified, and therefore be able to communicate with one another.
- *A Unique Location—can be fixed or mobile—within a network or system* (for example, a smart electricity grid) that makes sense of the function and purpose of the object in its specified environment, generating intelligence to enable autonomous actions in line with that purpose.
- *An Increase in Machine-Generated and Machine-Processed Information* that will surpass human-processed information, potentially linking in with other systems to create what some have called "the nervous system of the planet."

- *Complex New Capabilities in Security, Analytics, and Management,* achievable through more powerful software and processing devices, that enable a network of connected devices and systems to cluster and interoperate transparently in a "network of networks."
- *Time and Location Achieve New Levels of Importance* in information processing as Internet-connected objects work to generate ambient intelligence; for example, on the *Heating, Ventilation, and Air Conditioning* (HVAC) efficiency of a building, or to study soil samples and climatic change in relation to crop growth.

The concepts and technologies that have led to the IoT, or the interconnectivity of real-world objects, have existed for some time. Many people have referred to *Machine-to-Machine* (M2M) communications and IoT interchangeably and think they are the same. In reality, M2M is only a subset; IoT is a more encompassing phenomenon because it also includes *Machine-to-Human* communication (M2H). *Radio Frequency Identification* (RFID), *Location-Based Services* (LBS), *Lab-on-a-Chip* (LOC) sensors, *Augmented Reality* (AR), robotics, and vehicle telematics are some of the technology innovations that employ both M2M and M2H communications within the IoT as it exists today. They were spun off from earlier military and industrial supply chain applications; their common feature is to combine embedded sensory objects with communication intelligence, running data over a mix of wired and wireless networks.

What has really helped IoT gain traction outside these specific application areas is the greater commoditization of IP as a standard communication protocol, and the advent of IPv6 to allow for a unique IP address for each connected device and object. Researchers and early adopters have been further encouraged by advancements in wireless technologies, including radio and satellite; miniaturization of devices and industrialization; and increasing bandwidth, computing, and storage power.

All these factors have played a part in pushing the boundaries toward generating more context from data capture, communication, and analytics through various devices, objects, and machines in order to better understand our natural and man-made worlds. In exploring the relationship between the IoT and *Information-Centric Networking* (ICN), embedded distributed intelligence will be an important attribute for ICN. Context that is distributed as opposed to centralized is a core architectural component of the IoT for three main reasons:

- *Data Collection:* Centralized data collection and smart object management do not provide the scalability required by the Internet. Managing several hundreds of millions of sensors and actuators in a *Smart Grid* network, for example, cannot be done using a centralized approach.

- *Network Resource Preservation:* Network bandwidth is scarce and some smart objects are not mains-powered, meaning that collecting environmental data from a central point in the network unavoidably leads to using a large amount of the network capacity.
- *Closed-Loop Functioning:* The IoT needs reduced reaction times. For instance, sending an alarm via multiple hops from a sensor to a centralized system, which runs analytics before sending an order to an actuator, would entail unacceptable delays.

Service Management Systems (SMS) (also known as Management Systems, Network Management Systems, or back-end systems) are the brain in the IoT. SMS interacts with intelligent databases that contain *Intellectual Capital* (IC) information, contract information, and manufacturing and historical data. SMS also supports image-recognition technologies to identify objects, people, buildings, places, logos, and anything else that has value to consumers and enterprises. Smartphones and tablets equipped with cameras have pushed this technology from mainly industrial applications to broad consumer and enterprise applications.

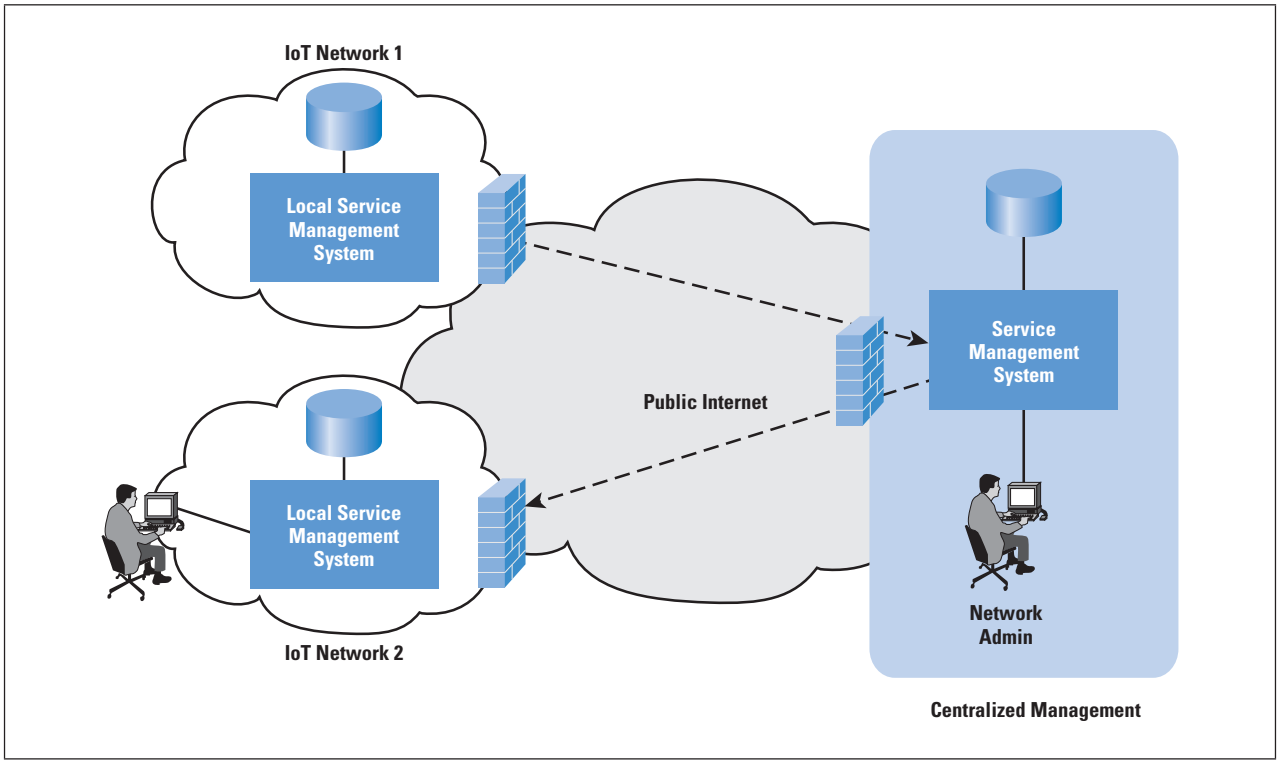
IC information includes intelligence of the vendor's (for example, Cisco) databases and systems such as contract DB, Manufacturing DB, and more importantly thousands of specific roles that are captured over the years by analyzing software bugs, technical support cases, etc.; that is, Cisco knows which devices were manufactured for which customers and with what features. Data collected by the collector is analyzed and correlated with the repository of proprietary Intellectual Capital, turning it into actionable intelligence to help network planners and administrators increase IT value, simplify IT infrastructure, reduce cost, and streamline processes.

Secure communications allow collected data to be sent securely from the agents or collection system to the SMS. SMS includes a database that stores the collected data and algorithms to correlate the collected data with Intellectual Capital information, turning the data into actionable intelligence that network planners and administrators can use with advanced analytics to determine the optimal solution for a problem (or potential problem) after the data is analyzed and corrected. More importantly, a secure mechanism allows the vendor to connect to the network remotely and take action. Secure communications also allows the SMS (automatically or via a network administer) to communicate back with the device to take action when needed.

However, centralized SMS for a large number of entities is very challenging given the near-real-time requirements and the effect on the network performance (see Figure 1). At the same time, centralized intelligence will be required for many IoT networks to interact with back-end centralized databases that are very difficult to distribute (for example, supplier Intellectual Capital databases).

This centralization is more demanding than the traditional multitier environments, servers, and back-end database types of applications where database caching was an effective approach to achieve high scalability and performance. Solution architects need to consider an optimal hybrid model that supports centralized and distributed systems at the same time. Distributed SMS may need to make sub-optimal decisions by using only narrow information to address real-time (or near-real-time) performance problems.

Figure 1: Typical Deployment of an IoT Network



Device and Data Security

The IoT will comprise many small devices, with varying operating systems, CPU types, memory, etc. Many of these devices will be inexpensive, single-function devices—for example, a temperature or pressure sensor—and could have rudimentary network connectivity. In addition, these devices could be in remote or inaccessible locations where human intervention or configuration is impossible.

The nature of sensors is such that they are embedded in what they are sensing—one can envisage a new workplace, hospital, or school construction project where the technology is introduced during the construction phase as part of the final fit rather than after completion as is common today. This paradigm in itself creates new challenges because the means of connectivity may exist only after the installation teams have left the site.

Additionally, methods must be taken to ensure that the authenticity of the data, the path from the sensor to the collector, and the connectivity authentication parameters cannot be compromised between the initial installation or configuration of the device and its eventual presence on the IoT infrastructure.

The challenges of designing and building IoT devices can be summarized as follows:

- IoT devices are typically small, inexpensive devices.
- They are designed to operate autonomously in the field.
- They may be installed prior to network availability.
- After deployment, these devices may require secure remote management.
- The computing platform may not support traditional security algorithms.

Because the IoT will not be a single-use, single-ownership "solution" with sources and the platform on which data may be consumed could be in different ownership, managerial, and connectivity domains, devices will be required to have equal and open access to numerous data consumers concurrently, while still retaining privacy and exclusivity of data where that is required between those consumers.

This requirement was neatly summarized by the IETF Security Area Directors as follows: "A house only needs one toaster even if it serves a family of four!"[4]

So we have seemingly competing, complex security requirements to be deployed on a platform with limited resources:

- Authenticate to multiple networks securely.
- Ensure that data is available to multiple endpoints.
- Manage the contention between that data access.
- Manage privacy concerns among multiple consumers.
- Provide strong authentication and data protection that cannot be compromised.

And we have to manage existing challenges that all network-attached devices have to contend with such as *Denial of Service* (DoS) attacks, transaction replays, compromised identity through subscriber theft, device theft, or compromised encryption.

These problems have particular relevance in the IoT, where the availability of data is of paramount importance. For example, a critical industrial process may rely on accurate and timely temperature measurement—if that sensor is undergoing a DoS attack, the process collection agent must understand that, and be able to either source data from another location or take evasive action.

It must also be able to distinguish between loss of data because of an ongoing DoS attack and loss of the device because of a catastrophic event in the plant. This ability could mean the difference between a safe shut-down and a major incident.

Authentication and authorization will require reengineering to be appropriate for the IoT. Today's strong encryption and authentication schemes are based on cryptographic suites such as *Advanced Encryption Standard* (AES), *Rivest-Shamir-Adelman* (RSA) for digital signatures and key transport, and *Diffie-Hellman* (DH) for key agreement. Although the protocols are robust, they make very high demands of the compute platform—resources that may not exist in all IoT-attached devices.

These authentication and authorization protocols also require a degree of user intervention in terms of configuration. However, many IoT devices will have limited access; initial configuration needs to be protected from tampering, stealing, and other forms of compromise between device build and install, and also for its usable life, which could be many years.

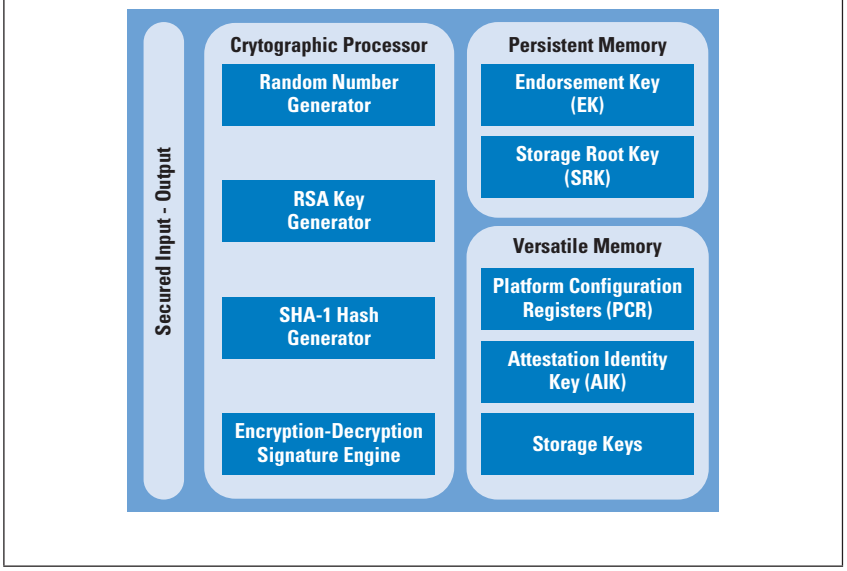
In order to overcome these difficulties, new authentication schemes that allow for strong authentication to many domains while building on the experience of today's strong encryption and authentication algorithms are required.

One possible approach could be to extend methodologies used in the PC industry such as the *Trusted Computing Group's Trusted Platform Model* (TPM).[5,6]

TPM-enabled devices are fitted at build time with a highly secure hardware device containing a variety of cryptographic elements. Keys and other factors known from this device by trusted third parties are then used in an attestation—a request to validate the authenticity of one device from known parameters.

Because the cryptographic keys are burned into the device during build and the signatures are known to a controlled, trusted third party, a high degree of confidence in the authenticity of the device being queried can be obtained. A typical TPM-compliant cryptographic chip is shown in Figure 2.

Figure 2: Trusted Platform Module



TPM has traditionally been limited by requiring access not only between the devices, but also to a trusted third party. In the IoT, where connectivity may be transient, this requirement is obviously a limitation. Extensions to the TPM to allow for high-confidence attestation between devices without involving a third party have been built; for example, *Direct Anonymous Attestation* (DAA).[7]

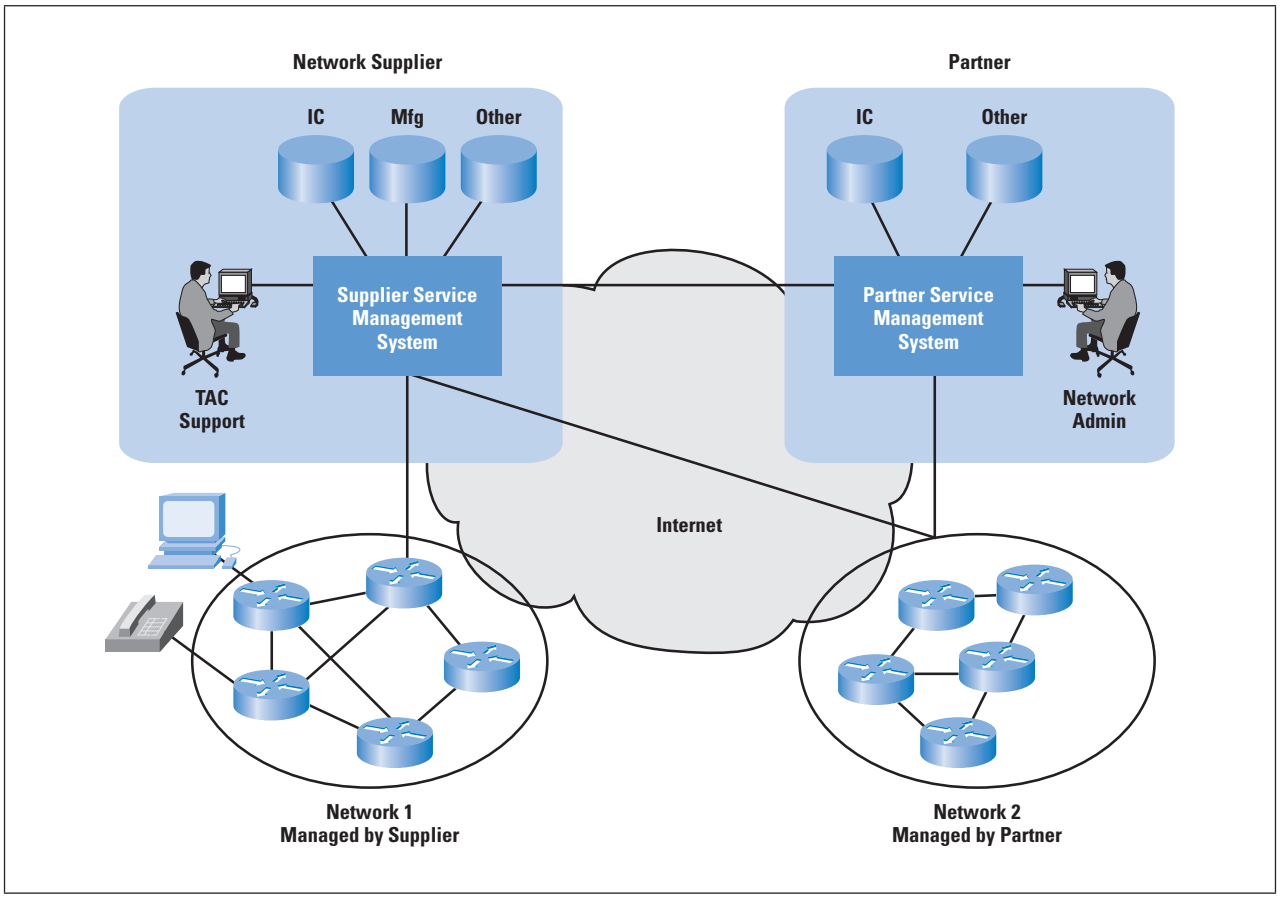
Other elements in security that could be considered include strong authentication between the device and the network attachment point (such as through electrical signatures at the *Media Access Control* [MAC] layer), application of geographic location and privacy levels to data, strengthening of other network-centric methods such as the *Domain Name System* (DNS) and the *Dynamic Host Configuration Protocol* (DHCP) to prevent attacks, and adoption of other protocols that are more tolerant to delay or transient connectivity (such as *Delay Tolerant Networks*).[8]

An IoT Case Study

The concepts behind the IoT allow management of assets within an enterprise with responsibility shared among customer, partner, and manufacturer in a manner that would previously have been difficult to control.

A typical IT network consists of routers, switches, IP phones, telepresence systems, network management systems such as call managers, data center managers, and many other entities (also known as "machines") with unique identification (for example, serial number, MAC address, or other address (for example, IP address). Such a solution is depicted in Figure 3.

Figure 3: Example of Smart Services



The system has the following components:

- The IT IP-based network: The network typically is owned by a business customer or an end customer (for example, a small business network). It includes IP devices that may be managed either by the supplier (via service contract), by a third party, Partner 2, or by the customer network administer.
- Smart agent or collection system (or sensor): An external collection system (for example, a server) or smart agent or collection systems on the managed devices gather the device and network information via numerous methods including *Simple Network Management Protocol* (SNMP) requests, *Command-Line Interface* (CLI) commands, syslog, etc. Collected information includes inventory, security data, performance data such as service-level agreement parameters, fault messages, etc.
- Supplier or partner back-end service management system: A service management system collects data from various devices and networks, correlates the collected data against intelligent Intellectual Capital rules and important databases (for example, Manufacturing database or Contact Management database), analyzes the results, and produces actionable and trending reports that examine the network and predict the performance.

- Two-way connectivity: Connectivity allows the front-end system (that is, smart agents and collection systems) to send data securely to the supplier or partner service management systems. It also allows the service management system to access the device or network secularly to take action when required.
- Secure entitlement and data-transfer capability to register and entitle customer networks and communicate securely (via encryption and security keys) with service providers or network vendors: Such capability is typically deployed on the collector and back-end systems.

A Smart Service provides a proactive intelligence-based solution addressing the installed-based lifecycle and *Fault, Configuration, Accounting, Performance, and Security* (FCAPS) management with the unique benefit of correlating data with the supplier's Intellectual Capital and recognized best practices. Using smart agents, Smart Services collects basic inventory information from the network in order to establish Install Base context.

Conclusions

The implications of the IoT on today's Internet are vast. With such a large number of devices and highly constrained network environments, provisioning and management of the IoT needs to be a part of the architecture. It is both unwise and impractical to provision each active device in the network manually throughout its lifecycle. Earlier technologies, including IP phones, wireless access points, or service provider *Customer Premises Equipment* (CPE), have demonstrated that provisioning can be carried out securely over the network.

The IoT encompasses heterogeneous types of devices that can be on public or private IP networks: from low-powered, low-cost sensors, to fully functioning multipurpose computers with commercial operating systems. For this reason, there can be no "one-size-fits-all" approach to IoT security. What is required is a series of architectural approaches that are dictated by specific IoT use cases. In certain industry solutions, most notably healthcare, security is not just important; information privacy is specifically mandated in many countries.

The challenges of designing, deploying, and supporting billions of IP-enabled endpoints, each producing data that needs to be analyzed and acted on, present exciting opportunities for the next generation of the Internet.

References

[1] Dave Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," April 2011, **http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf**

[2] "ARM targets Internet of Things with new low-power chip," Institute of Nanotechnology, **http://www.instituteofnanotechnology.co.uk/arm-targets-internet-of-things-with-new-low-power-chip/**

[3] **http://share.cisco.com/internet-of-things.html**

[4] Tim Polk and Sean Turner, "Security Challenges for The Internet of Things," IETF Security Area Directors, Feb. 14, 2011, **http://www.iab.org/wp-content/IAB-uploads/2011/03/Turner.pdf**

[5] Guillaume Piolle and Yves Demazeau, "Une architecture pour la protection étendue des données personelles," 2010, **http://guillaume.piolle.fr/doc/piolle10b.pdf**

[6] "Trusted Platform Module," ISO/IEC 11889, **http://www.iso.org/iso/catalogue_detail.htm?csnumber=50970**

[7] Jan Camenisch, "Direct Anonymous Attestation: Achieving Privacy in Remote Authentication," June 15, 2004. **http://www.zurich.ibm.com/security/daa/daa-slides-ZISC.pdf**

[8] Delay Tolerant Networking Research Group: **http://www.dtnrg.org/wiki**

DAVID LAKE, B.Sc., is a Consulting Engineer in the Research and Advanced Development Group at Cisco. He has more than 20 years of network design and deployment experience, ranging from X.25 and SNA, through the era of multiprotocol routing to IP, covering a wide range of networking technologies. He has extensive experience in transporting rich-media technologies across complex enterprise and service provider networks. David has worked as customer, network integrator, and manufacturer, and he understands the unique positioning of each of these areas in the IT industry. E-mail: **dlake@cisco.com**

AMMAR RAYES is a Distinguished Service Engineer at Cisco Systems focusing on Smart Service Technology Strategy. He has authored and co-authored over a hundred papers, patents and books on advances intelecommunications-related technologies. He is the President of the *International Society of Service Innovation Professionals* **www.issip.org**, an Associate Editor of ACM *Transactions on Internet Technology* and Editor-in-Chief of *Advances of Internet of Things Journal*. Dr. Rayes received his BS and MS Degrees in EE from the University of Illinois at Urbana and his Ph.D. degree in EE from Washington University in St. Louis, Missouri.
E-mail: **rayes@cisco.com**

MONIQUE MORROW is a Distinguished Engineer in the Research and Advanced Development Group at Cisco. She has over 20 years experience in IP internetworking that includes design, implementation of complex customer projects and service development for service providers. She has presented in various conferences on the topic of RFID, Grid Networking and Cloud Computing; and, she co-authored several books and publications. She is currently focused on the Internet of Things/Machine-to-Machine Communications in eHealth and security and is active in various SDOS and forums such the IETF, ITU-T and the FTTH Council Asia-Pacific, holding leadership positions in these organizations. Monique is a Senior Member of the IEEE the and a Life Member of the ACM. Monique has a Masters of Science Degree in Telecommunications Management and an MBA.
E-mail: **mmorrow@cisco.com**

The Demise of Web 2.0 and Why You Should Care

by David Strom

The term Web 2.0 has been around for about a decade[1], but we are finally seeing its disuse. No, the web itself is not going away, but the notion that an interactive layer of applications, protocols, programming languages, and tools has become subsumed into a new kind of web—one where everything is a *service,* mobile browsing is more important, and social networking has helped discover and promote new content. As a result, we do not really need the term anymore, because it is so much of what the web has become.

Think of this concept as going beyond the 2.0 label of the web: now we have a richer world of interactions that is just the beginning of how we use that tired old TCP port 80. All these developments mean that the readers of *The Internet Protocol Journal* are well poised to help others take advantage of this new complex web environment, because it has become the norm rather than some fancy address in the better part of town. Understanding its new structure and purpose is critical to building the next generation of websites and interactive applications.

Back in the early days of the web in the mid-1990s, it was largely static content that a browser would access from a web server. The notion of having dynamic pages that would automatically update from a database server was exciting and difficult to accomplish without a lot of programming help.

But then came Web 2.0, where the interactive web was born. We had blogging tools such as Google's *Blogger* and Automattic's *Wordpress,* and anyone could create a website that could be easily changed and instantly updated. Web and database servers became better connected, and new protocols were invented to better marry the two.

Everything as a Service

The past few years have seen the rise of *Software as a Service, Infrastructure as a Service,* and even *Platforms as a Service*.[2] The coming of *Cloud Computing* has meant that just about anything can be virtualized and moved into a far-away data center, where it can be managed and replicated easily, obviating the need for any physical infrastructure in the traditional enterprise data center.

Why is this change relevant for the modern web era? Four reasons:

- The web browser is still used as the main remote-access tool to configure and manage a wide variety of applications, network equipment, and servers, including all kinds of cloud-based infrastructures.

- Most of these "as-a-service" entities still run over ports 80 and 443 and piggyback on top of web protocols, for better or worse. We have gotten used to having these ports carry all sorts of traffic that has nothing to do with ordinary web browsing, and we have to do a better job of sorting out the ways apps use the traditional web ports too.
- We do not need to buy any software or install it on our own desktops; everything is available in the cloud at a moment's notice. What is more, we have gotten used to having the web as the go-to place to get new tools, software drivers, and programs. Software repositories such as *GitHub* and open source projects such as *Apache* have blossomed into places that corporate developers use daily for building their own apps. And why not? They have large support communities and hundreds of projects that are as well tended as something out of Oracle or Microsoft (and some would argue better, too).
- The days of a simple web server serving up pages is ever more complex, with typical commercial websites having ad servers, built-in analytics to track page views and visitors, discussion forums to moderate comments, connections to share the post on *Twitter* and *Facebook* (more on these in a moment), and videos embedded in various ways. All of these websites require coordinated applications and add-ons to the basic web server that require various cloud services. For example, the sites that I run for *ReadWriteWeb* use *Moveable Type* for our content, *Google Analytics*, *Disqus* discussions, interactive polls from `PollDaddy.com`, and custom-built advertising servers, just to name a few of the numerous add-ons. The ever increasing numbers of add-ons means maintaining this system is not easy, and it requires a lot of detailed adjustments on a too-frequent basis.

The Rise of Mobile Browsers

According to the research firm NetApplications[3], the share of web browsing originating from mobile devices has more than doubled in the past year. Although desktops still account for more than 90 percent of the data accessed from browsers, mobile devices are consuming the web at an increasing rate.

Part of this trend is that we are using more devices and they have become more capable. Android-based phones constitute the largest market share, and they have the fastest-growing consumer mobile phone adoption rate.[4] Certainly, more and more of us are browsing more webpages from mobile devices these days.

Another part of the trend of increased roaming on mobile devices is that more people are creating and using more mobile apps, too. Hundreds of new mobile apps with a wide variety of content are created every day. Professors at major universities teach computer science students how to code mobile apps, and you can even take online courses on *Java* programming.

But mobile browsing poses a conundrum for web designers. One school of thought is to build custom tablet applications for your website, to show off the features of the tablet interface and to make it easier for tablet users to interact with your content. The U.K. *Guardian,* for example, is leading the way in this area.[5]

Another school of thought is to improve the mobile experience, by either building a separate site that is optimized for smaller screens and lower bandwidth connections or allowing the site to work automatically under the constraints of the mobile browser itself.[6]

One real challenge for the mobile web browsing experience is the role of Adobe Flash and the newest of the *Hypertext Markup Language* (HTML) standards, HTMLv5. Apple decided when it released its first iPads to not support Flash, and since then there has been additional effort and movement to migrate many Flash-based sites, such as **YouTube.com**, toward HTMLv5, which is supported by Apple's tablets and can be more efficient for lower-bandwidth connections. Although this topic could easily be the subject of an entire article for this journal, our point in mentioning it here is that displaying video and similar content is still a problem for the web, even today.

Our mobile traffic at *ReadWriteWeb* has increased tremendously in the past year, and I suspect our site is typical of other sites. But this increase in traffic presents challenges for content creators: is it better to sell ad units around the content, even ads that have sub-par browsing experiences on mobile devices? Or code up your own iPad app (or use Verve's tools [**http://www.vervewireless.com/**] or something equivalent)? Certainly the level of engagement with the custom mobile app is greater, but it amazes me that sites with just static pages still are not optimized for mobile browsers yet, with large image downloads or multiple included links, for example.

Let's consider the site **Remodelista.com** as a case study of how to properly optimize a site for mobile browsing. The owners have implemented tricks to adjust its layout for different screen sizes. As you make your browsing window smaller (or as you run it on a mobile device with a small screen), the integrity of the site content remains intact, meaning that font sizes change and ad blocks appear on wider, higher-resolution screens and disappear on smaller ones, but the overall content stream remains the same, no matter what device is used to view it. This consistency is achieved by adding a lot of special coding to the webpages, as the following snippet shows:

```
<!--[if IEMobile 7]> <html class="no-js iem7 oldie" itemscope itemtype="http://schema org/"><![endif]-->
<!--[if lt IE 7]> <html lang="en" class="no-js ie6 oldie" xmlns="http://www.w3.org/1999/xhtml"
xmlns:nectar="http://saymedia.com/2011/swml" itemscope itemtype="http://schema.org/"><![endif]-->
<!--[if (IE 7)&!(IEMobile)]> <html lang="en" class="no-js ie7 oldie" xmlns="http://www.w3.org/1999/xhtml"
xmlns:nectar="http://saymedia.com/2011/swml" itemscope itemtype="http://schema.org/"><![endif]-->
<!--[if (IE 8)&!(IEMobile)]> <html lang="en" class="no-js ie8 oldie" xmlns="http://www.w3.org/1999/xhtml"
xmlns:nectar="http://saymedia.com/2011/swml" itemscope itemtype="http://schema.org/"><![endif]-->
<!--[if gt IE 8]> <html class="no-js" lang="en" itemscope itemtype="http://schema.org/"><![endif]-->
<!--[if (gte IE 9)|(gt IEMobile 7)]> <html class="no-js" lang="en" itemscope itemtype="http://schema.org/">
<![endif]-->
```

The Social Web Is Now Everywhere

It used to be the odd person in your professional circle who did not have or use an Internet e-mail account. Now the odd person is the one who does not have an account on Facebook or some other social networking site. What began in a Harvard dorm room in this decade has turned into a juggernaut of more than a billion users—and it is growing rapidly.

But the social web is more than a bunch of college kids swapping photos of their party pictures. A recent study from the University of Massachusetts at Dartmouth[7] shows that nearly 75 percent of the Inc. 500 (the fastest-growing 500 American private companies) are using *Facebook* or *LinkedIn,* a level that is about twice the percentage that are using corporate blogs. "Ninety percent of responding executives report that social media tools are important for brand awareness and company reputation. Eighty-eight percent see these tools as important for generating web traffic while 81% find them important for lead generation. Seventy-three percent say that social media tools are important for customer support programs." Clearly, these tools have become the accepted corporate intranet, the mainstream mechanism for communications among distributed work teams, and the way that many of us share events in our professional lives as well.

The social web means more than a "Like" button on a particular page of content; it is a way to curate and disseminate that content quickly and easily. It has replaced the Usenet *news groups* that many of us remember with a certain fondness for their arcane and complex structure. Or maybe that is just nostalgia talking.

In the presocial web past, even in the days when Web 2.0 was the rage, sharing and curation was not easy. If you wanted to share something you found online, more than likely you would e-mail your colleagues a URL. Now you can *Tweet,* post on *Facebook* and *Google+,* add an update to your *LinkedIn* account, put up a page on your corporate `Yammer.com` or `tibbr.com` server, or use one of dozens more services that will stream your likes and notable sites to the world at large. Or you likely have to do all of these tasks.

Back in the days of yore (say 2000), when I wrote a freelance article, it was sufficient to post a link to the story on my own personal website, in addition to perhaps sending an e-mail message or two to the people I thought might be interested in reading the content. Those days seem so quaint. Today, the process of writing the article is actually just the beginning, not the end. When the article appears online, a whole series of promotional activities must take place, including monitoring online discussions and adding my own comments, posting on the various social media sites, and re-Tweeting a link to my article several times over the next several days—all to ensure generation of lots of traffic.

There are even services such as **Ping.fm** and **Graspr.com** that can coordinate batch updates to numerous services, so that at the push of a button all of your social media will get your news at once. Or services such as **Nimble.com** that attempt to coordinate your entire social graph (as it is called) of friends and admirers so you can track what is going out across all your various networks.

Where We Go from Here

I have just tried to touch on a few topics to show that the days of the simple static web are "so over," as Generation Y says. Clearly, we have a long and rich future ahead of us for more interesting web applications.

References

[1] **http://en.wikipedia.org/wiki/Web_2.0**

[2] See "Alphabet Soup in the Cloud": **http://www.readwriteweb.com/cloud/2011/10/alphabet-soup-in-the-cloud-und.php**

[3] NetApplications research cited in this September 2011 article in *Computerworld:* **http://www.computerworld.com/s/article/9219696/Apple_rules_phone_tablet_browsing_market**

[4] Nielsen's statistics are typical: **http://blog.nielsen.com/nielsenwire/?p=29786**

[5] See **http://m.guardian.co.uk/**, but you really need to view it on an iPad or other tablet device to understand what they are trying to do with their content. See also: **http://www.readwriteweb.com/archives/the_guardian_ipad_edition_hits_ios_5_newsstands.php**

[6] See Thomas Husson's May 2011 Forrester Research blog post here: **http://blogs.forrester.com/thomas_husson/11-05-03-why_the_web_versus_application_debate_is_irrelevant**

Also see my own January 2012 article in ReadWriteWeb here: **http://www.readwriteweb.com/hack/2012/01/do-you-really-need-your-own-mo.php**

[7] See their January 2012 study here: **http://www.umassd.edu/cmr/studiesandresearch/2011inc500socialmediaupdate/**

DAVID STROM has created dozens of editorial-rich websites for publications such as *ReadWriteWeb, Tom'sHardware.com, eeTimes,* and others, as well as written thousands of articles for numerous IT magazines. He was the founding editor-in-chief of *Network Computing* magazine and author of two books on computer networking. He lives in St. Louis, Mo. and can be found at **strominator.com**, on Twitter **@dstrom**, and **david@strom.com** for those that still prefer e-mail.

Binary Floor Control Protocol

by Pat Jensen, Cisco Systems

Over the last decade, communication technologies have evolved to encompass new modalities of collaboration across IP networks—from instant messaging on a personal computer, to being able to make *Voice-over-IP* (VoIP) calls and also now including the growing adoption of *High-Definition* (HD) videoconferencing.

Operating systems, device types, and physical locations now are less affected as continued growth in networking has evolved to promote high bandwidth across wireless and wired networks. An example is the emergence of growing network-access technologies such as *Multiprotocol Label Switching* (MPLS), *Very-High-Speed Digital Subscriber Line 2* (VDSL2), *Long Term Evolution* (LTE), and *Data over Cable Service Interface Specification* (DOCSIS). With both availability of bandwidth and broadband user penetration increasing, the user's expectation of delivering immersive collaboration now becomes more apparent.

This evolution includes modern use cases accelerating the adoption of videoconferencing, such as enabling telemedicine for remote surgeries and diagnostic procedures as well as distance learning applications being used to connect educators with students across the globe.

This article introduces the *Binary Floor Control Protocol* (BFCP) as a standard for managing floor control during collaboration sessions across dedicated video endpoints, mobile devices, and personal computers running collaboration software. These capabilities can be delivered using an enabled *Session Initiation Protocol* (SIP) standards-based endpoint or as a software implementation in a collaboration application stack.

History

BFCP is a deliverable developed as part of the *Internet Engineering Task Force* (IETF) XCON Centralized Conferencing working group. The IETF XCON working group was formed to focus on delivering a standards-based approach to managing IP conferencing while promoting broad interoperability between software and equipment vendors.[1]

This mandate includes defining the objects, mechanisms, and provisions to assist in scheduling conferencing resources. These resources could be consumed as a conference enabled in a web browser, via an audio conference call or during a videoconference.

As defined, privacy, security, and authorization are considered integral in protecting the ability to join, participate in, and manage each conference session. The IETF XCON working group's initial focus was on unicast media conferences.

The IETF XCON working group was proposed in August 2003, with work starting early in October of that year.[2] Early requirements for BFCP were defined in RFC 4376, which describes important concepts, including a model for floor control and how it should be integrated in a conferencing platform.[3] Other important aspects such as security, including using authentication and encryption to provide protection against man-in-the-middle attacks, were also outlined.

In November 2006, Gonzalo Camarillo, Joerg Ott, and Keith Drage authored RFC 4582, which defined the Binary Floor Control Protocol.[4]

Besides BFCP, other standardization efforts around conference role and content management also were defined, including the ITU-T H.239 recommendation.[5] Unlike BFCP, H.239 applies specifically to H.323-enabled *Integrated Services Digital Network* (ISDN) and IP conferencing endpoints, whereas BFCP is designed to be agnostic of the underlying signaling protocol.

Protocol Details

The basic concept of floor control is analogous to managing a live in-person presentation, where you want to control who is presenting, manage and transition your presenters, and maintain a feedback loop. Also important is the ability to allow a presenter to show slides and share with your audience a white board or transparency projector.

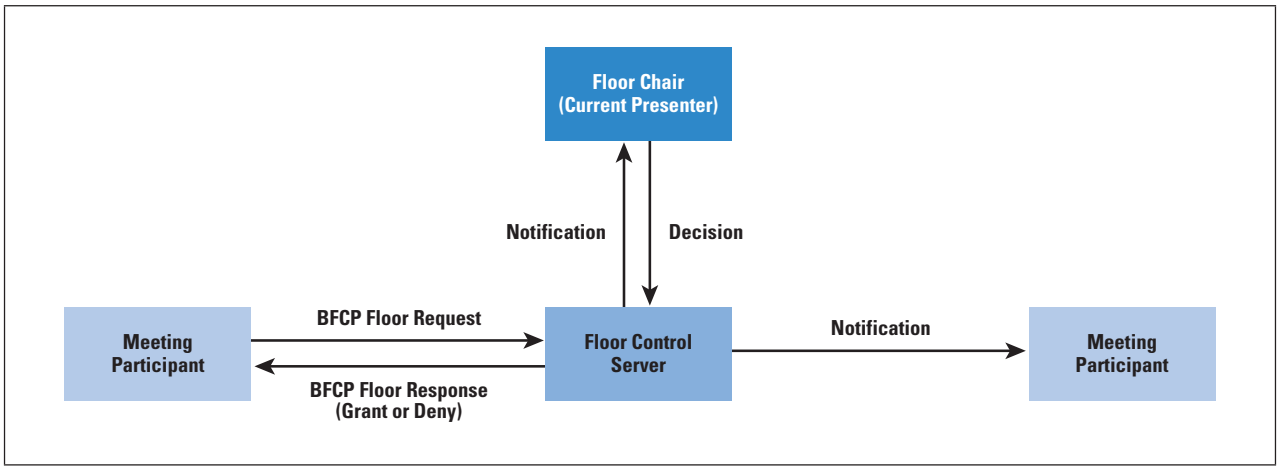
During an active collaboration session, a presenter may choose to present material to a remote user, or optionally to an audience on a call with multiple endpoints through a *Multipoint Control Unit* (MCU). This session could include many additional sources; for example, using a secondary video camera to show zoomed-in content (that is, an optical examination camera used in telemedicine) or any external video source.

This floor-control mechanism can also encompass functions available in a collaboration application stack, such as the ability to share the content of the presenter's desktop, application, or web browser.

BFCP provides the ability to manage multiple streams being presented during a collaboration session using floor control. BFCP accomplishes this management using a token-based mechanism where a single presenter can request control of the floor from the floor-control server.

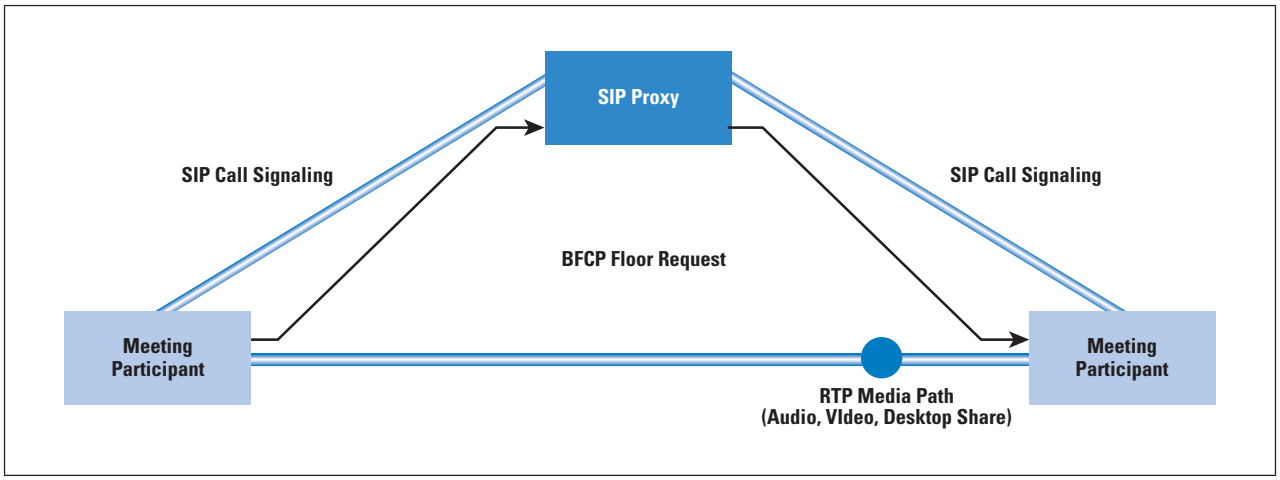
When this request is granted, the presenter holds the token and has the ability to open an additional stream to provide presentation data. Figure 1 examines this process in detail, with a meeting attendee requesting the token from the floor-control server to become an active presenter during the session.

Figure 1: BFCP Floor Request from Floor-Control Server



This same interaction can also take place during a point-to-point audio or video call with only two parties. In this case, a token can be used to signify which party will be presenting an additional stream, such as a secondary camera or application providing a desktop sharing session. Figure 2 shows an overview of this process. One of the critical differences here is that in a point-to-point call, the floor-control server capability is being provided by the user's device or application instead of using a multipoint control unit or conference server.

Figure 2: BFCP Floor Request in a Point-to-Point SIP Call



For instance, as a presenter, you can choose to present auxiliary streams via your application or endpoint and determine whether it is your primary, secondary, or tertiary stream. As a conference participant, you can also choose which stream you are currently viewing, also including the definition and quality of the secondary stream. In this case, current network conditions such as bandwidth and latency will also dictate the quality of additional streams.

BFCP is designed to be signaling protocol–agnostic, in that it is relying on the capabilities of the underlying signaling and transport protocols to set up each stream that is being managed, including whether voice, video, or content is being provided in the *Real-Time Transport Protocol* (RTP) stream.

For example, using a standards-based endpoint and *Session Initiation Protocol* (SIP), a SIP INVITE message is sent with the media capabilities line specifying the session description information about the stream. This data provides relevant information about the underlying video codec being used and the bit rate that is required to support the video and presentation streams.

In this case as multiple RTP media streams are transported across the network carrying audio and video traffic, *Call Admission Control* (CAC) and *Quality of Service* (QoS) tagging can be applied and enforced by the call-control platform, providing the ability to limit bandwidth usage and helping ensure that bandwidth is available on the network after the additional media stream is added.

Also important to note, BFCP can use *Transport Layer Security* (TLS) to provide encryption of floor information pertaining to each resource that is being controlled as well as the participants using and viewing them. BFCP provides the ability to support anonymous users as well for sessions where you may have a large audience or where anonymity is desired. An example of where this feature could be used is hosting a large web conferencing event where you have external attendees who may be outside of your organization.

One use case for BFCP includes the ability to focus on the presenter while the presenter is sharing a desktop application. With the ability to control the presenter's media stream, this feature adds additional immersion in a collaboration session, allowing you to both identify the presenter's visual cues and posture as well as focus on relevant content the presenter supplies.

Summary

The Binary Floor Control Protocol plays a very important role in helping manage diverse types of content being shared across multiple parties in a conference session. Today's modern implementations of BFCP span web conferencing applications as well as video and audio conferencing solutions across a wide array of vendors.

While these vendors are focused on delivering these capabilities across screen-led PC-centric types of devices, because of its inherent transport-agnostic capabilities, it is likely we will see BFCP being used to enable new modalities of content sharing across collaboration applications in the future.

Industry efforts are focusing on promoting collaboration applications across new arrays of devices, including using touchscreen technology on handheld computers and stationary LCD televisions to manipulate and visualize data in new ways.

Concepts such as manipulating session content using cognitive mapping as an evolution of electronic whiteboarding and transitioning an active conference from a tablet device to another type of room-based video-enabled endpoint during a collaboration session are two powerful examples of ways BFCP could be used in the future. On the horizon, touchscreen-enabled tablet and smartphone devices and HTML5-enabled web browsers also provide yet another avenue to enable rich standards-based multimedia conferencing with advanced content management.

Disclaimer

The views of this article do not necessarily represent the views or positions of Cisco Systems.

For Further Reading

[1] `http://datatracker.ietf.org/wg/xcon/charter/`

[2] `http://datatracker.ietf.org/wg/xcon/history/`

[3] Petri Koskelainen, Joerg Ott, Henning Schulzrinne, and Xiaotao Wu, "Requirements for Floor Control Protocols," RFC 4376, February 2006.

[4] Gonzalo Camarillo, Joerg Ott, and Keith Drage, "The Binary Floor Control Protocol," RFC 4582, November 2006.

[5] "Role management and additional media channels for H.300-series terminals," International Telecommunication Union Standard H.239, September 2005.

PAT JENSEN is a member of the Unified Communications consulting systems engineering team at Cisco Systems. Since 2010, he has designed collaboration architectures for Cisco's customers across the western United States. Prior to joining Cisco Systems, he served as an IP telephony design engineer designing and implementing unified communications and telepresence technologies at AT&T and SBC DataComm. E-mail, XMPP, SIP: `patjense@cisco.com`

Fragments

© *Stonehouse Photography/Internet Society*

Pierre Ouedraogo Receives 2012 Jonathan B. Postel Service Award

The Internet Society recently announced that its prestigious *Jonathan B. Postel Service Award* was presented to Pierre Ouedraogo for his exceptional contributions to the growth and vitality of the Internet in Africa. The international award committee, comprised of former Jonathan B. Postel award winners, noted that Mr. Ouedraogo played a significant role in the growth of the Internet in Africa and demonstrated an extraordinary commitment to training young engineers and participating in regional Internet organizations.

Mr. Ouedraogo is the Director of Digital Francophonie at *Organisation Internationale de la Francophonie* (OIF) based in Paris, France. Over the years, he has established networks of IT experts to coordinate African efforts to develop IT and use it as a tool for development. Mr. Ouedraogo initiated many IT technical workshops in Africa and is a founding member of numerous African regional organizations, including AfriNIC (the African Internet Registry for IP addresses); AfTLD *(African Internet Top Level Domain Names Association);* AFNOG *(African Network Operators Group);* AfCERT *(African CERT network),* and AfrICANN *(African network of participants to the ICANN process).*

"Pierre Ouedraogo is a highly-regarded technical leader in Africa, and he has been instrumental in bringing the Internet to Burkina Faso as well as other French-speaking African countries," said Lynn St. Amour, President and Chief Executive Officer of the Internet Society."

"His commitment to the expansion of the Internet and encouragement of young engineers to help them build their skills through training workshops has had a profound impact on the growth of the Internet across Africa."

The Postel Award was established by the Internet Society to honour individuals or organisations that, like Jon Postel, have made outstanding contributions in service to the data communications community. The committee places particular emphasis on candidates who have supported and enabled others in addition to their own specific actions. The award is focused on sustained and substantial technical contributions, service to the community, and leadership.

For more information about the Internet Society and the Postel award, see: **http://www.internetsociety.org/**

Atsushi Seike (L) with Vint Cerf and Jun Murai.

Vint Cerf Awarded Honorary Doctorate by Keio University

Keio University in Tokyo recently awarded Dr. Vinton Gray Cerf an honorary doctorate in Media and Governance for his work in the creation and governance of our modern Internet over the last forty years. On the recommendation of Professor Jun Murai, dean of the Faculty of Environment and Information Studies, Keio University president Atsushi Seike presented Dr. Cerf with the degree. The ceremony was held in the Enzetsu-kan, the historic public speaking hall on Keio's Mita Campus in Tokyo, and streamed live via the Internet to viewers around the world.

Professor Murai's recommendation for the degree, read during the ceremony, said that not only is Dr. Cerf the founding father of internetworking technology, "he is the global leader in many ways of the largest innovation for the 21st century, the Internet itself, which has become the core of today's information-based society." In addition to his work on TCP/IP with Robert Khan, Dr. Cerf's work in establishing the Internet Society and his stewardship of ICANN as its chairman were highlighted. Also mentioned was his role in *Delay/Disruption-Tolerant Networking* (DTN) and the first experiments connecting a space probe twenty million miles away using Internet protocols.

In his remarks, President Seike mentioned Dr. Cerf's forty-year commitment to advancing the role of networks in creating our global society, from the earliest days of the ARPANET through today's Internet. "[Dr. Cerf] understood quickly and clearly the international nature of the Internet and its potential for having a positive impact on the lives of not just the technical elite, but for all of the people of the world, as a tool for education, commerce, and the advance of democracy," he noted. Professor Seike compared Dr. Cerf's role in using technology to make the world a better place to the efforts of Yukichi Fukuzawa, the founder of Keio University, who in the mid-19th century was instrumental in bringing knowledge to Japan from the outside world, not as an academic exercise but in order to improve society.

Following the ceremony, Dr. Cerf gave an invited technical talk titled "Re-Inventing the Internet." He discussed the potential of DTN and *Mobile Ad Hoc Networks* as tools for disaster recovery. He presented his view of urgent technical problems, including the need for strong authentication and digital forensics. He also outlined society's need for preserving data, the programs that create and manipulate that data, and even the systems that are used to run those programs. Without such an effort, we will fail to preserve our own technical and cultural history for the thousands of years we have come to expect, he noted.

Dr. Cerf left behind the inscription, "I cannot imagine a greater honor than to be brought into this august and highly regarded university where contrary thinking is rewarded! I am most grateful to my good friend, Jun Murai, for his decades long commitment to the Internet."

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

December 2012 Volume 15, Number 4

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

Accurate timekeeping has long been an engineering challenge if not obsession in some circles. Take for example the iconic Swiss *chronometer* watch or the pendulum-controlled clock mechanism in London's Palace of Westminster, often referred to as "Big Ben." Such mechanical systems—accurate as they may be—are no match for the clocks we use in telecommunication and computer networks. In our last issue, Geoff Huston described the glitches encountered last June when a *Leap Second* was applied to *Coordinated Universal Time* (UTC). In this issue he explains the operation of the *Network Time Protocol* (NTP). The article is another installment in our series "Protocol Basics."

It is difficult to believe that it has been more than 25 years since the first publication of Douglas Comer's book series *Internetworking With TCP/IP*. Volume 1 of this series will soon be available in its sixth edition, and we asked the author to write an article about *Packet Classification* based on material in the book.

The recent *World Conference on International Telecommunications* (WCIT) did not have the outcome with respect to the Internet that many had hoped for. We plan to publish an analysis of this event in our next issue. This time—in our "Fragments" section—we have some reactions from the *Number Resource Organization* (NRO) and the Internet Society, as well as pointers to further information about WCIT.

January 1, 2013, marked the 30th anniversary of the *Transmission Control Protocol/Internet Protocol* (TCP/IP). A transition from the earlier *Network Control Program* (NCP) took place on January 1, 1983, also known as "Flag Day." Such an instant technology change would have been desirable for the transition from IPv4 to IPv6, but sadly this isn't possible. Instead we are happy to honor those who dedicate their careers to IPv6 deployment with an *Itojun Service Award*. See page 25 for more details.

On page 30 you will find some frequently asked questions about subscriptions to this journal. If you have other questions or comments, please contact us at ipj@cisco.com

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Protocol Basics: The Network Time Protocol

by Geoff Huston, APNIC

Back at the end of June 2012[0] there was a brief IT hiccup as the world adjusted the *Coordinated Universal Time* (UTC) standard by adding an extra second to the last minute of the 31st of June. Normally such an adjustment would pass unnoticed by all but a small dedicated collection of time keepers, but this time the story spread out into the popular media as numerous Linux systems hiccupped over this additional second, and they supported some high-profile services, including a major air carrier's reservation and ticketing backend system. The entire topic of time, time standards, and the difficulty of keeping a highly stable and regular clock standard in sync with a slightly wobbly rotating Earth has been a longstanding debate in the *International Telecommunication Union Radiocommunication Sector* (ITU-R) standards body that oversees this coordinated time standard. However, I am not sure that anyone would argue that the challenges of synchronizing a strict time signal with a less than perfectly rotating planet is sufficient reason to discard the concept of a coordinated time standard and just let each computer system drift away on its own concept of time. These days we have become used to a world that operates on a consistent time standard, and we have become used to our computers operating at sub-second accuracy. But how do they do so? In this article I will look at how a consistent time standard is spread across the Internet, and examine the operation of the *Network Time Protocol* (NTP).

Some communications protocols in the IP protocol suite are quite recent, whereas others have a long and rich history that extends back to the start of the Internet. The ARPANET switched over to use the TCP/IP protocol suite in January 1983, and by 1985 NTP was in operation on the network. Indeed it has been asserted that NTP is the longest running, continuously operating, distributed application on the Internet[1].

The objective of NTP is simple: to allow a client to synchronize its clock with UTC time, and to do so with a high degree of accuracy and a high degree of stability. Within the scope of a WAN, NTP will provide an accuracy of small numbers of milliseconds. As the network scope gets finer, the accuracy of NTP can increase, allowing for sub-millisecond accuracy on LANs and sub-microsecond accuracy when using a precision time source such as a *Global Positioning System* (GPS) receiver or a caesium oscillator.

If a collection of clients all use NTP, then this set of clients can operate with a synchronized clock signal. A shared data model, where the modification time of the data is of critical importance, is one example of the use of NTP in a networked context.

(I have relied on NTP timer accuracy at the microsecond level when trying to combine numerous discrete data sources, such as a web log on a server combined with a *Domain Name System* (DNS) query log from DNS resolvers and a packet trace.)

NTP, Time, and Timekeeping

To consider NTP, it is necessary to consider the topic of timekeeping itself. It is useful to introduce some timekeeping terms at this juncture:

Stability	How well a clock can maintain a constant frequency
Accuracy	How well the frequency and absolute value of the clock compares with a standard reference time
Precision	How well the accuracy of a clock can be maintained within a particular timekeeping system
Offset	The time difference in the absolute time of two clocks
Skew	The variation of offset over time (first-order derivative of offset over time)
Drift	The variation of skew over time (second-order derivative of offset over time)

NTP is designed to allow a computer to be aware of three critical metrics for timekeeping: the *offset* of the local clock to a selected reference clock, the *round-trip delay* of the network path between the local computer and the selected reference clock server, and the *dispersion* of the local clock, which is a measure of the maximum error of the local clock relative to the reference clock. Each of these components is maintained separately in NTP. They provide not only precision measurements of offset and delay, to allow the local clock to be adjusted to synchronize with a reference clock signal, but also definitive maximum error bounds of the synchronization process, so that the user interface can determine not only the time, but the quality of the time as well.

Universal Time Standards

It would be reasonable to expect that the time is just the time, but that is not the case. The Universal Time reference standard has several versions, but these two standards are of interest to network timekeeping.

UT1 is the principal form of Universal Time. Although conceptually it is *Mean Solar Time* at 0° longitude, precise measurements of the Sun are difficult. Hence, it is computed from observations of distant quasars using long baseline interferometry, laser ranging of the Moon and artificial satellites, as well as the determination of GPS satellite orbits. UT1 is the same everywhere on Earth, and is proportional to the rotation angle of the Earth with respect to distant quasars, specifically the *International Celestial Reference Frame* (ICRF), neglecting some small adjustments.

The observations allow the determination of a measure of the Earth's angle with respect to the ICRF, called the *Earth Rotation Angle* (ERA), which serves as a modern replacement for *Greenwich Mean Sidereal Time*). UT1 is required to follow the relationship

ERA = 2π(0.7790572732640 + 1.00273781191135448Tu) radians
where Tu = (Julian UT1 date – 2451545.0)

Coordinated Universal Time (UTC) is an atomic timescale that approximates UT1. It is the international standard on which civil time is based. It ticks SI seconds, in step with *International Atomic Time* (TAI). It usually has 86,400 SI seconds per day, but is kept within 0.9 seconds of UT1 by the introduction of occasional intercalary leap seconds. As of 2012 these leaps have always been positive, with a day of 86,401 seconds.[9]

NTP uses UTC, as distinct from the *Greenwich Mean Time* (GMT), as the reference clock standard. UTC uses the TAI time standard, based on the measurement of 1 second as 9,192,631,770 periods of the radiation emitted by a caesium-133 atom in the transition between the two hyperfine levels of its ground state, implying that, like UTC itself, NTP has to incorporate leap second adjustments from time to time.

NTP is an "absolute" time protocol, so that local time zones—and conversion of the absolute time to a calendar date and time with reference to a particular location on the Earth's surface—are not an intrinsic part of the NTP protocol. This conversion from UTC to the wall-clock time, namely the local date and time, is left to the local host.

Servers and Clients

NTP uses the concepts of *server* and *client*. A server is a source of time information, and a client is a system that is attempting to synchronize its clock to a server.

Servers can be either a *primary server* or a *secondary server*. A primary server (sometimes also referred to as a *stratum 1* server using terminology borrowed from the time reference architecture of the telephone network) is a server that receives a UTC time signal directly from an authoritative clock source, such as a configured atomic clock or—very commonly these days—a GPS signal source. A *secondary server* receives its time signal from one or more *upstream servers*, and distributes its time signal to one of more *downstream servers* and *clients*. Secondary servers can be thought of as clock signal repeaters, and their role is to relieve the client query load from the primary servers while still being able to provide their clients with a clock signal of comparable quality to that of the primary servers. The secondary servers need to be arranged in a strict hierarchy in terms of upstream and downstream, and the stratum terminology is often used to assist in this process.

As noted previously, a stratum 1 server receives its time signal from a UTC reference source. A stratum 2 server receives its time signal from a stratum 1 server, a stratum 3 server from stratum 2 servers, and so on. A stratum *n* server can peer with many stratum *n* – 1 servers in order to maintain a reference clock signal. This stratum framework is used to avoid synchronization loops within a set of time servers.

Clients peer with servers in order to synchronize their internal clocks to the NTP time signal.

The NTP Protocol

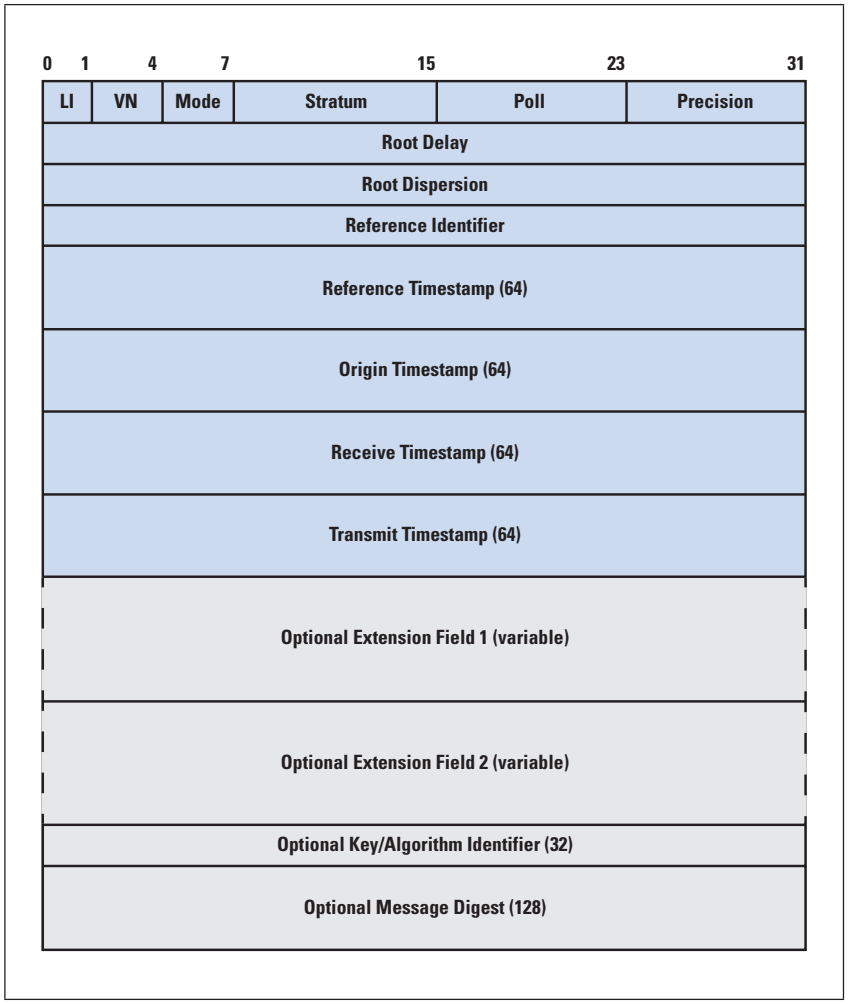
At its most basic, the NTP protocol is a clock request transaction, where a client requests the current time from a server, passing its own time with the request. The server adds its time to the data packet and passes the packet back to the client. When the client receives the packet, the client can derive two essential pieces of information: the reference time at the server and the elapsed time, as measured by the local clock, for a signal to pass from the client to the server and back again. Repeated iterations of this procedure allow the local client to remove the effects of network jitter and thereby gain a stable value for the delay between the local clock and the reference clock standard at the server. This value can then be used to adjust the local clock so that it is synchronized with the server. Further iterations of this protocol exchange can allow the local client to continuously correct the local clock to address local clock skew.

NTP operates over the *User Datagram Protocol* (UDP). An NTP server listens for client NTP packets on port 123. The NTP server is stateless and responds to each received client NTP packet in a simple transactional manner by adding fields to the received packet and passing the packet back to the original sender, without reference to preceding NTP transactions.

Upon receipt of a client NTP packet, the receiver time-stamps receipt of the packet as soon as possible within the packet assembly logic of the server. The packet is then passed to the NTP server process. This process interchanges the IP Header Address and Port fields in the packet, overwrites numerous fields in the NTP packet with local clock values, time-stamps the egress of the packet, recalculates the checksum, and sends the packet back to the client.

The NTP packets sent by the client to the server and the responses from the server to the client use a common format, as shown in Figure 1.

Figure 1: NTP Message Format



The header fields of the NTP message are as follows:

LI Leap Indicator (2 bits)
This field indicates whether the last minute of the current day is to have a leap second applied. The field values follow:

0: No leap second adjustment
1: Last minute of the day has 61 seconds
2: Last minute of the day has 59 seconds
3: Clock is unsynchronized

VN NTP Version Number (3 bits) (current version is 4).

Mode	NTP packet mode (3 bits) The values of the Mode field follow: 0: Reserved 1: Symmetric active 2: Symmetric passive 3: Client 4: Server 5: Broadcast 6: NTP control message 7: Reserved for private use
Stratum	Stratum level of the time source (8 bits) The values of the Stratum field follow: 0: Unspecified or invalid 1: Primary server 2–15: Secondary server 16: Unsynchronized 17–255: Reserved
Poll	Poll interval (8-bit signed integer) The $\log_2$ value of the maximum interval between successive NTP messages, in seconds.
Precision	Clock precision (8-bit signed integer) The precision of the system clock, in $\log_2$ seconds.
Root Delay	The total round-trip delay from the server to the primary reference sourced. The value is a 32-bit signed fixed-point number in units of seconds, with the fraction point between bits 15 and 16. This field is significant only in server messages.
Root Dispersion	The maximum error due to clock frequency tolerance. The value is a 32-bit signed fixed-point number in units of seconds, with the fraction point between bits 15 and 16. This field is significant only in server messages.
Reference Identifier	For stratum 1 servers this value is a four-character ASCII code that describes the external reference source (refer to Figure 2). For secondary servers this value is the 32-bit IPv4 address of the synchronization source, or the first 32 bits of the *Message Digest Algorithm 5* (MD5) hash of the IPv6 address of the synchronization source.

Figure 2: Reference Identifier Codes (from RFC 4330)

Code	External Reference Source
LOCL	uncalibrated local clock
CESM	calibrated Cesium clock
RBDM	calibrated Rubidium clock
PPS	calibrated quartz clock or other pulse-per-second source
IRIG	Inter-Range Instrumentation Group
ACTS	NIST telephone modem service
USNO	USNO telephone modem service
PTB	PTB (Germany) telephone modem service
TDF	Allouis (France) Radio 164 kHz
DCF	Mainflingen (Germany) Radio 77.5 kHz
MSF	Rugby (UK) Radio 60 kHz
WWV	Ft. Collins (US) Radio 2.5, 5, 10, 15, 20 MHz
WWVB	Boulder (US) Radio 60 kHz
WWVH	Kauai Hawaii (US) Radio 2.5, 5, 10, 15 MHz
CHU	Ottawa (Canada) Radio 3330, 7335, 14670 kHz
LORC	LORAN-C radionavigation system
OMEG	OMEGA radionavigation system
GPS	Global Positioning Service

The next four fields use a 64-bit time-stamp value. This value is an unsigned 32-bit seconds value, and a 32-bit fractional part. In this notation the value 2.5 would be represented by the 64-bit string:

0000|0000|0000|0000|0000|0000|0000|0010.|1000|0000|0000|0000|0000|0000|0000|0000

The unit of time is in seconds, and the epoch is 1 January 1900, meaning that the NTP time will cycle in the year 2036 (two years before the 32-bit Unix time cycle event in 2038).

The smallest time fraction that can be represented in this format is 232 picoseconds.

Reference Timestamp This field is the time the system clock was last set or corrected, in 64-bit time-stamp format.

Originate Timestamp This value is the time at which the request departed the client for the server, in 64-bit time-stamp format.

Receive Timestamp This value is the time at which the client request arrived at the server in 64-bit time-stamp format.

Transmit Timestamp This value is the time at which the server reply departed the server, in 64-bit time-stamp format.

The basic operation of the protocol is that a client sends a packet to a server and records the time the packet left the client in the *Origin Timestamp* field (T1). The server records the time the packet was received (T2). A response packet is then assembled with the original Origin Timestamp and the *Receive Timestamp* equal to the packet receive time, and then the *Transmit Timestamp* is set to the time that the message is passed back toward the client (T3). The client then records the time the packet arrived (T4), giving the client four time measurements, as shown in Figure 3.

Figure 3: NTP Transaction Timestamps (from RFC 4330)

Timestamp Name	ID	When Generated
Originate Timestamp	T1	time request sent by client
Receive Timestamp	T2	time request received by server
Transmit Timestamp	T3	time reply sent by server
Destination Timestamp	T4	time reply received by client

These four parameters are passed into the client timekeeping function to drive the clock synchronization function, which we will look at in the next section.

The optional Key and Message Digest fields allow a client and a server to share a secret 128-bit key, and use this shared secret to generate a 128-bit MD5 hash of the key and the NTP message fields. This construct allows a client to detect attempts to inject false responses from a man-in the-middle attack.

The final part of this overview of the protocol operation is the polling frequency algorithm. A NTP client will send a message at regular intervals to a NTP server. This regular interval is commonly set to be 16 seconds. If the server is unreachable, NTP will back off from this polling rate, doubling the back-off time at each unsuccessful poll attempt to a minimum poll rate of 1 poll attempt every 36 hours. When NTP is attempting to resynchronize with a server, it will increase its polling frequency and send a burst of eight packets spaced at 2-second intervals.

When the client clock is operating within a sufficient small offset from the server clock, NTP lengthens the polling interval and sends the eight-packet burst every 4 to 8 minutes (or 256 to 512 seconds).

Timekeeping on the Client

The next part of the operation of NTP is how an NTP process on a client uses the information generated by the periodic polls to a server to moderate the local clock.

From an NTP poll transaction, the client can estimate the delay between the client and the server. Using the time fields described in Figure 3, the transmission delay can be calculated as the total time from transmission of the poll to reception of the response minus the recorded time for the server to process the poll and generate a response:

$$\delta = (T4 - T1) - (T3 - T2)$$

The offset of the client clock from the server clock can also be estimated by the following:

$$\Theta = \tfrac{1}{2}\,[(T2 - T1) + (T3 - T4)]$$

It should be noted that this calculation assumes that the network path delay from the client to the server is the same as the path delay from the server to the client.

NTP uses the minimum of the last eight delay measurements as δ_0. The selected offset, Θ_0, is one measured at the lowest delay. The values (Θ_0,δ_0) become the NTP update value.

When a client is configured with a single server, the client clock is adjusted by a slew operation to bring the offset with the server clock to zero, as long as the server offset value is within an acceptable range.

When a client is configured with numerous servers, the client will use a selection algorithm to select the preferred server to synchronize against from among the candidate servers. Clustering of the time signals is performed to reject outlier servers, and then the algorithm selects the server with the lowest stratum with minimal offset and jitter values. The algorithm used by NTP to perform this operation is *Marzullo's Algorithm*[2].

When NTP is configured on a client, it attempts to keep the client clock synchronized against the reference time standard. To do this task NTP conventionally adjusts the local time by small offsets (larger offsets may cause side effects on running applications, as has been found when processing leap seconds). This small adjustment is undertaken by an *adjtime()* system call, which slews the clock by altering the frequency of the software clock until the time correction is achieved. Slewing the clock is a slow process for large time offsets; a typical slew rate is 0.5 ms per second.

Obviously this informal description has taken a rather complex algorithm and some rather detailed math formulas without addressing the details. If you are interested in how NTP operates at a more detailed level, consult the references that follow, which will take you far deeper into the algorithms and the underlying models of clock selection and synchronization than I have done here.

Conclusion

NTP is in essence an extremely simple stateless transaction protocol that provides a quite surprising outcome. From a regular exchange of simple clock readings between a client and a server, it is possible for the client to train its clock to maintain a high degree of precision despite the possibility of potential problems in the stability and accuracy of the local clock and despite the fact that this time synchronization is occurring over network paths that impose a noise element in the form of jitter in the packet exchange between client and server. Much of today's distributed Internet service infrastructure relies on a common time base, and this base is provided by the common use of the Network Time Protocol.

References and Further Reading

[0] Geoff Huston, "Leaping Seconds," *The Internet Protocol Journal*, Volume 15, No. 3, September 2012.

[1] David L. Mills, "A Brief History of NTP Time: Confessions of an Internet Timekeeper," ACM SIGCOMM, *Computer Communication Review,* Vol. 33, No. 2, pp. 9–12, April 2003, **http://www.eecis.udel.edu/~mills/database/papers/history.pdf**

[2] K. A. Marzullo, "Maintaining the Time in a Distributed System: An Example of a Loosely-Coupled Distributed Service," Ph.D. dissertation, Stanford University, Department of Electrical Engineering, February 1984, **http://en.wikipedia.org/wiki/Marzullo%27s_algorithm**

[3] David L. Mills, "NTP Architecture, Protocol and Algorithms," University of Delaware, **www.eecis.udel.edu/~mills/database/brief/arch/arch.ppt**

[4] Jack Burbank, William Kasch, and David Mills, "Network Time Protocol Version 4: Protocol and Algorithms Specification," RFC 5905, June 2010.

[5] David L. Mills, "Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI," RFC 4330, January 2006.

[6] **http://www.ntp.org**

[7] **http://www.eecis.udel.edu/~mills/ntp.html**

[8] David Mills, *Computer Network Time Synchronization: the Network Time Protocol on Earth and in Space*, Second Edition, CRC Press, 2011.

[9] **http://en.wikipedia.org/wiki/Universal_Time**

Disclaimer

The views expressed are the author's and not those of APNIC, unless APNIC is specifically identified as the author of the communication. APNIC will not be legally responsible in contract, tort or otherwise for any statement made in this publication.

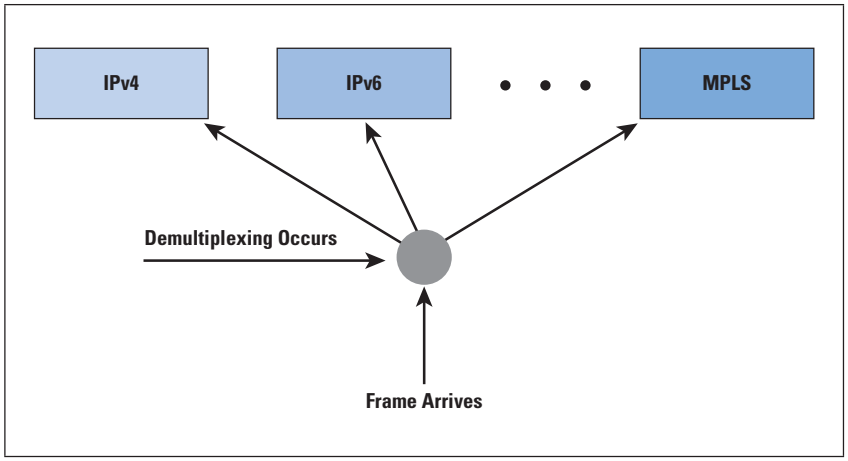
GEOFF HUSTON, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of numerous Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005. He served on the Board of Trustees of the Internet Society from 1992 until 2001.
E-mail: **gih@apnic.net**

Packet Classification: A Faster, More General Alternative to Demultiplexing

by Douglas Comer, Purdue University

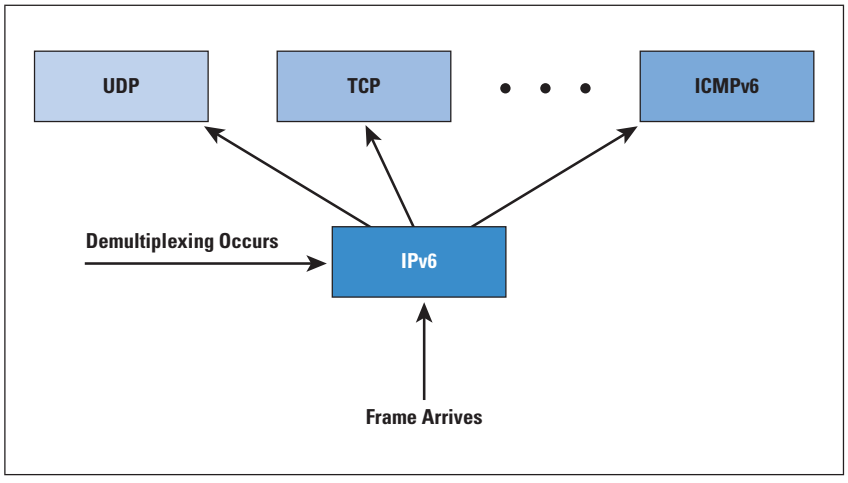
Traditional packet-processing systems use an approach known as *demultiplexing* to handle incoming packets (refer to [1] for details). When a packet arrives, protocol software uses the contents of a *Type Field* in a protocol header to decide how to process the payload in the packet. For example, the Type field in a frame is used to select a Layer 3 module to handle the frame, as Figure 1 illustrates.

Figure 1: Frame Demultiplexing



Demultiplexing is repeated at each level of the protocol stack. For example, IPv6 uses the *Next Header* field to select the correct transport layer protocol module, as Figure 2 illustrates.

Figure 2: Demultiplexing at Layer 3



Modern, high-speed network systems take an entirely different view of packet processing. In place of demultiplexing, they use a technique known as *classification*[2]. Instead of assuming that a packet proceeds through a protocol stack one layer at a time, they allow processing to cross layers. (In addition to being used by companies such as Cisco and Juniper, classification has been used in Linux[3] and with network processors by companies such as Intel and Netronome[4].)

Packet classification is especially pertinent to three key network technologies. First, Ethernet switches use classification instead of demultiplexing when they choose how to forward packets. Second, a router that sends incoming packets over *Multiprotocol Label Switching* (MPLS) tunnels uses classification to choose the appropriate tunnel. Third, classification provides the basis for *Software-Defined Networking* (SDN) and the *OpenFlow* protocol.

Motivation for Classification

To understand the motivation for classification, consider a network system that has protocol software arranged in a traditional layered stack. Packet processing relies on demultiplexing at each layer of the protocol stack. When a frame arrives, protocol software looks at the Type field to learn about the contents of the frame payload. If the frame carries an IP datagram, the payload is sent to the IP protocol module for processing. IP uses the destination address to select a next-hop address. If the datagram is in *transit* (that is, passing through the router on its way to a destination), IP forwards the datagram by sending it back out one of the interfaces. A datagram reaches TCP only if the datagram is destined for the router itself. TCP then uses the protocol port numbers in the TCP segment to further demultiplex the incoming datagram among multiple application programs.

To understand why traditional layering does not solve all problems, consider MPLS processing. In particular, consider a router at the border between a traditional internet and an MPLS core. Such a router must accept packets that arrive from the traditional internet and choose an MPLS path over which to send the packet. Why is layering pertinent to path selection? In many cases, network managers use transport layer protocol port numbers when choosing a path. For example, suppose a manager wants to send all web traffic down a specific MPLS path. All the web traffic will use TCP port 80, meaning that the selection must examine TCP port numbers.

Unfortunately, in a traditional demultiplexing scheme, a datagram does not reach the transport layer unless the datagram is destined for the local network system. Therefore, protocol software must be reorganized to handle MPLS path selection. We can summarize:

> *A traditional protocol stack is insufficient for the task of MPLS path selection because path selection often involves transport layer information and a traditional stack will not send transit datagrams to the transport layer.*

Classification Instead of Demultiplexing

How should protocol software be structured to handle tasks such as MPLS path selection? The answer lies in the use of *classification*. A classification system differs from conventional demultiplexing in two ways:

- Ability to cross multiple layers
- Higher speed than demultiplexing

To understand classification, imagine a packet that has been received at a router and placed in memory. *Encapsulation* means that the packet will have a set of contiguous protocol headers at the beginning. For example, Figure 3 illustrates the headers in a TCP packet (for example, a request sent to a web server) that has arrived over an Ethernet.

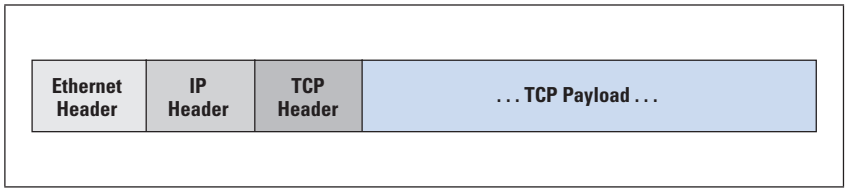


Figure 3: Layout of a Packet in Memory

Given a packet in memory, how can we quickly determine whether the packet is destined to the web? A simplistic approach simply looks at one field in the headers: the TCP destination port number. However, it could be that the packet is not a TCP packet at all. Maybe the frame is carrying *Address Resolution Protocol* (ARP) data instead of IP. Or maybe the frame does indeed contain an IP datagram, but instead of TCP the transport layer protocol is the *User Datagram Protocol* (UDP). To make certain that it is destined for the web, software needs to verify each of the headers: the frame contains an IP datagram, the IP datagram contains a TCP segment, and the TCP segment is destined for the web.

Instead of parsing protocol headers, think of the packet as an array of octets in memory. Consider IPv4 as an example. To be an IPv4 datagram, the Ethernet Type field (located in array positions 12 and 13) must contain `0x0800`. The IPv4 Protocol field, located at position 23, must contain `6` (the protocol number for TCP). The Destination Port field in the TCP header must contain `80`. To know the exact position of the TCP header, we must know the size of the IP header. Therefore, we check the header length octet of the IPv4 header. If the octet contains `0x45`, the TCP destination port number will be found in array positions 36 and 37.

As another example, consider classifying *Voice over IP* (VoIP) traffic that uses the *Real-Time Transport Protocol* (RTP). Because RTP is not assigned a specific UDP port, vendors use a heuristic to determine whether a given packet carries RTP traffic: check the Ethernet and IP headers to verify that the packet carries UDP, and then examine the octets at a known offset in the RTP packet to verify that the value matches the value used by a known codec.

Observe that all the checks described in the preceding paragraphs require only array lookup. That is, the lookup mechanism treats the packet as an array of octets and merely checks to verify that location *X* contains value *Y*, location *Z* contains value *W*, and so on—the mechanism does not need to understand any of the protocol headers or the meaning of values. Furthermore, observe that the lookup scheme crosses multiple layers of the protocol stack.

We use the term *classifier* to describe a mechanism that uses the lookup approach described previously, and we say that the result is a packet *classification*. In practice, a classification mechanism usually takes a list of classification *rules* and applies them until a match is found. For example, a manager might specify three rules: send all web traffic to MPLS path *1*, send all FTP traffic to MPLS path *2*, and send all VPN traffic to MPLS path *3*.

Layering When Classification Is Used

If classification crosses protocol layers, how does it relate to traditional layering diagrams? We can think of classification as an extra layer that has been squeezed between Layer 2 and Layer 3. When a packet arrives, the packet passes from a Layer 2 module to the classification module. All packets proceed to the classifier; no demultiplexing occurs before classification. If any of the classification rules matches the packet, the classification layer follows the rule. Otherwise, the packet proceeds up the traditional protocol stack. For example, Figure 4 illustrates layering when classification is used to send some packets across MPLS paths.

Interestingly, a classification layer can subsume all demultiplexing. That is, instead of classifying packets only for MPLS paths, the classifier can be configured with additional rules that check the Type field in a frame for IPv4, IPv6, ARP, *Reverse ARP* (RARP), and so on.

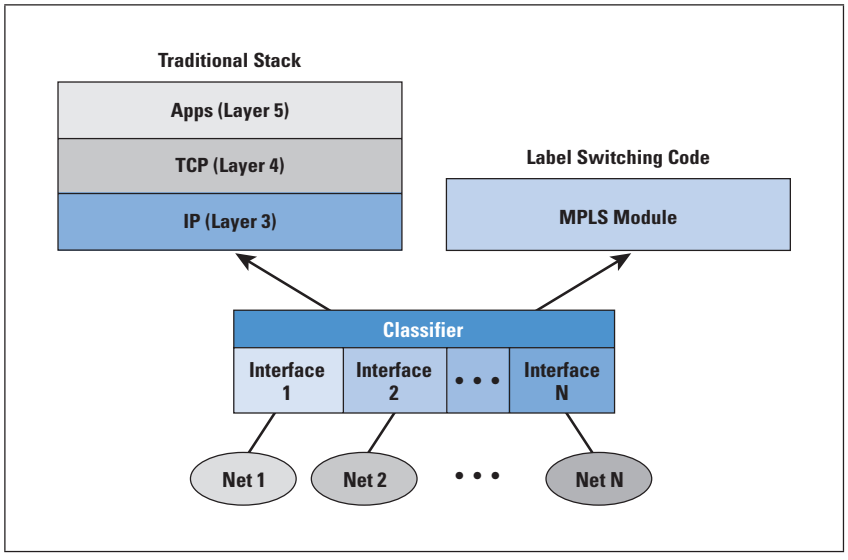


Figure 4: Layering in a Router that Uses Classification to Select MPLS Paths

Classification Hardware and Network Switches

The text in the previous section describes a classification mechanism that is implemented in software—an extra layer is added to a software protocol stack that classifies frames after they arrive at a router. Classification can also be implemented in hardware. In particular, Ethernet switches and other packet-processing hardware devices contain classification hardware that allows packet classification and forwarding to proceed at high speed. The next sections explain hardware classification mechanisms.

We think of network devices, such as switches, as being divided into broad categories by the level of protocol headers they examine and the consequent level of functions they provide:

- Layer 2 Switching
- Layer 2 *Virtual Local-Area Network* (VLAN) Switching
- Layer 3 Switching
- Layer 4 Switching

A *Layer 2 Switch* examines the *Media Access Control* (MAC) source address in each incoming frame to learn the MAC address of the computer that is attached to each port. When a switch learns the MAC addresses of all the attached computers, the switch can use the destination MAC address in each frame to make a forwarding decision. If the frame is unicast, the switch sends only one copy of the frame on the port to which the specified computer is attached. For a frame destined to the broadcast or a multicast address, the switch delivers a copy of the frame to all ports.

A *VLAN Switch* adds one level of virtualization by permitting a manager to assign each port to a specific VLAN. Internally, VLAN switches extend forwarding in a minor way: instead of sending broadcasts and multicasts to all ports on the switch, a VLAN switch consults the VLAN configuration and sends them only to ports on the same VLAN as the source.

A *Layer 3 Switch* acts like a combination of a VLAN switch and a router. Instead of using only the Ethernet header when forwarding a frame, the switch can look at fields in the IP header. In particular, the switch watches the source IP address in incoming packets to learn the IP address of the computer attached to each switch port. The switch can then use the IP destination address in a packet to forward the packet to its correct destination.

A *Layer 4 Device* extends the examination of a packet to the transport layer. That is, the device can include the TCP or UDP Source and Destination Port fields when making a forwarding decision.

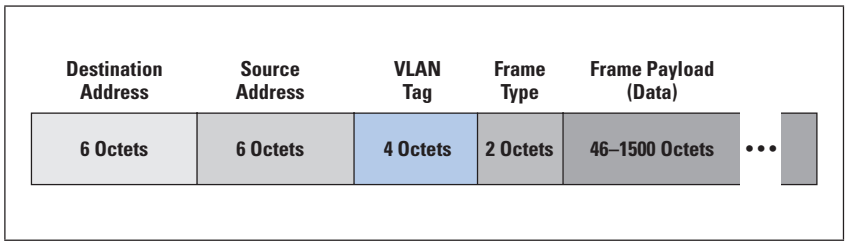
Switching Decisions and VLAN Tags

All types of switching hardware described previously use classification. That is, switches operate on packets as if a packet is merely an array of octets, and individual fields in the packet are specified by giving offsets in the array. Thus, instead of demultiplexing packets, a switch treats a packet syntactically by applying a set of classification rules similar to the rules described previously.

Surprisingly, even VLAN processing is handled in a syntactic manner. Instead of merely keeping VLAN information in a separate data structure that holds meta information, the switch inserts an extra field in an incoming packet and places the VLAN number of the packet in the extra field. Because it is just another field, the classifier can reference the VLAN number just like any other header field.

We use the term *VLAN Tag* to refer to the extra field inserted in a packet. The tag contains the VLAN number that the manager assigned to the port over which the frame arrived. For Ethernet, IEEE standard 802.1Q specifies placing the VLAN Tag field after the MAC Source Address field. Figure 5 illustrates the format.

Figure 5: An Ethernet Frame with a VLAN Tag Inserted



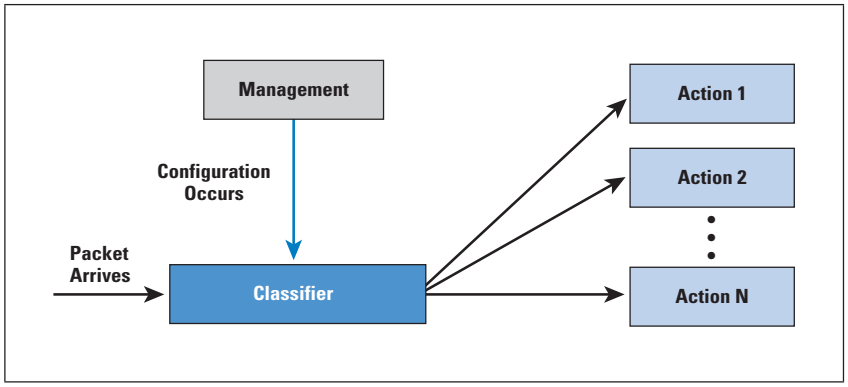
A VLAN tag is used only internally—after the switch has selected an output port and is ready to transmit the frame, the tag is removed. Thus, when computers send and receive frames, the frames do not contain a VLAN tag.

An exception can be made to the rule: a manager can configure one or more ports on a switch to leave VLAN tags in frames when sending the frame. The purpose is to allow two or more switches to be configured to operate as a single, large switch. That is, the switches can share a set of VLANs—a manager can configure each VLAN to include ports on one or both of the switches.

Classification Hardware

We can think of hardware in a switch as being divided into three main components: a classifier, a set of units that perform actions, and a management component that controls the overall operation. Figure 6 illustrates the overall organization and the flow of packets.

Figure 6: Hardware Components Used for Classification



As black arrows in the figure indicate, the classifier provides the high-speed data path that packets follow. When a packet arrives, the classifier uses the rules that have been configured to choose an action. The management module usually consists of a general-purpose processor that runs management software. A network administrator can interact with the management module to configure the switch, in which case the management module can create or modify the set of rules the classifier follows.

A network system, such as a switch, must be able to handle two types of traffic: transit traffic and traffic destined for the switch itself. For example, to provide management or routing functions, a switch may have a local TCP/IP protocol stack and packets destined for the switch must be passed to the local stack. Therefore, one of the actions a classifier takes may be *"pass packet to the local stack for Demultiplexing"*.

High-Speed Classification and TCAM

Modern switches can allow each interface to operate at 10 Gbps. At 10 Gbps, a frame takes only 1.2 microseconds to arrive, and a switch usually has many interfaces. A conventional processor cannot handle classification at such speeds, so a question arises: how can a hardware classifier achieve high speed? The answer lies in a hardware technology known as *Ternary Content Addressable Memory* (TCAM).

TCAM uses parallelism to achieve high speed—instead of testing one field of a packet at a given time, TCAM checks all fields simultaneously. Furthermore, TCAM performs multiple checks at the same time. To understand how TCAM works, think of a packet as a string of bits. We imagine TCAM hardware as having two parts: one part holds the bits from a packet and the other part is an array of values that will be compared to the packet. Entries in the array are known as *slots*. Figure 7 illustrates the idea.

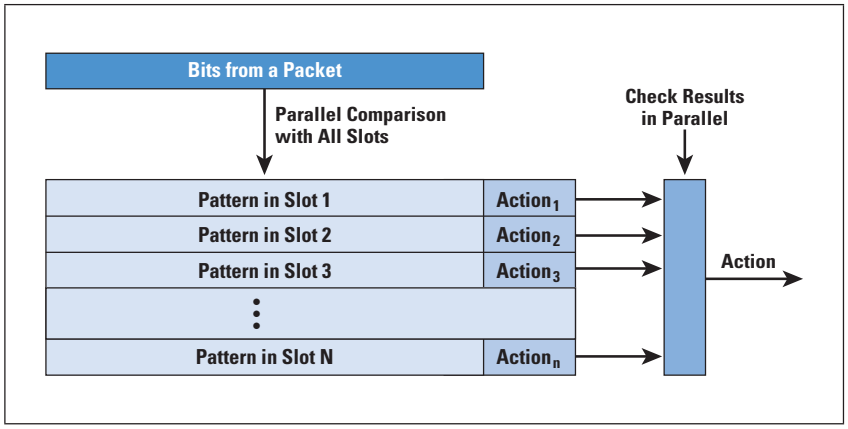


Figure 7: The Conceptual Organization of TCAM

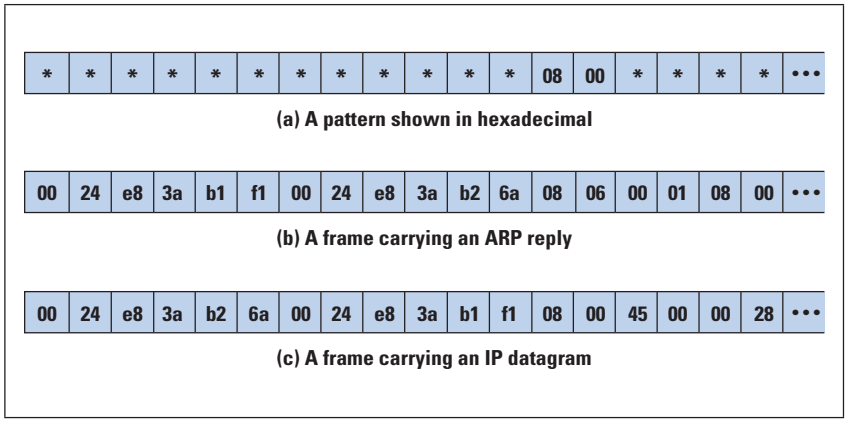
In the figure, each slot contains two parts. The first part consists of hardware that compares the bits from the packet to the pattern stored in the slot. The second part stores a value that specifies an action to be taken if the pattern matches the packet. If a match occurs, the slot hardware passes the action to the component that checks all the results and announces an answer.

One of the most important details concerns the way TCAM handles multiple matches. In essence, the output circuitry selects one match and ignores the others. That is, if multiple slots each pass an action to the output circuit, the circuit accepts only one and passes the action as the output of the classification. For example, the hardware may choose the lowest slot that matches. In any case, the action that the TCAM announces corresponds to the action from one of the matching slots.

The figure indicates that a slot holds a *pattern* rather than an exact value. Instead of merely comparing each bit in the pattern to the corresponding bit in the packet, the hardware performs a pattern match. The adjective *ternary* is used because each bit position in a pattern can have three possible values: a one, a zero, or a "don't care". When a slot compares its pattern to the packet, the hardware checks only the one and zero bits in the pattern—the hardware ignores pattern bits that contain "don't care". Thus, a pattern can specify exact values for some fields in a packet header and omit other fields.

To understand TCAM pattern matching, consider a pattern that identifies IP packets. Identifying such packets is easy because an Ethernet frame that carries an IPv4 datagram will have the value `0x0800` in the Ethernet Type field. Furthermore, the Type field occupies a fixed position in the frame: bits 96 through 111. Thus, we can create a pattern that starts with 96 "don't care" bits (to cover the Ethernet destination and source MAC addresses) followed by 16 bits with the binary value `0000100000000000` (the binary equivalent of `0x0800`) to cover the Type field. All remaining bit positions in the pattern will be "don't care". Figure 8 illustrates the pattern and example packets.

Figure 8: A TCAM Pattern and Example Packets



Although a TCAM hardware slot has one position for each bit, the figure does not display individual bits. Instead, each box corresponds to one octet, and the value in a box is a hexadecimal value that corresponds to 8 bits. We use hexadecimal simply because binary strings are too long to fit into a figure comfortably.

The Size of a TCAM

A question arises: how large is a TCAM? The question can be divided into two important aspects:

- *The number of bits in a slot:* The number of bits per slot depends on the type of Ethernet switch. A basic switch uses the destination MAC address to classify a packet. Because a MAC address is 48 bits, TCAM in a basic switch needs only 48 bit positions. A VLAN switch needs 128 bit positions to cover the VLAN tag as well as source and destination MAC addresses. A Layer 3 switch must have sufficient bit positions to cover the IP header as well as the Ethernet header. For IPv6, the header size is large and variable—in most cases, a pattern will need to cover extension headers as well as the base header.
- *The total number of slots:* The total number of TCAM slots determines the maximum number of patterns a classifier can hold. When a switch learns the MAC address of a computer that has been plugged into a port, the switch can store a pattern for the address. For example, if a computer with MAC address X is plugged into port 29, the switch can create a pattern in which destination address bits match X and the action is *"send packet to output port 29"*.

A switch can also use patterns to control broadcasting. When a manager configures a VLAN, the switch can add an entry for the VLAN broadcast. For example, if a manager configures VLAN 9, an entry can be added in which the destination address bits are all 1s (that is, the Ethernet broadcast address) and the VLAN tag is 9. The action associated with the entry is *"broadcast on VLAN 9"*.

A Layer 3 switch can learn the IP source address of computers attached to the switch, and can use TCAM to store an entry for each IP address. Similarly, it is possible to create entries that match Layer 4 protocol port numbers (for example, to direct all web traffic to a specific output). SDN technologies allow a manager to place patterns in the classifier to establish paths through a network and direct traffic along the paths. Because such classification rules cross multiple layers of the protocol stack, the potential number of items stored in a TCAM can be large.

TCAM seems like an ideal mechanism because it is both extremely fast and versatile. However, TCAM has two significant drawbacks: cost and heat. The cost is high because TCAM has parallel hardware for each slot and the overall system is designed to operate at high speed. In addition, because it operates in parallel, TCAM consumes much more energy than conventional memory (and generates more heat). Therefore, designers minimize the amount of TCAM to keep costs and power consumption low. A typical switch has 32,000 entries.

Classification-Enabled Generalized Forwarding

Perhaps the most significant advantage of a classification mechanism arises from the generalizations it enables. Because classification examines arbitrary fields in a packet before any demultiplexing occurs, cross-layer combinations are possible. For example, classification can specify that all packets from a given MAC address should be forwarded to a specific output port regardless of the packet contents. In addition, classification can make forwarding decisions depend on combinations of source and destination. An *Internet Service Provider* (ISP) can choose to forward all packets with IP source address X that are destined for web server W along one path while forwarding packets with IP source address Y that are destined to the same web server along another path.

ISPs need the generality that classification offers to handle traffic engineering that is not usually available in a conventional protocol stack. In particular, classification allows an ISP to offer tiered services in which the path a packet follows depends on a combination of the type of traffic and how much the customer pays.

Summary

Classification is a fundamental performance optimization that allows a packet-processing system to cross layers of the protocol stack without demultiplexing. A classifier treats each packet as an array of bits and checks the contents of fields at specific locations in the array.

Classification offers high-speed forwarding for network systems such as Ethernet switches and routers that send packets across MPLS tunnels. To achieve the highest speed, classification can be implemented in hardware; a hardware technology known as TCAM is especially useful because it employs parallelism to perform classification at extremely high speed.

The generalized forwarding capabilities that classification provides allow ISPs to perform traffic engineering. When making a forwarding decision, a classification mechanism can use the source of a packet as well as the destination (for example, to choose a path based on the tier of service to which a customer subscribes).

Acknowledgment

Material in this article has been taken with permission from Douglas E. Comer, *Internetworking With TCP/IP Volume 1: Principles, Protocols, and Architecture,* Sixth edition, 2013.

References

[1] Douglas E. Comer and David L. Stevens, *Internetworking With TCP/IP Volume 2: Design, Implementation, and Internals,* Prentice-Hall, Upper Saddle River, NJ, Third edition, 1999.

[2] **yuba.stanford.edu/~nickm/papers/classification_tutorial_01.pdf**

[3] Patrick McHardy, "nfttables: A Successor to iptables, ip6tables, ebtables and arptables," *Netfilter Workshop 2008,* Paris, 2008.

[4] Douglas E. Comer, *Network Systems Design Using Network Processors,* Intel IXP 2xxx version, Prentice-Hall, Upper Saddle River, NJ, 2006.

DOUGLAS E. COMER is a Distinguished Professor of Computer Science at Purdue University. Formerly, he served as VP of Research and Research Collaboration at Cisco Systems. As a member of the original IAB, he participated in early work on the Internet, and is internationally recognized as an authority on TCP/IP protocols and Internet technologies. He has written a series of best-selling technical books, and his three-volume *Internetworking* series is cited as an authoritative work on Internet technologies. His books, which have been translated into 16 languages, are used in industry and academia in many countries. Comer consults for industry, and has lectured to thousands of professional engineers and students around the world. For 20 years he was editor-in-chief of the journal *Software—Practice and Experience.* He is a Fellow of the ACM and the recipient of numerous teaching awards. E-mail: **comer@cs.purdue.edu**

Fragments

Internet Society Disappointed over Fundamental Divides at WCIT-12

On December 14, 2012, The Internet Society released the following statement from President and CEO Lynn St. Amour:

"The Internet Society, like other participants at the *World Conference on International Telecommunications* (WCIT), came to this conference looking for a successful outcome. We were hopeful that it would result in a treaty that would enable growth, further innovation, and advance interoperability in international telecommunications. It was extremely important that this treaty not extend to content, or implicitly or explicitly undermine the principles that have made the Internet so beneficial.

While progress was made in some areas such as transparency in international roaming fees, fundamental divides were exposed leaving a significant number of countries unable to sign the *International Telecommunication Regulations* (ITRs). Statements made by a host of delegations today made it very clear that Internet issues did not belong in the ITRs and that they would not support a treaty that is inconsistent with the multi-stakeholder model of Internet Governance.

We are disappointed that the conference has not been successful in reaching consensus. The Internet Society is dedicated to working with all stakeholders around the world to create the environment that will allow the Internet to grow for the betterment of all people."

For more information, see:
http://www.internetsociety.org/wcit

See also:

[0] Geoff Huston, "December in Dubai," *The Internet Protocol Journal*, Volume 15, No. 2, June 2012.

[1] World Conference on International Telecommunications (WCIT-12), **http://www.itu.int/en/wcit-12/Pages/default.aspx**

[2] "NRO contribution to the WCIT Public Consultation Process," **http://www.nro.net/wp-content/uploads/2012/joint-submission-WCIT-RIR.pdf**

[3] "Stop the Net Grab": NRO Shares Concerns About the WCIT Process," **http://www.nro.net/news/nro-shares-concerns-aboutwcit-process**

[4] WCITLeaks.org "Bringing transparency to the ITU," **http://wcitleaks.org**

NRO Observations on WCIT-12 Process

The *Number Resource Organization* (NRO), representing the world's five *Regional Internet address Registries* (RIRs), issued the following statement from Dubai, the site of the recent *World Conference on International Telecommunications* (WCIT):

The conference has clearly not met expectations of many *International Telecommunications Union* (ITU) Member States, and with this unfortunate outcome now clear, we feel compelled to put the following observations on record.

The NRO is concerned about aspects of the WCIT-12 meetings, which have just ended in Dubai, particularly with events in the last days of the conference. Neither the content of this conference, nor its conduct during this critical final period, have met community expectations or satisfied public assurances given prior to the event.

Internet stakeholders around the world watched the WCIT preparations closely, and were hopeful, throughout those processes, of two things: that WCIT would have no bearing on the Internet, its governance or its content; and that the event would allow all voices to be heard. The ITU Secretary General himself made these assurances on multiple occasions, and reiterated them in his opening remarks to the conference.

Regrettably, expected WCIT discussions on traditional telecommunication issues were eclipsed by debates about Internet-related issues. The intensity and length of these debates revealed clearly the depth of genuine concern about the proposals, and also the determination of those who brought them to the meeting.

Perhaps more importantly, an open multi-stakeholder conduct of the WCIT conference did not eventuate. Plenary sessions of the conference were webcast, but contributions were allowed only from official Government delegates and ITU officials, relegating all other stakeholders to an observer role.

Furthermore, an important number of critical negotiations occurred in small groups accessible only to Member States; and key experts and other stakeholders were unable even to observe them.

The NRO strongly supports the principles established in 2005 by the *World Summit on the Information Society,* which call for Internet Governance to be carried out in a multi-stakeholder manner, and we note that these represent the view of the global community as expressed through the United Nations system itself.

The NRO has also participated in many ITU conferences and study groups over the years, at very substantial cost, in genuine efforts to build relationships between our communities and to demonstrate the value of multi-stakeholder cooperation and collaboration. The NRO will continue to participate in the ITU, itself a member of the UN system, in expectation that its processes can evolve visibly, and much more rapidly, towards these accepted principles.

John Jason Brzozowski, Donn Lee, and Paul Saab win 2012 Itojun Awards

The fourth annual *Itojun Service Awards* were recently presented to John Jason Brzozowski for his tireless efforts in providing IPv6 connectivity to cable broadband users across North America and evangelizing the importance of IPv6 deployment globally, and to Donn Lee and Paul Saab for their efforts in making high-profile online content available over IPv6 and for their key contributions to *World IPv6 Day* and *World IPv6 Launch*. The awardees were recognized at the *Internet Engineering Task Force* (IETF) 85 meeting in November 2012 in Atlanta, Georgia.

First awarded in 2009, the award honors the memory of Dr. Junichiro "Itojun" Hagino, who passed away in 2007 at the age of 37. The award, established by the friends of Itojun and administered by the Internet Society, recognizes and commemorates the extraordinary dedication exercised by Itojun over the course of IPv6 development. IPv6, the next-generation Internet protocol developed within the IETF, provides more than 340 trillion, trillion, trillion addresses, enabling billions of people and a huge range of devices to connect with one another, and helping ensure the Internet continues its current growth rate indefinitely.

"The combined work of John, Donn, and Paul has made IPv6 a technology used every day by people around the world as they access some of the most popular websites from their homes and offices," said Jun Murai of the Itojun Service Award committee and founder of the WIDE Project.

"On behalf of the Itojun Service Award committee, I am extremely pleased to present this award to them for their ongoing efforts that have made IPv6 a mainstream technology for global web companies looking to ensure their continued growth."

The Itojun Service Award is focused on pragmatic contributions to developing and deploying IPv6 in the spirit of serving the Internet. With respect to the spirit, the selection committee seeks contributors to the Internet as a whole; open source developers are a common example of such contributors, although this is not a requirement for expected nominees.

While the committee primarily considers practical contributions such as software development or network operation, higher level efforts that help those direct contributions will also be appreciated in this regard. The contribution should be substantial, but could be at an immature stage or be ongoing; this award aims to encourage the contributor to continue their efforts, rather than just recognizing well established work. Finally, contributions of a group of individuals will be accepted, as deployment work is often done by a large project, not just a single outstanding individual.

The award includes a presentation crystal, a US$3,000 honorarium, and a travel grant.

John Jason Brzozowski said, "It is truly humbling to be a recipient of the Itojun Service Award, being recognized with others that have worked tirelessly to make IPv6 a reality is rewarding personally and professionally. I would like to thank the award committee and the Internet Society as well as my family and co-workers for their support. As many are aware, the IPv6 journey at Comcast has been unfolding since 2005. It is an honor and pleasure to provide the technical and strategic leadership for IPv6 that has led to the success of our program and the widespread adoption of IPv6."

Donn Lee said, "Deploying IPv6 continues to be an amazing experience. I'm thankful to be sharing this award with my colleagues Paul and John, whom I have worked alongside through the challenging and exciting milestones of World IPv6 Day 2011 and World IPv6 Launch 2012. I especially want to thank the award committee for this honor that remembers Itojun, a truly inspirational IPv6 scientist, leader, and visionary."

Paul Saab said, "I'm honored to be sharing the Itojun Service Award with Donn and John. We should never forget that we would not be here today if it were not for Itojun's trailblazing work and passion for IPv6. To be recognized is extremely humbling, as Facebook's participation could not have been done without our amazing co-workers and their own hard work to bring IPv6 to our users. Thank you for recognizing us and remember that this journey is only 2% complete."

For more information about the Itojun Service Award see:
http://www.internetsociety.org/what-we-do/grants-and-awards/awards/itojun-service-award

Left to right: Jun Murai, John Jason Brzozowski, Paul Saab and Don Lee

Leading Global Standards Organizations Endorse "OpenStand" Principles

Five leading global organizations—the *Institute for Electrical and Electronics Engineers* (IEEE), the *Internet Architecture Board* (IAB), *the Internet Engineering Task Force* (IETF), the *Internet Society* and the *World Wide Web Consortium* (W3C)—recently announced that they have signed a statement affirming the importance of a jointly developed set of principles establishing a modern paradigm for global, open standards. The shared "OpenStand" principles—based on the effective and efficient standardization processes that have made the Internet and Web the premiere platforms for innovation and borderless commerce—are proven in their ability to foster competition and cooperation, support innovation and interoperability and drive market success.

The IEEE, IAB, IETF, Internet Society and W3C invite other standards organizations, governments, corporations and technology innovators globally to endorse the principles, available at `open-stand.org`

The OpenStand principles strive to encapsulate that successful standardization model and make it extendable across the contemporary, global economy's gamut of technology spaces and markets. The principles comprise a modern paradigm in which the economics of global markets—fueled by technological innovation—drive global deployment of standards, regardless of their formal status within traditional bodies of national representation. The OpenStand principles demand:

- Cooperation among standards organizations;
- Adherence to due process, broad consensus, transparency, balance and openness in standards development;
- Commitment to technical merit, interoperability, competition, innovation and benefit to humanity;
- Availability of standards to all; and
- Voluntary adoption.

"New dynamics and pressures on global industry have driven changes in the ways that standards are developed and adopted around the world," said Steve Mills, president of the IEEE Standards Association.

"Increasing globalization of markets, the rapid advancement of technology and intensifying time-to-market demands have forced industry to seek more efficient ways to define the global standards that help expand global markets. The OpenStand principles foster the more efficient international standardization paradigm that the world needs."

Added Leslie Daigle, chief Internet technology officer with the Internet Society: "International standards development for borderless economics is not ad hoc; rather, it has a paradigm—one that has demonstrated agility and is driven by technical merit.

The OpenStand principles convey the power of bottom-up collaboration in harnessing global creativity and expertise to the standards of any technology space that will underpin the modern economy moving forward."

Standards developed and adopted via the OpenStand principles include IEEE standards for the Internet's physical connectivity, IETF standards for end-to-end global Internet interoperability and the W3C standards for the World Wide Web.

"The Internet and World Wide Web have fueled an economic and social transformation, touching billions of lives. Efficient standardization of so many technologies has been key to the success of the global Internet," said Russ Housley, IETF chair. "These global standards were developed with a focus toward technical excellence and deployed through collaboration of many participants from all around the world. The results have literally changed the world, surpassing anything that has ever been achieved through any other standards-development model."

Globally adopted design-automation standards, which have paved the way for a giant leap forward in industry's ability to define complex electronic solutions, provide another example of standards developed in the spirit of the OpenStand principles. Another technology space that figures to demand such standards over the next decades is the global smart-grid effort, which seeks to augment regional facilities for electricity generation, distribution, delivery and consumption with a two-way, end-to-end network for communications and control.

"Think about all that the Internet and Web have enabled over the past 30 years, completely transforming society, government and commerce," said W3C chief executive officer Jeff Jaffe. "It is remarkable that a small number of organizations following a small number of principles have had such a huge impact on humanity, innovation and competition in global markets."

Bernard Aboba, chair of the IAB said: "The Internet has been built on specifications adopted voluntarily across the globe. By valuing running code, interoperability and deployment above formal status, the Internet has democratized the development of standards, enabling specifications originally developed outside of standards organizations to gain recognition based on their technical merit and adoption, contributing to the creation of global communities benefiting humanity. We now invite standards organizations, as well as governments, companies and individuals to join us at **`open-stand.org`** in order to affirm the principles that have nurtured the Internet and underpin many other important standards—and will continue to do so."

New Year's Day 2013 Marks 30th Anniversary of Major Milestone for the Internet

On January 1, 1983, the ARPANET, a direct predecessor of today's Internet, implemented the *Transmission Control Protocol/Internet Protocol* (TCP/IP) in a transition that required all connected computers to convert to the protocol simultaneously. The open TCP/IP protocol is now a foundational technology for the networks around the world that make up the global Internet and interconnect billions of devices. The transition, which was carefully planned over several years before it actually took place, is documented in RFC 801[1] authored by Jon Postel[2].

Throughout its history, the Internet has continued to evolve. Today, deploying IPv6, the latest generation of the IP protocol, is critical to ensuring the Internet's continued growth and to connect the billions of people not yet online. Thousands of major *Internet Service Providers* (ISPs), home networking equipment manufacturers, and web companies around the world are coming together to permanently enable IPv6 for their products and services through efforts such as *World IPv6 Launch*[3] organized by the Internet Society.

For more information about the Internet Society's work to facilitate the open development of standards, protocols, and administration, and to ensure a robust, secure technical infrastructure, see the *Internet Technology Matters* blog[4] and the *Deploy360 Programme*[5]. For further details about the Internet's history and development, see [6].

[1] Jon Postel, "NCP/TCP transition plan," RFC 801, November 1981.

[2] **http://www.internethalloffame.org/inductees/jon-postel**

[3] **http://www.worldipv6launch.org/**

[4] **http://www.internetsociety.org/what-we-do/internet-technology-matters**

[5] **http://www.internetsociety.org/deploy360/**

[6] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, and Stephen Wolff, "Brief History of the Internet," **http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet**

IPJ Subscription FAQ

What is my "Subscription ID" for The Internet Protocol Journal (IPJ) and where do I find it?

Your Subscription ID is a unique combination of letters and numbers used to locate your subscription in our database. It is printed on the back of your IPJ issue or on the envelope. You will also find information about your subscription expiration date near your Subscription ID. Here is an example:

How do I renew or update my subscription?

From the IPJ homepage (**www.cisco.com/ipj**) click "Subscriber Service" and then enter your Subscription ID and your e-mail address in the boxes. After you click "Login" the system will send you an e-mail message with a unique URL that allows access to your subscription record. You can then update your postal and e-mail details, change delivery options, and of course *renew* your subscription.

What will you use my e-mail address and postal address for?

This information is used *only* to communicate with you regarding your subscription. You will receive renewal reminders as well as other information about your subscription. We will never use your address for any form of marketing or unsolicited e-mail.

I didn't receive the special URL that allows me to renew or update my Subscription. Why?

This is likely due to some form of spam filtering. Just send an e-mail message to **ipj@cisco.com** with your Subscription ID and any necessary changes and we will make the changes for you.

Do I need my Subscription ID to read IPJ online? What is my username and password?

Your Subscription ID is used *only* for access to your subscription record. No username or password is required to read IPJ. All back issues are available for online browsing or for download at **www.cisco.com/ipj**

I can't find my Subscription ID and I have since changed e-mail address anyway; what do I do now?

Just send a message to **ipj@cisco.com** and we will take care of it for you.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

March 2013 Volume 16, Number 1

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

This is the 60th edition of *The Internet Protocol Journal,* and in June we will celebrate our 15th anniversary. Fifteen years is not a long time in absolute terms, but when it comes to networking technology a lot can happen in a short time.

Throughout this 15-year period we have published numerous articles on "emerging technologies," and in this issue we present yet another. *Software-Defined Networks* (SDNs) have become a mainstream topic for research, development, and standardization. We asked William Stallings to give us an overview of SDNs, and we plan further articles on this topic in the future.

A recurring theme in this journal has been Internet *security* at all levels of the protocol stack. We have covered security in routing, securing the *Domain Name System* (DNS), secure wireless networks, secure HTTP, and much more. This time, Scott Hogg discusses the advantages and disadvantages of using IPv4 or IPv6 addresses as a form of user authentication.

In our previous issue we published some reactions to the outcomes of the *World Conference on International Telecommunications* (WCIT) held in Dubai in December 2012. In this edition, Robert Pepper and Chip Sharp provide analysis and background on this conference and discuss how the revised *International Telecommunication Regulations* (ITRs) might affect the future of the Internet.

It has been some time since we have published a book review, but we are happy to bring you one in this issue. For the first time in history, we are reviewing a book that exists only in electronic form, another sign of a rapidly changing technology landscape. We are always looking for new book reviews. Please send your reviews, letters to the editor, or any subscription questions to **ipj@cisco.com**

If you want to look back at 15 years of IPJ, visit our website at **www.cisco.com/ipj** where you will find all of our back issues (as a single PDF file, as a collection of individual PDF files, or in HTML format), as well as an index of all IPJ articles.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Software-Defined Networks and OpenFlow

by William Stallings

A network organizing technique that has come to recent prominence is the *Software-Defined Network* (SDN)[1]. In essence, an SDN separates the data and control functions of networking devices, such as routers, packet switches, and LAN switches, with a well-defined *Application Programming Interface* (API) between the two. In contrast, in most large enterprise networks, routers and other network devices encompass both data and control functions, making it difficult to adjust the network infrastructure and operation to large-scale addition of end systems, virtual machines, and virtual networks. In this article we examine the characteristics of an SDN, and then describe the *OpenFlow* specification, which is becoming the standard way of implementing an SDN.

Evolving Network Requirements

Before looking in more detail at SDNs, let us examine the evolving network requirements that lead to a demand for a flexible, response approach to controlling traffic flows within a network or the Internet.

One key leading factor is the increasingly widespread use of *Server Virtualization*. In essence, server virtualization masks server resources, including the number and identity of individual physical servers, processors, and operating systems, from server users. This masking makes it possible to partition a single machine into multiple, independent servers, conserving hardware resources. It also makes it possible to migrate a server quickly from one machine to another for load balancing or for dynamic switchover in the case of machine failure. Server virtualization has become a central element in dealing with "big data" applications and in implementing cloud computing infrastructures. But it creates problems with traditional network architectures (for example, refer to [2]). One problem is configuring *Virtual LANs* (VLANs). Network managers need to make sure the VLAN used by the *Virtual Machine* is assigned to the same switch port as the physical server running the virtual machine. But with the virtual machine being movable, it is necessary to reconfigure the VLAN every time that a virtual server is moved. In general terms, to match the flexibility of server virtualization, the network manager needs to be able to dynamically add, drop, and change network resources and profiles. This process is difficult to do with conventional network switches, in which the control logic for each switch is co-located with the switching logic.

Another effect of server virtualization is that traffic flows differ substantially from the traditional client-server model. Typically, there is a considerable amount of traffic among virtual servers, for such purposes as maintaining consistent images of the database and invoking security functions such as access control. These server-to-server flows change in location and intensity over time, demanding a flexible approach to managing network resources.

Another factor leading to the need for rapid response in allocating network resources is the increasing use by employees of mobile devices such as smartphones, tablets, and notebooks to access enterprise resources. Network managers must be able to respond to rapidly changing resource, *Quality of Service* (QoS), and security requirements.

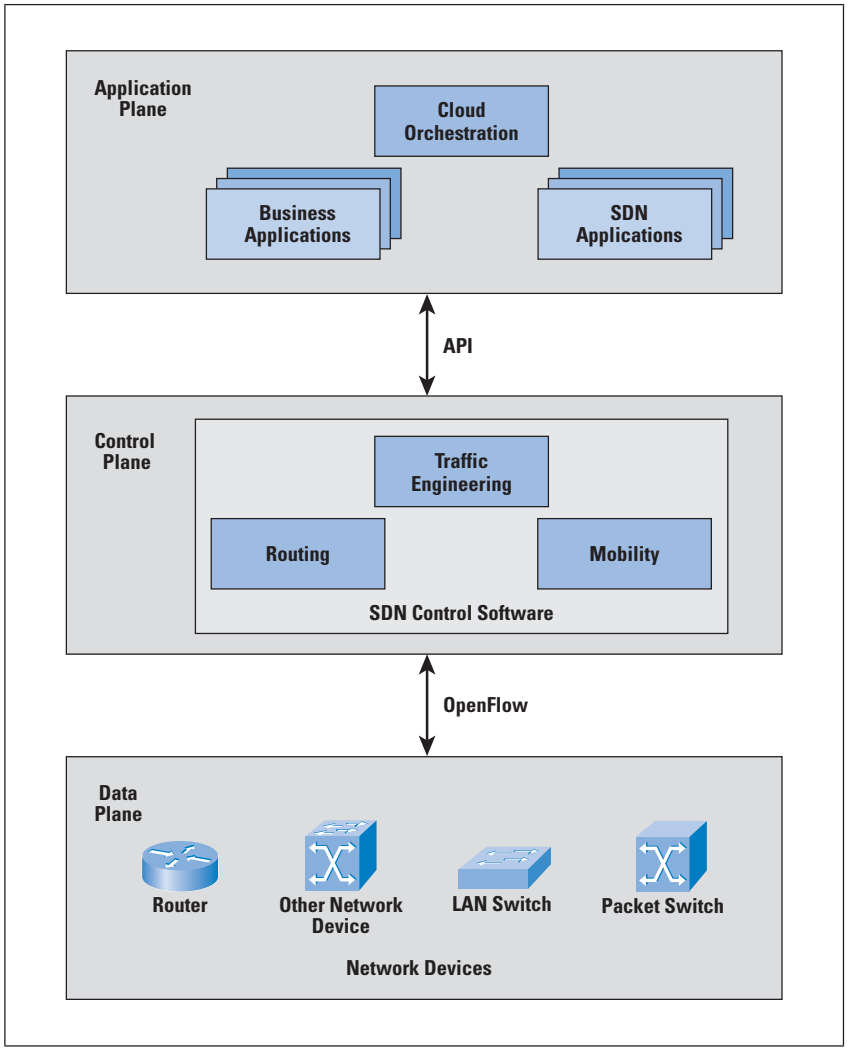
Existing network infrastructures can respond to changing requirements for the management of traffic flows, providing differentiated QoS levels and security levels for individual flows, but the process can be very time-consuming if the enterprise network is large and/or involves network devices from multiple vendors. The network manager must configure each vendor's equipment separately, and adjust performance and security parameters on a per-session, per-application basis. In a large enterprise, every time a new virtual machine is brought up, it can take hours or even days for network managers to do the necessary reconfiguration[3].

This state of affairs has been compared to the mainframe era of computing[4]. In the era of the mainframe, applications, the operating system, and the hardware were vertically integrated and provided by a single vendor. All of these ingredients were proprietary and closed, leading to slow innovation. Today, most computer platforms use the x86 instruction set, and a variety of operating systems (Windows, Linux, or Mac OS) run on top of the hardware. The OS provides APIs that enable outside providers to develop applications, leading to rapid innovation and deployment. In a similar fashion, commercial networking devices have proprietary features and specialized control planes and hardware, all vertically integrated on the switch. As will be seen, the SDN architecture and the OpenFlow standard provide an open architecture in which control functions are separated from the network device and placed in accessible control servers. This setup enables the underlying infrastructure to be abstracted for applications and network services, enabling the network to be treated as a logical entity.

SDN Architecture

Figure 1 illustrates the logical structure of an SDN. A central controller performs all complex functions, including routing, naming, policy declaration, and security checks. This plane constitutes the *SDN Control Plane,* and consists of one or more SDN servers.

Figure 1: SDN Logical Structure



The *SDN Controller* defines the data flows that occur in the *SDN Data Plane*. Each flow through the network must first get permission from the controller, which verifies that the communication is permissible by the network policy. If the controller allows a flow, it computes a route for the flow to take, and adds an entry for that flow in each of the switches along the path. With all complex functions subsumed by the controller, switches simply manage flow tables whose entries can be populated only by the controller. Communication between the controller and the switches uses a standardized protocol and API. Most commonly this interface is the OpenFlow specification, discussed subsequently.

The SDN architecture is remarkably flexible; it can operate with different types of switches and at different protocol layers. SDN controllers and switches can be implemented for Ethernet switches (Layer 2), Internet routers (Layer 3), transport (Layer 4) switching, or application layer switching and routing. SDN relies on the common functions found on networking devices, which essentially involve forwarding packets based on some form of flow definition.

In an SDN architecture, a switch performs the following functions:

- The switch encapsulates and forwards the first packet of a flow to an SDN controller, enabling the controller to decide whether the flow should be added to the switch flow table.
- The switch forwards incoming packets out the appropriate port based on the flow table. The flow table may include priority information dictated by the controller.
- The switch can drop packets on a particular flow, temporarily or permanently, as dictated by the controller. Packet dropping can be used for security purposes, curbing *Denial-of-Service* (DoS) attacks or traffic management requirements.

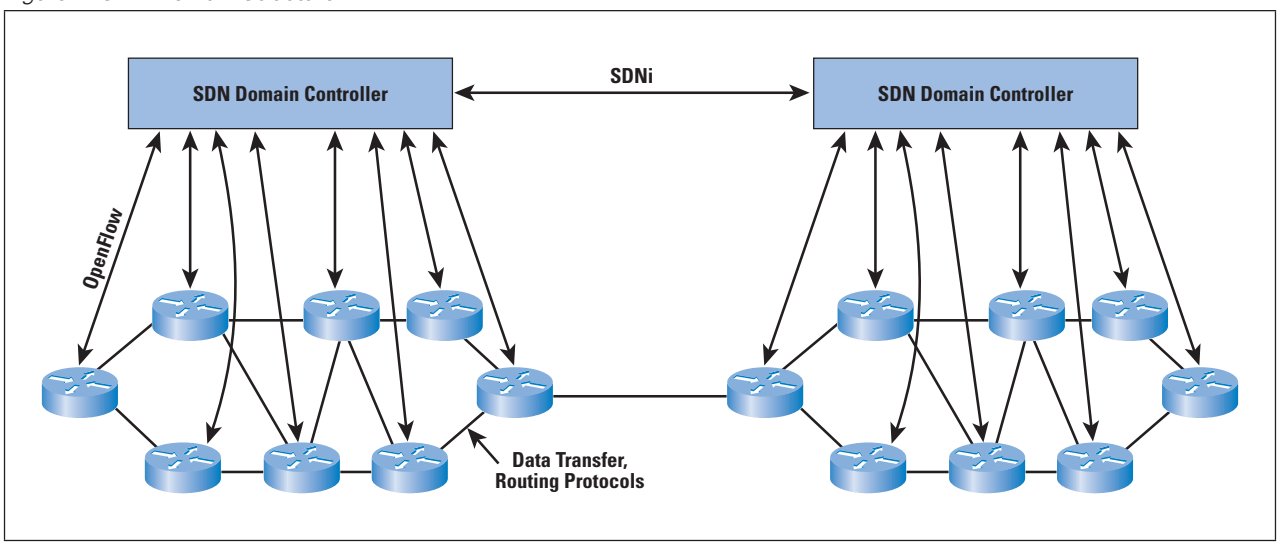
In simple terms, the SDN controller manages the forwarding state of the switches in the SDN. This management is done through a vendor-neutral API that allows the controller to address a wide variety of operator requirements without changing any of the lower-level aspects of the network, including topology.

With the decoupling of the control and data planes, SDN enables applications to deal with a single abstracted network device without concern for the details of how the device operates. Network applications see a single API to the controller. Thus it is possible to quickly create and deploy new applications to orchestrate network traffic flow to meet specific enterprise requirements for performance or security.

SDN Domains

In a large enterprise network, the deployment of a single controller to manage all network devices would prove unwieldy or undesirable. A more likely scenario is that the operator of a large enterprise or carrier network divides the whole network into numerous nonoverlapping SDN domains as shown in Figure 2.

Figure 2: SDN Domain Structure



Reasons for using SDN domains include the following:

- *Scalability:* The number of devices an SDN controller can feasibly manage is limited. Thus, a reasonably large network may need to deploy multiple SDN controllers.
- *Privacy:* A carrier may choose to implement different privacy policies in different SDN domains. For example, an SDN domain may be dedicated to a set of customers who implement their own highly customized privacy policies, requiring that some networking information in this domain (for example, network topology) not be disclosed to an external entity.
- *Incremental deployment:* A carrier's network may consist of portions of traditional and newer infrastructure. Dividing the network into multiple, individually manageable SDN domains allows for flexible incremental deployment.

The existence of multiple domains creates a requirement for individual controllers to communicate with each other via a standardized protocol to exchange routing information. The IETF is currently working on developing a protocol, called *SDNi,* for "interfacing SDN Domain Controllers"[5]. SDNi functions include:

- Coordinate flow setup originated by applications containing information such as path requirement, QoS, and service-level agreements across multiple SDN domains.
- Exchange reachability information to facilitate inter-SDN routing. This information exchange will allow a single flow to traverse multiple SDNs and have each controller select the most appropriate path when multiple such paths are available.

The message types for SDNi tentatively include the following:

- Reachability update
- Flow setup/tear-down/update request (including application capability requirements such as QoS, data rate, latency etc.)
- Capability update (including network-related capabilities such as data rate and QoS, and system and software capabilities available inside the domain)

OpenFlow

To turn the concept of SND into practical implementation, two requirements must be met. First, there must be a common logical architecture in all switches, routers, and other network devices to be managed by an SDN controller. This logical architecture may be implemented in different ways on different vendor equipment and in different types of network devices, so long as the SDN controller sees a uniform logical switch function. Second, a standard, secure protocol is needed between the SDN controller and the network device.

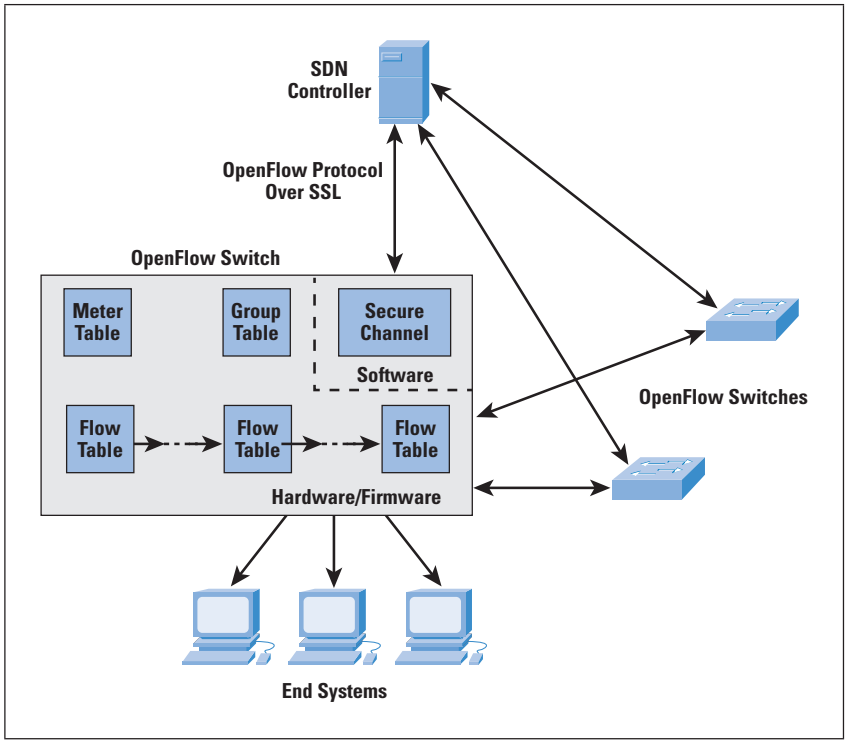
Both of these requirements are addressed by *OpenFlow,* which is both a protocol between SDN controllers and network devices, as well as a specification of the logical structure of the network switch functions[6, 7]. OpenFlow is defined in the *OpenFlow Switch Specification,* published by the *Open Networking Foundation* (ONF). ONF is a consortium of software providers, content delivery networks, and networking equipment vendors whose purpose is to promote software-defined networking.

This discussion is based on the current OpenFlow specification, Version 1.3.0, June 25, 2012[8]. The original specification, 1.0, was developed at Stanford University and was widely implemented. OpenFlow 1.2 was the first release from ONF after inheriting the project from Stanford. OpenFlow 1.3 significantly expands the functions of the specification. Version 1.3 is likely to become the stable base upon which future commercial implementations for OpenFlow will be built. ONF intends for this version to be a stable target for chip and software vendors, so little if any change is planned for the foreseeable future[9].

Logical Switch Architecture

Figure 3 illustrates the basic structure of the OpenFlow environment. An SDN controller communicates with OpenFlow-compatible switches using the OpenFlow protocol running over the *Secure Sockets Layer* (SSL). Each switch connects to other OpenFlow switches and, possibly, to end-user devices that are the sources and destinations of packet flows. Within each switch, a series of tables—typically implemented in hardware or firmware—are used to manage the flows of packets through the switch.

Figure 3: OpenFlow Switch



The OpenFlow specification defines three types of tables in the logical switch architecture. A *Flow Table* matches incoming packets to a particular flow and specifies the functions that are to be performed on the packets. There may be multiple flow tables that operate in a pipeline fashion, as explained subsequently. A flow table may direct a flow to a *Group Table*, which may trigger a variety of actions that affect one or more flows. A *Meter Table* can trigger a variety of performance-related actions on a flow.

Before proceeding, it is helpful to define what the term *flow* means. Curiously, this term is not defined in the OpenFlow specification, nor is there an attempt to define it in virtually all of the literature on OpenFlow. In general terms, a flow is a sequence of packets traversing a network that share a set of header field values. For example, a flow could consist of all packets with the same source and destination IP addresses, or all packets with the same VLAN identifier. We provide a more specific definition subsequently.

Flow-Table Components

The basic building block of the logical switch architecture is the flow table. Each packet that enters a switch passes through one or more flow tables. Each flow table contains entries consisting of six components:

- *Match Fields:* Used to select packets that match the values in the fields.
- *Priority:* Relative priority of table entries.
- *Counters:* Updated for matching packets. The OpenFlow specification defines a variety of timers. Examples include the number of received bytes and packets per port, per flow table, and per flow-table entry; number of dropped packets; and duration of a flow.
- *Instructions:* Actions to be taken if a match occurs.
- *Timeouts:* Maximum amount of idle time before a flow is expired by the switch.
- *Cookie:* Opaque data value chosen by the controller. May be used by the controller to filter flow statistics, flow modification, and flow deletion; not used when processing packets.

A flow table may include a *table-miss* flow entry, which renders all Match Fields wildcards (every field is a match regardless of value) and has the lowest priority (priority 0). The Match Fields component of a table entry consists of the following required fields:

- *Ingress Port:* The identifier of the port on the switch where the packet arrived. It may be a physical port or a switch-defined virtual port.
- *Ethernet Source and Destination Addresses:* Each entry can be an exact address, a bitmasked value for which only some of the address bits are checked, or a wildcard value (match any value).

- *IPv4 or IPv6 Protocol Number:* A protocol number value, indicating the next header in the packet.
- *IPv4 or IPv6 Source Address and Destination Address:* Each entry can be an exact address, a bitmasked value, a subnet mask value, or a wildcard value.
- *TCP Source and Destination Ports:* Exact match or wildcard value.
- *User Datagram Protocol (UDP) Source and Destination Ports:* Exact match or wildcard value.

The preceding match fields must be supported by any OpenFlow-compliant switch. The following fields may be optionally supported:

- *Physical Port:* Used to designate underlying physical port when packet is received on a logical port.
- *Metadata:* Additional information that can be passed from one table to another during the processing of a packet. Its use is discussed subsequently.
- *Ethernet Type:* Ethernet Type field.
- *VLAN ID and VLAN User Priority:* Fields in the IEEE 802.1Q Virtual LAN header.
- *IPv4 or IPv6 DS and ECN:* Differentiated Services and Explicit Congestion Notification fields.
- *Stream Control Transmission Protocol (SCTP) Source and Destination Ports:* Exact match or wildcard value.
- *Internet Control Message Protocol (ICMP) Type and Code Fields:* Exact match or wildcard value.
- *Address Resolution Protocol (ARP) Opcode:* Exact match in Ethernet Type field.
- *Source and Target IPv4 Addresses in Address Resolution Protocol (ARP) Payload:* Can be an exact address, a bitmasked value, a subnet mask value, or a wildcard value.
- *IPv6 Flow Label:* Exact match or wildcard.
- *ICMPv6 Type and Code fields:* Exact match or wildcard value.
- *IPv6 Neighbor Discovery Target Address:* In an IPv6 Neighbor Discovery message.
- *IPv6 Neighbor Discovery Source and Target Addresses:* Link-layer address options in an IPv6 Neighbor Discovery message.
- *Multiprotocol Label Switching (MPLS) Label Value, Traffic Class, and Bottom of Stack (BoS):* Fields in the top label of an MPLS label stack.

Thus, OpenFlow can be used with network traffic involving a variety of protocols and network services. Note that at the MAC/link layer, only Ethernet is supported. Thus, OpenFlow as currently defined cannot control Layer 2 traffic over wireless networks.

We can now offer a more precise definition of the term *flow.* From the point of view of an individual switch, a flow is a sequence of packets that matches a specific entry in a flow table. The definition is packet-oriented, in the sense that it is a function of the values of header fields of the packets that constitute the flow, and not a function of the path they follow through the network. A combination of flow entries on multiple switches defines a flow that is bound to a specific path.

The *instructions component* of a table entry consists of a set of instructions that are executed if the packet matches the entry. Before describing the types of instructions, we need to define the terms "Action" and "Action Set." Actions describe packet forwarding, packet modification, and group table processing operations. The OpenFlow specification includes the following actions:

- *Output:* Forward packet to specified port.
- *Set-Queue:* Sets the queue ID for a packet. When the packet is forwarded to a port using the output action, the queue id determines which queue attached to this port is used for scheduling and forwarding the packet. Forwarding behavior is dictated by the configuration of the queue and is used to provide basic QoS support.
- *Group:* Process packet through specified group.
- *Push-Tag/Pop-Tag:* Push or pop a tag field for a VLAN or MPLS packet.
- *Set-Field:* The various Set-Field actions are identified by their field type; they modify the values of respective header fields in the packet.
- *Change-TTL:* The various Change-TTL actions modify the values of the IPv4 Time To Live (TTL), IPv6 Hop Limit, or MPLS TTL in the packet.

An *Action Set* is a list of actions associated with a packet that are accumulated while the packet is processed by each table and executed when the packet exits the processing pipeline. Instructions are of four types:

- *Direct packet through pipeline:* The Goto-Table instruction directs the packet to a table farther along in the pipeline. The Meter instruction directs the packet to a specified meter.
- *Perform action on packet:* Actions may be performed on the packet when it is matched to a table entry.
- *Update action set:* Merge specified actions into the current action set for this packet on this flow, or clear all the actions in the action set.
- *Update metadata:* A metadata value can be associated with a packet. It is used to carry information from one table to the next.

Flow-Table Pipeline

A switch includes one or more flow tables. If there is more than one flow table, they are organized as a pipeline as shown in Figure 4, with the tables labeled with increasing numbers starting with 0.

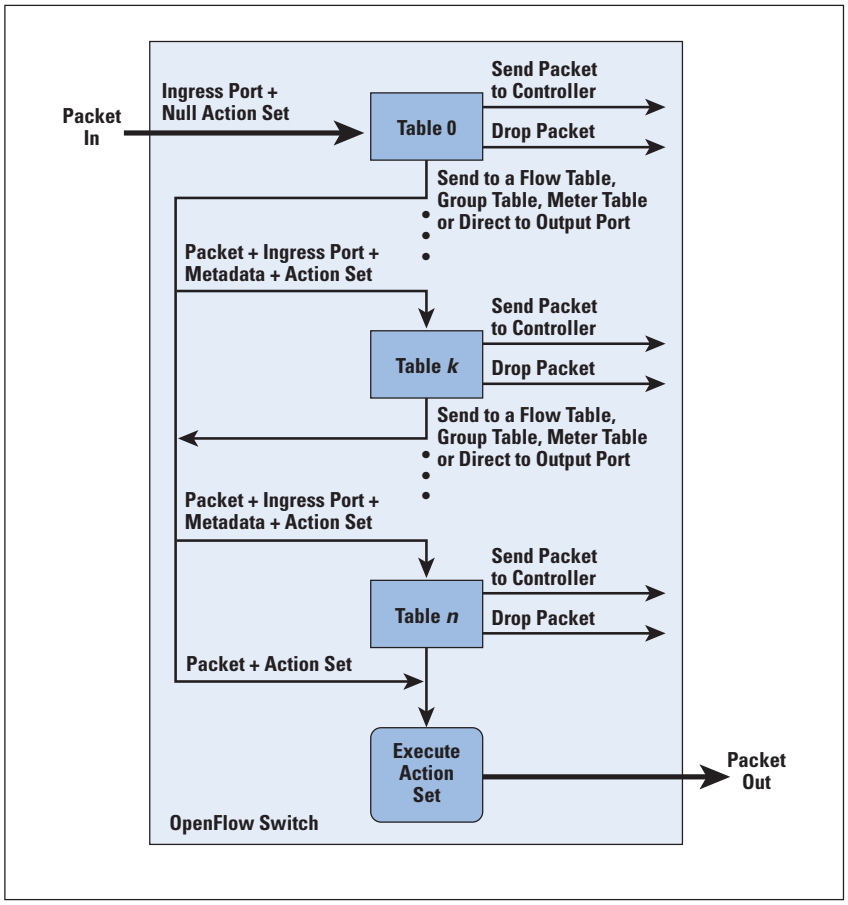


Figure 4: Packet Flow Through OpenFlow-Compliant Switch

When a packet is presented to a table for matching, the input consists of the packet, the identity of the ingress port, the associated metadata value, and the associated action set. For Table 0, the metadata value is blank and the action set is null. Processing proceeds as follows:

1. Find the highest-priority matching flow entry. If there is no match on any entry and there is no table-miss entry, then the packet is dropped. If there is a match only on a table-miss entry, then that entry specifies one of three actions:

   a. Send packet to controller. This action will enable the controller to define a new flow for this and similar packets, or decide to drop the packet.

   b. Direct packet to another flow table farther down the pipeline.

   c. Drop the packet.

2. If there is a match on one or more entries other than the table-miss entry, then the match is defined to be with the highest-priority matching entry. The following actions may then be performed:
   a. Update any counters associated with this entry.
   b. Execute any instructions associated with this entry. These instructions may include updating the action set, updating the metadata value, and performing actions.
   c. The packet is then forwarded to a flow table further down the pipeline, to the group table, or to the meter table, or it could be directed to an output port.

For the final table in the pipeline, forwarding to another flow table is not an option.

If and when a packet is finally directed to an output port, the accumulated action set is executed and then the packet is queued for output.

OpenFlow Protocol

The OpenFlow protocol describes message exchanges that take place between an OpenFlow controller and an OpenFlow switch. Typically, the protocol is implemented on top of SSL or *Transport Layer Security* (TLS), providing a secure OpenFlow channel.

The OpenFlow protocol enables the controller to perform add, update, and delete actions to the flow entries in the flow tables. It supports three types of messages, as shown in Table 1.

- *Controller-to-Switch:* These messages are initiated by the controller and, in some cases, require a response from the switch. This class of messages enables the controller to manage the logical state of the switch, including its configuration and details of flow- and group-table entries. Also included in this class is the Packet-out message. This message is used when a switch sends a packet to the controller and the controller decides not to drop the packet but to direct it to a switch output port.
- *Asynchronous:* These types of messages are sent without solicitation from the controller. This class includes various status messages to the controller. Also included is the Packet-in message, which may be used by the switch to send a packet to the controller when there is no flow-table match.
- *Symmetric:* These messages are sent without solicitation from either the controller or the switch. They are simple yet helpful. Hello messages are typically sent back and forth between the controller and switch when the connection is first established. Echo request and reply messages can be used by either the switch or controller to measure the latency or bandwidth of a controller-switch connection or just verify that the device is operating. The Experimenter message is used to stage features to be built into future versions of OpenFlow.

Table 1: OpenFlow Messages

Message	Description
Controller-to-Switch	
Features	Request the capabilities of a switch. Switch responds with a features reply that specifies its capabilities.
Configuration	Set and query configuration parameters. Switch responds with parameter settings.
Modify-State	Add, delete, and modify flow/group entries and set switch port properties.
Read-State	Collect information from switch, such as current configuration, statistics, and capabilities.
Packet-out	Direct packet to a specified port on the switch.
Barrier	Barrier request/reply messages are used by the controller to ensure message dependencies have been met or to receive notifications for completed operations.
Role-Request	Set or query role of the OpenFlow channel. Useful when switch connects to multiple controllers.
Asynchronous-Configuration	Set filter on asynchronous messages or query that filter. Useful when switch connects to multiple controllers.
Asynchronous	
Packet-in	Transfer packet to controller.
Flow-Removed	Inform the controller about the removal of a flow entry from a flow table.
Port-Status	Inform the controller of a change on a port.
Error	Notify controller of error or problem condition.
Symmetric	
Hello	Exchanged between the switch and controller upon connection startup.
Echo	Echo request/reply messages can be sent from either the switch or the controller, and they must return an echo reply.
Experimenter	For additional functions.

The OpenFlow protocol enables the controller to manage the logical structure of a switch, without regard to the details of how the switch implements the OpenFlow logical architecture.

Summary

SDNs, implemented using OpenFlow, provide a powerful, vendor-independent approach to managing complex networks with dynamic demands. The software-defined network can continue to use many of the useful network technologies already in place, such as virtual LANs and an MPLS infrastructure. SDNs and OpenFlow are likely to become commonplace in large carrier networks, cloud infrastructures, and other networks that support the use of big data.

References

[1] Greg Goth, "Software-Defined Networking Could Shake Up More than Packets," *IEEE Internet Computing,* July/August, 2011.

[2] Robin Layland,"The Dark Side of Server Virtualization," *Network World,* July 7, 2010.

[3] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," ONF White Paper, April 12, 2012.

[4] Dell, Inc., "Software Defined Networking: A Dell Point of View," Dell White Paper, October 2012.

[5] Hongtao Yin, Haiyong Xie, Tina Tsou, Diego Lopez, Pedro Aranda, and Ron Sidi, "SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains," Internet Draft, work in progress, June 2012, **draft-yin-sdn-sdni-00.txt**

[6] Steven Vaughan-Nichols, "OpenFlow: The Next Generation of the Network?" *Computer,* August 2011.

[7] Thomas A. Limoncelli, "OpenFlow: A Radical New Idea in Networking," *Communications of the ACM,* August 2012.

[8] Open Networking Foundation, "OpenFlow Switch Specification Version 1.3.0," June 25, 2012.

[9] Sean Michael Kerner, "OpenFlow Protocol 1.3.0 Approved," *Enterprise Networking Planet,* May 17, 2012.

WILLIAM STALLINGS is an independent consultant and author of many books on security, computer networking, and computer architecture. His latest book is *Data and Computer Communications* (Pearson, 2013). He maintains a computer science resource site for computer science students and professionals at **ComputerScienceStudent.com**. He has a Ph.D. in computer science from M.I.T. He can be reached at **ws@shore.net**

IPv4 and IPv6 Address Authentication

by Scott Hogg, GTRI

Some Internet services use the source address of the client's computer as a form of authentication. These systems keep track of the *Internet Protocol* (IP) address that an end user used the last time that user accessed the site and try to determine if the user is legitimate. When that same user accesses the site from a different source IP address, the site asks for further authentication to revalidate the client's computer. The theory is that a user's typical location computer has a somewhat persistent IP address, but when the user has a new address, that user may be mobile or using a less secure wireless media, and then require further authentication. For example, many organizations have firewall policies with objects named like "Bob's Laptop" with the single IP address of his computer. This technique is used by some banking sites, some online gaming sites, and Gmail (for example, *Google Authenticator*)[1].

Some online retailers track the client IP address for Business Intelligence or fraud detection and forensics purposes. The retailer tracks the client IP address using the source address to analyze fraudulent purchases and to track down criminal activity. Some industries frequently use the customer's IP address as a form of authentication. Also, many sites that use *Server Load Balancers* (SLBs) and *Application Delivery Controllers* (ADCs) use *X-forwarded-for* (XFF)[2] or *Hypertext Transfer Protocol* (HTTP) header insertion so that the back-end real servers are aware of the client's original IP address associated with the reverse proxy connection. The application can then use the IP address for tracking purposes or simply log the address with the transaction details.

Other applications try to validate the client's source IP address when the server receives an inbound connection. E-mail *Simple Mail Transfer Protocol* (SMTP) servers or *Internet Relay Chat* (IRC) servers can use the *Ident* protocol[3] to try to validate the originating e-mail server or client computer validity. SMTP e-mail servers[4] also use other protocols such as *SenderID*[5], *Sender Policy Framework* (SPF)[6], and *DomainKeys Identified Mail* (DKIM)[7] in an effort to restrict spam. *Domain Name System* (DNS) *pointer* (PTR) records are sometimes used as a way to confirm that the client IP address is configured in DNS (for example, forward-confirmed reverse DNS[8]).

Statically configured IP addresses are frequently used to signify some limited form of authentication. These addresses may not be used to authenticate a user, but authenticate IT systems to each other. Many manually configured systems rely on IP address to permit connectivity, including manually configured tunnels, *IP Security* (IPsec) peers, Apache `.htaccess`[9], `.rhosts`[10], SAMBA, and *Border Gateway Protocol* (BGP) peers, among many others.

The address is used as one part of the connection authentication. Obviously, IPsec connections are authenticated with certificates or preshared keys to strengthen their validation of the endpoints. Similarly, BGP peers use passwords (and/or *Time To Live* [TTL][11]) to help secure the peer beyond just IP address confirmation.

Identity-based firewalls police users' network behavior by IP address through *Windows Active Directory, Remote Authentication Dial-In User Service* (RADIUS), or *Lightweight Directory Access Protocol* (LDAP). Palo Alto firewalls championed the *UserID* concept as part of their analysis of connections to permit or deny authentication[12]. The Cisco *Adaptive Security Appliance* (ASA) firewalls running Version 8.4 or later can be configured for Identify firewall functions[13]. Firewalls have always used manually configured IP addresses as the fundamental element of their policies. The IP address is used in the policy as if that concretely defines a system and/or user. This process of adding rules based on IP address continues until the firewall is a pincushion full of pinholes.

Organizations that rely on using an IP address as a form of authentication run the risk of an attacker learning that IP address and attacking using that address. Attackers who know the addresses that are being used could perform a *Man-in-the-Middle* (MITM) attack or use TCP session hijacking. The attacker needs to know only the information about which IP addresses are used for the communications. The attacker might be able to ascertain the IP addresses the organization uses by guessing or by other means. The attacker could find the external IP address of the company's firewall and assume that IPv4 *Network Address Translation* (NAT)[24] was being performed. The attacker could also suppose the business partner IP address. Organizations that use these techniques are relying on the secrecy of their IP addressing for the purposes of security.

Address Quality

The quality of the IP address is an important concept to consider. For example, a global address is of higher surety and authenticity than a private address. Many organizations use private addresses and overlap between private networks, whereas global addresses are unique and they are registered to a specific entity. Public addresses can reveal the client's *Internet Service Provider* (ISP), the organization that has registered the IP addresses, and some geolocation information. However, any IP packet can be spoofed and the source-address modified or crafted. Of course, if the source IP addresses is spoofed, the return packets will not necessarily be sent back to the attacker's source in these cases, but one-way blind attacks are still possible. Furthermore, systems such as *Tor*[14] are intended to protect the identity of the end user.

Using the IP address as a form of authentication does not work if the client changes its location frequently. Today, many clients use mobile devices that can change their Layer 3 addresses often. The source IP address of the mobile device could change frequently and could even change during the transaction.

With increasing mobile device usage for business purposes, the ability to determine the typical IP address of the client becomes impossible. Increased scarcity of IPv4 addresses is leading service providers to use *Carrier-Grade NAT* (CGN) or *Large-Scale NAT* (LSN) and shorter and shorter *Dynamic Host Configuration Protocol* (DHCP) lease times, meaning that the client IP address is not static.

Many organizations and systems assume that a single computer with a single IP address represents a single user. The problem arises where IPv4 public addresses may not uniquely identify a single user. The industry may be trying to anticipate the implications of CGN/LSN and the effect of systems that rely on the uniqueness of a public IP address. Similar problems related to the mega-proxies of the late 1990s occurred (for example, AOL). With CGN/LSN systems in place, online retailers and banks will no longer be able to use the client IPv4 as the "real client IP." Instead, the IP address observed on the retailer's web servers will come from a pool of IPv4 addresses configured in the LSN system. In this situation, one bad actor could spoil that NAT pool IPv4 address for subsequent lawful users who follow. When a legitimate user tries to make an online purchase and that user's system happens to use that IPv4 address of the bad actor, then the purchase attempt might be blocked. This situation would be bad for business on Cyber-Monday, or any day for that matter.

Table 1 compares IPv4 and IPv6 for their authentication purposes.

Table 1: IPv4 vs. IPv6 for Authentication

IPv4	IPv6
Extensive use of NAT	No motivation for NAT
End users use private addresses	End users use global addresses
Use of CGN/LSN starting	Abundance of IPv6 addresses
Robust geolocation	Geolocation needs improvement
Addresses could be spoofed	Addresses could be spoofed

Public Addresses

Public IPv4 addresses are becoming increasingly scarce[15, 25], however, an abundance of global IPv6 addresses are available[16]. Global IPv6 addresses can be obtained from *Regional Internet Registries* (RIRs) or from an IPv6-capable service provider. Residential broadband Internet users today use private IPv4 addresses on their internal computers, but these computers will soon start to use global IPv6 addresses as they upgrade to IPv6-capable *Customer Premises Equipment* (CPE). IPv6-enabled residential subscribers and employees of IPv6-enabled enterprises will be using global addresses when they access an IPv6-capable Internet service.

To online retailers, this situation may represent a change to their IP address authentication measures. As IPv4 residential users start to go through CGN/LSN systems, their IPv4 addresses will be useless for authentication.

However, their IPv6 addresses will be global addresses with no NAT taking place between the client and the server[17]. It will be seemingly more accurate to use the IPv6 address to determine the validity of the source. IPv6 could potentially help to create an environment with more "trustworthiness" and less anonymity. For example, IPv6 IPsec connections could use the *Authentication Header* (AH) and *Encapsulating Security Payload* (ESP) together to create stronger connections, where IPv4 IPsec connections rely on NAT-Traversal and can use only ESP[18].

As we head toward an increasingly dual-stack world, applications will need to do "dual-checking" of both the client's IPv4 and IPv6 addresses. In a dual-stack world, there is more work to do[19], and servers using IP address authentication will need to understand that a single user will have both an IPv4 address and an IPv6 address and keep track of both. The other consideration is that IPv6 nodes may have multiple global IPv6 addresses in some situations.

Authentication with Addresses

Security experts know that the secrecy of the encryption algorithm is not important, but the secrecy of the key is vitally important (Kerckhoffs's Principle[20]). The same concept should hold true for an IP address. Users should not rely on the secrecy of their IP addresses to be secure; the security of the individual node should be strong enough to defend against attacks. To the extreme, users should feel confident enough in their security posture that they feel comfortable widely publicizing their IP address. However, even if you are using *LifeLock*[21], you should still keep your Social Security Number or government ID number private.

Security practitioners know that authentication should involve multiple factors. A combination of "something you are" (biometrics), "something you know" (username/password) and "something you have" (token, *Common Access Card*[22]) forms a more solid foundation for identifying a user. Combining two factors provides more assurance than just one factor. We are all aware of the weaknesses of using username and password as a means of authentication[23].

The systems mentioned so far in this article are three-factor systems (username, password, and IP address) which are presumably better than just username/password. However, we should acknowledge that an IP address is not a characteristic of a person. IP addresses have more to do with "somewhere you are," because the IP address reflects location within a network topology by the prefix/subnet. The last few bits of an IPv4 address representing the point-of-attachment or an IPv6 *Interface Identifier* (IID) do not necessarily uniquely identify a user. Having authentication based on your location becomes difficult with mobile devices that roam widely. However, controlling authentication to users who are within the office subnet rather than outside the office may be useful.

An IP address is not something anyone really owns outright. Few organizations actually have complete ownership of their IP addresses. Organizations should read the fine print in the policies of their RIR. Organizations just pay RIR annual fees for their addresses, but if they stop paying those dues, the IP address allocation is revoked and the addresses go back into a pool for reallocation to another organization. Therefore, public IP addresses do not truly represent unequivocal ownership or legitimacy of a network.

Conclusion

Many different types of systems use the client's source address as a form of authentication. Systems that rely on IP address checking will need to do so for IPv4 and will need to be modified to use IPv6 addresses. IPv6 systems will use global addresses without NAT, so the security systems must stand on their own even though the IPv6 address is publicized. IPv4 and IPv6 addresses can be spoofed, and as CGN/LSN systems become widely deployed the validity of a public IPv4 address decreases. However, IPv6 addresses are not necessarily any more trustworthy than IPv4 addresses when used for authentication. Regardless, the IP address should not be the only factor used for authentication, and we should not be using IPv4 or IPv6 addresses as a form of authentication. The truth is that the IT industry needs to be aware of where IP addresses are used as a form of authentication and seek out better forms of authentication beyond just username, password, and IP address.

References

[1] Google Authenticator, **http://support.google.com/a/bin/answer.py?hl=en&answer=1037451**

[2] **http://en.wikipedia.org/wiki/X-Forwarded-For**

[3] Mike St. Johns, "Identification Protocol," RFC 1413, February 1993.

[4] **http://en.wikipedia.org/wiki/Email_authentication**

[5] Meng Weng Wong and Jim Lyon, "Sender ID: Authenticating E-Mail," RFC 4406, April 2006.

[6] Wayne Schlitt and Meng Weng Wong, "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1," RFC 4408, April 2006.

[7] Miles Libbey, Michael Thomas, and Mark Delany, "DomainKeys Identified Mail (DKIM) Signatures," RFC 4871, May 2007.

[8] **http://en.wikipedia.org/wiki/Forward-confirmed_reverse_DNS**

[9] **http://en.wikipedia.org/wiki/Htaccess**

[10] **http://en.wikipedia.org/wiki/Rlogin**

[11] Vijay Gill, John Heasley, and David Meyer, "The Generalized TTL Security Mechanism (GTSM)," RFC 3682, February 2004.

[12] Palo Alto Networks, UserID, **http://www.paloaltonetworks.com/products/technologies/user-id.html**

[13] Cisco ASA firmware 8.4 Identify Firewall, **http://www.cisco.com/en/US/docs/security/asa/asa84/configuration/guide/access_idfw.html**

[14] **http://en.wikipedia.org/wiki/Tor_(anonymity_network)**

[15] **http://en.wikipedia.org/wiki/IPv4_address_depletion**

[16] **http://en.wikipedia.org/wiki/IPv6**

[17] Scott Hogg and Owen DeLong, "IPv6 NAT - You can get it, but you may not need or want it," Infoblox Blog, October 2, 2012. **http://www.infoblox.com/community/blog/ipv6-nat-you-can-get-it-you-may-not-need-or-want-it**

[18] **http://en.wikipedia.org/wiki/Ipsec**

[19] Scott Hogg, "Dual-Stack Will Increase Operating Expenses," *Network World,* July 31, 2012, **http://www.networkworld.com/community/blog/dual-stack-will-increase-operating-expenses**

[20] **http://en.wikipedia.org/wiki/Kerckhoffs%27s_Principle**

[21] **http://en.wikipedia.org/wiki/LifeLock**

[22] Common Access Card (CAC), **http://www.cac.mil/**

[23] Mat Honan, "Kill the P@55W0rD," *WIRED Magazine,* December 2012, **http://www.wired.com/gadgetlab/2012/11/ff-mat-honan-password-hacker/all/**

[24] Geoff Huston, "Anatomy: A Look inside Network Address Translators," *The Internet Protocol Journal,* Volume 7, No. 3, September 2004.

[25] Several articles on IPv4 Exhaustion and IPv6 Transition in *The Internet Protocol Journal,* Volume 14, No. 1, March 2011.

SCOTT HOGG is the Director of Technology Solutions at GTRI in Denver Colorado. He holds a B.S. in Computer Science from Colorado State University, a M.S. in Telecommunications from the University of Colorado, and CCIE® #5133 and CISSP #4610 certifications. Hogg is active in the IPv6 community, Chair Emeritus of the RMv6TF, author of *IPv6 Security* (Cisco Press), a member of the Infoblox IPv6 Center of Excellence, a frequent presenter, and a *Network World* blogger. He can be reached at **scott@hoggnet.com** or followed on twitter **@scotthogg**

Summary Report of the ITU-T World Conference on International Telecommunications

by Robert Pepper and Chip Sharp, Cisco Systems

From 3–14 December, 2012, 151 Member States of the *International Telecommunication Union* (ITU) met in Dubai[0] at the *World Conference on International Telecommunications* (WCIT-12)[1] to revise the *International Telecommunication Regulations* (ITRs), a treaty-level document establishing policies governing international telecommunications services. During the 2-week conference the delegates debated several proposed changes on topics such as international mobile roaming, numbering, naming, addressing, fraud, the Internet, *Quality of Service* (QoS), etc. In the end, a revised version of the treaty was finalized[2], but only 89 of the 151 Member States attending signed it.

There have been many articles discussing different aspects of the conference and its outcomes. This article provides background on the ITRs and focuses on the potential impact of the WCIT and its revised treaty on development of the Internet.

Background

The ITRs originated from the development of international telegraphy in Europe in the late 1800s and the need for a treaty defining how the government-operated national telegraph networks would interconnect and interoperate[3]. As telephony and radio communications were invented, new treaties were developed to regulate their international operation. Up until the 1980s most telephone and telegraph companies were government-owned monopolies with some government licensed private companies operating as a monopoly. In 1988, the separate telegraph and telephone treaties were merged into the International Telecommunications Regulations while the *Radio Regulations* remained a separate treaty. By 1988, though some liberalization and privatization had started in a few countries in some regions, most international telecommunications services globally were still provided by monopoly, government-owned carriers, and services were dominated by voice rather than data. International Internet connectivity and traffic were practically nonexistent in most countries. Of course, international data traffic (including Internet) was growing in importance to some countries such as the United States and some large multinational companies such as IBM (which wanted to provide international *Virtual Private Networks* [VPNs]).

One important aspect of the ITRs in 1988 was the telephony accounting rate system. Briefly, this system consisted of a calling-party-pays business model for telephony in which the originating country pays the terminating country settlements based on a bilaterally agreed-upon accounting rate. Because developed countries tended to make more calls to developing countries than conversely and the accounting rate tended to be substantially above cost in many cases, the accounting rate system effectively became a subsidy program and a source for hard currency for developing countries.

Since 1988 market liberalization, reduced regulation, increased competition, and the rise of the Internet and mobile wireless industries have drastically changed the global communications landscape. In 1997, the U.S. *Federal Communications Commission* (FCC) opted out of the accounting rate system defined in the ITRs[4], with many countries subsequently following suit. Voice over the Internet, arbitrage, hubbing, and other factors have reduced the telephony settlements revenue for developing countries. The 1988 ITRs[5] allowed for special arrangements between network operators outside the rules of the ITRs. These special arrangements allowed for the international physical connectivity on which growth of the international Internet depended.

As the Internet grew and the telecom market changed, there was increased pressure from some countries to revise the ITRs. Contributions submitted in the preparatory meetings for WCIT-12 reflected widely varying views on the nature and extent of possible changes to the ITRs to account for this greatly changed environment. Although some countries believed that the ITRs should set forth high-level strategic and policy principles that could adapt to further changes in the market, others proposed the inclusion of expanded regulatory provisions of a detailed and specific nature in the ITRs to address a wide range of new concerns and services, including the Internet, or even to include the intergovernmental regulation of content (for example, spam and information security).

High-Level Take-Aways

Out of 151 countries attending the conference, the treaty was signed by 89 countries, consisting of mostly emerging countries led by Russia, China, Brazil, and the Arab States; 55 countries, including the United States, Japan, Australia, Canada, United Kingdom, and most of Europe, did not sign at the time. Countries that did not sign the treaty in Dubai can accede to the treaty after the WCIT by notifying the Secretary-General of the ITU. It is quite likely that some countries that did not sign the treaty will accede to it over the next few years.

The treaty takes effect on January 1, 2015 (after the 2014 *Plenipotentiary Conference*). Each signing country has to go through its national process for approval (for example, ratification) before the treaty takes effect for that country.

Although there has been a lot of negative commentary on the WCIT in the Internet community, in the end there are some important positive results for the Internet:

- No provisions were added to treaty text explicitly concerning the Internet, Internet Governance, or information security.
- No provisions were added to the treaty text concerning naming or addressing.
- No provisions modifying the basic business models of the Internet or mandating QoS on the Internet were made.

- The updated treaty explicitly recognizes commercial arrangements in addition to the old accounting rate regime for telecommunications.
- Article 9 on *Special Arrangements* allowing for telecommunications arrangements outside the treaty was retained mostly unchanged, thus allowing such special arrangements to continue to be used even between nonsignatory and signatory countries.
- A new resolution on landlocked countries could encourage access of such countries to landing stations in other countries and ease landlocked countries' ability to acquire international connectivity.

Some results that could be of concern to the Internet follow:

- The term identifying the operators to which the treaty applies ("authorized operating agencies") was modified. The supporters of the new term claim it does not expand the scope of the treaty, but it will bear watching.
- A provision on "unsolicited bulk electronic communications," developed after a long debate on spam, could lead governments to regulate and filter e-mail in addition to having unintended consequences such as disallowing bulk electronic emergency warning systems.
- Numbering provisions and requirements to deliver *Calling Party Number* were intended by some countries to allow for restrictions on international *Voice over IP* (VoIP) and VoIP services (including VoIP over the Internet).
- A new provision on network security could encourage more multilateral discussions in an intergovernmental setting (as opposed to multistakeholder).
- A new Resolution 3 on the Internet instructs the Secretary-General to engage further in Internet Governance discussions and further supports intergovernmental Internet policy processes.
- A new Resolution 5 mentions the transition to IP-based networks. It originally was aimed at over-the-top providers, but was modified to apply to service providers of international services. The end result is rather ambiguous in many respects and will bear watching.
- A new Article was added concerning telecommunication exchange points. Although the Internet is not mentioned explicitly, the originators of this article intended for it to apply to Internet Exchange Points. This Article could be used to support development of an enabling environment for regional telecommunication connectivity, but could also be used to justify regulation of Internet Exchange Points.
- Resolution Plen/4 requires PP'14 to consider a review of the ITRs every 8 years. This provision could result in another WCIT in 2020.

Table 1 lists the Member States that signed and did not sign the treaty in Dubai[6].

Table 1: Treaty Signatories and Nonsignatories

Signatories			Nonsignatories	
Afghanistan	Guatemala	Qatar	Albania	Latvia
Algeria	Guyana	Russia	Andorra	Lichtenstein
Angola	Haiti	Rwanda	Armenia	Lithuania
Argentina	Indonesia	Saint Lucia	Australia	Luxembourg
Azerbaijan	Iran	Saudi Arabia	Austria	Malawi
Bahrain	Iraq	Senegal	Belarus	Malta
Bangladesh	Jamaica	Sierra Leone	Belgium	Marshall Islands
Barbados	Jordan	Singapore	Bulgaria	Moldova
Belize	Kazakhstan	Somalia	Canada	Mongolia
Benin	Korea (Rep. of)	South Africa	Chile	Montenegro
Bhutan	Kuwait	South Sudan	Colombia	Netherlands
Botswana	Kyrgyzstan	Sri Lanka	Costa Rica	New Zealand
Brazil	Lebanon	Sudan	Croatia	Norway
Brunei	Lesotho	Swaziland	Cyprus	Philippines
Burkina Faso	Liberia	Tanzania	Czech Republic	Poland
Burundi	Libya	Thailand	Denmark	Peru
Cambodia	Malaysia	Togo	Estonia	Portugal
Cape Verde	Mali	Trinidad and Tobago	Finland	Serbia
Central African Rep.	Mauritius	Tunisia	France	Slovak Republic
China	Mexico	Turkey	Gambia	Slovenia
Comoros	Morocco	Uganda	Georgia	Spain
Congo	Mozambique	Ukraine	Germany	Sweden
Cote d'Ivoire	Namibia	UAE	Greece	Switzerland
Cuba	Nepal	Uruguay	Hungary	United Kingdom
Djibouti	Niger	Uzbekistan	India	United States
Dominican Rep.	Nigeria	Venezuela	Ireland	
Egypt	Oman	Vietnam	Israel	
El Salvador	Panama	Yemen	Italy	
Gabon	Papua New Guinea	Zimbabwe	Japan	
Ghana	Paraguay		Kenya	

Note: Other *United Nations* (UN) member states were not eligible to sign or did not attend the conference but might still accede to the treaty: Antigua and Barbuda, Bahamas, Bolivia, Bosnia and Herzegovina, Cameroon, Chad, Dem. People's Republic of Korea, Dem. Rep. of the Congo, Dominica, Ecuador, Equatorial Guinea, Eritrea, Ethiopia, Fiji, Grenada, Guinea, Guinea-Bissau, Honduras, Iceland, Kiribati, Lao P.D.R., T.F.Y.R. Macedonia, Madagascar, Maldives, Mauritania, Micronesia, Monaco, Myanmar, Nauru, Nicaragua, Pakistan, Romania, Saint Kitts and Nevis, Saint Vincent and the Grenadines, Samoa, San Marino, Sao Tome and Principe, Seychelles, Solomon Islands, Suriname, Syria, Tajikistan, Timor-Leste, Tonga, Turkmenistan, Tuvalu, Vanuatu, the Vatican, and Zambia.

Proposals and Outcomes

When the conference began there were several provisions that either explicitly or implicitly applied to the Internet, including:

- A proposal to define the term "Internet" and explicitly bring the Internet into the regulatory structure of the treaty
- Proposals to bring Internet naming, addressing, and identifiers into the treaty
- A proposal to include a provision on access to Internet websites
- A proposal on "traffic exchange points" that was intended to apply to *Internet Exchange Points*
- Proposals from multiple states on spam, information security, and *cybersecurity*

Although the Secretary-General of the ITU declared that the WCIT was not about the Internet or Internet Governance[7], by rule, the WCIT had to consider input from its Member States. Given that Member States submitted proposals on the Internet, the Internet and Internet Governance was a substantive topic of discussion.

The following sections provide a brief review of some of the more difficult discussions related to the Internet.

Security

There were several proposals[8] going into the WCIT to include cybersecurity, including information security, in the new ITRs. These proposals generated significant discussions and negotiations during the conference. The final text (Article 5A) is a great improvement over the proposals into the conference in that it focuses on the security and robustness of networks and prevention of technical harm to networks, with no mention of information security or cybersecurity.

The new provision mentions that Member States shall "collectively endeavour," a provision that could engender more multilateral discussions in an intergovernmental setting (for example, ITU).

Organizations to Which the Treaty Applies

The 1988 ITR treaty focused on licensed carriers and government-owned *Post, Telephone, and Telegraph* (PTT) entities. Proposals[8] into the WCIT would have applied the treaty to a wider range of organizations and companies. In the end, the treaty developed a new term, *Authorized Operating Agencies* (AOA). The proponents of this new term argued that it does not broaden the scope of the ITRs in terms of the organizations to which it applies. This interpretation of the new term should be supported, but monitored.

Internet-Specific Proposals and Resolutions (Resolutions Plen/3 and Plen/5)

Proposals[8] were submitted to the WCIT to define the term "Internet" and to encode into the treaty the right of countries to regulate the "national segment" of the Internet. At the end of the first week of WCIT, Algeria, Saudi Arabia, Bahrain, China, United Arab Emirates, Iraq, Sudan, and Russia announced development of a new draft set of Resolutions that contained provisions that Member States shall have the right to manage the Internet, including Internet numbering, naming, addressing, and identification resources.

Although the United States, United Kingdom, and others were successful in removing any mention of the Internet from the treaty text, Internet-related language was moved into a nonbinding resolution *(Resolution Plen/3)* proposed by Russia "to foster an enabling environment for the greater growth of the Internet." Resolution Plen/3 instructs the ITU Secretary-General "to continue to take the necessary steps for ITU to play an active and constructive role in the development of broadband and the multistakeholder model of the Internet as expressed in § 35 of the *Tunis Agenda*." It also invites Member States to elaborate their positions on Internet-related concerns in the relevant ITU-related fora (something they could have done anyway).

This does not look too bad until one reads Paragraph 35 of the Tunis Agenda[9]. This paragraph lays out the roles of each type of stakeholder (private industry, civil society, *Intergovernmental Organizations* [IGOs], governments, etc.). It reserves an explicit role in "Internet-related public policy issues" for governments and intergovernmental organizations. It does not provide for any role in this area for the private sector or civil society. So although the Resolution seems to support the multistakeholder model of the Internet, it really restricts the roles of several of the main stakeholders.

Several countries pushed for inclusion of Paragraph 55 of the Tunis Agenda, recognizing that the existing arrangements have worked effectively, to balance the inclusion of Paragraph 35, but it was not included in the final Resolution.

Resolution Plen/3 may be used by some governments to reinforce the ITU's role in Internet Governance, including at future ITU conferences in 2013 and 2014.

On the other hand, the Resolution also instructs the Secretary-General "to support the participation of Member States and *all* other stakeholders, as applicable, in the activities of ITU in this regard." This statement supports participation of all stakeholders in the activities of the ITU, not restricted just to ITU Members, or in the case of ITU Council or some Council Working Groups just to Member States.

In signing the Final Acts, Russia added a Declaration/Reservation that it views the Internet as a new global telecommunication infrastructure and reserves the right to implement public policy, including international policy, on matters of Internet Governance. This reservation could signal that Russia plans to apply the telecommunications provisions in the ITRs to the Internet and to further regulate the Internet.

In addition to Resolution Plen/3, some of the proposals on the Internet were part of the discussion on Resolution Plen/5. This Resolution began as a basic resolution on invoicing for international telecommunication services, but ended up including numerous other provisions that did not make it into the main text of the treaty. Although the final text does not contain provisions explicitly mentioning the Internet, the introductory text of the Resolution mentions the transition of phone and data networks to IP-based networks. Also, the proposal that evolved into "resolves" originally applied to the relationship between network operators and application providers. During the discussions this proposal was modified to refer to "providers of international services" instead of application providers. Even with this modification, the application of this provision is ambiguous and could be applied to over-the-top providers.

Resolution Plen/5 is likely to reinforce work in Study Group 3 on accounting, fraud and charges for international telecommunications service traffic termination and exchange, etc.

Telecommunications Traffic Exchange Points

A proposal[8] concerning "telecommunication traffic exchange points" was included as an Article in the ITRs. The term "telecommunication traffic exchange point" was left undefined. This article does not mention the Internet or Internet Exchange Points, but the discussion of this Article included discussion on how it related to Internet Exchange Points. At least one delegation indicated that the Article was intended to help enable development of regional Internet Exchange Points.

Although this provision raised concerns over possible regulation of Internet Exchange Points, it focuses on creating an enabling environment for creation of regional telecommunication traffic exchange points. This environment could provide support for development of trans-border telecommunications and connectivity.

Route-Related Factors

Prior to the conference, there were several proposals [8] to require transparency into the international routes used for a Member States' traffic and to allow Member States to control what routes were used between them. Note that the definition of "route" in the ITRs is different from the concept of a "route" on the Internet. In the ITRs, a route is defined as the technical facilities used for telecommunications traffic between two telecommunication terminal exchanges or offices.

Coming into the WCIT, the proposal to control routes by Member States was dropped from the proposal, so the debate centered over whether Member States should have the right to know what routes were being used. After much discussion, the final result was a provision allowing "authorized operating agencies" (not Member States) to determine which routes are to be used between them and allowing the originating operator to determine the outbound route for traffic. This provision is not much different from how network operators manage their networks today.

Quality of Service Proposals

Several proposals[8] were made to WCIT to require QoS to be negotiated between network providers including Internet providers. Some proposals also allowed network providers to charge over-the-top providers for QoS.

The final provisions did not add any new requirements for QoS other than a nonspecific requirement related to mobile roaming. Although no new provisions were added specific to the Internet, it does not mean that countries could not try to impose the current QoS provisions to VoIP services. The debate over QoS on the Internet will continue outside the ITRs.

Naming, Numbering, and Addressing Proposals

Several countries and regions proposed[8] to extend provisions on telephone numbering to include naming, addressing, and origin identifiers. Several proposals were made to require delivery of calling party number and to cooperate in preventing the misuse ("misuse" not defined) of numbering, naming, and addressing resources. Although the Internet was not explicitly mentioned, these proposals were intended to apply to VoIP based on comments at pre-WCIT preparatory meetings[10].

In the end, several provisions were added related to delivery of calling party number and prevention of misuse of telecommunications numbering resources as defined in ITU-T Recommendations. Provisions to include naming, addressing, and more general "origin identifiers" were not accepted.

Even though there were no provisions specifically on the Internet, some countries could apply these provisions to VoIP services that use E.164 telephone numbers and that provide for bypass of the international telephony accounting system.

However, it is not clear that these provisions add any more authority than what these countries have today.

Content and Spam

Proposals[8] to include spam in the treaty caused a lot of contentious discussion, in ad hoc groups, plenary, and in consultations. Some countries took a strong position that spam is a content topic that was out of scope of the ITRs. There was a concern that adding a provision on spam would legitimize content filtering by governments. Some African countries insisted on including a provision on spam, claiming that it consumed a large percentage of their international bandwidth. In the end to address concern about content, a statement was added to Article 1.1:

> "These Regulations do not address the content-related aspects of telecommunications."

To address the proposals on spam, Article 5B was added on unsolicited bulk electronic communication:

> "Member States should endeavour to take necessary measures to prevent the propagation of unsolicited bulk electronic communications and minimize its impact on international telecommunication services. Member States are encouraged to cooperate in that sense."

As written the final text, it is fairly vague and could have implications beyond spam; for example, there are no exemptions for broadcasters or for emergency alert systems (for example, tsunami alerts). It is also not clear how Article 5B can be implemented consistent with the statement on content in Article 1.1.

It was clear from the discussion that many of the delegates from countries supporting this provision do not understand spam or spam-mitigation techniques and their usage (or not) in their own countries. It is clear that many of the delegates were not aware of basic best practices from the *Messaging Anti-Abuse Working Group* (MAAWG) and other organizations. These discussions highlighted the need for capacity building for developing countries on spam-mitigation techniques.

Human Rights and Member State Access to International Telecommunications

In a plenary session on the penultimate night of the WCIT, a provision on human rights was added to the final draft of the ITRs. This discussion led to a debate concerning the right of Member States to access international telecommunication services, originating from a proposal from Sudan and Cuba creating a right of Member States to access Internet websites. This provision was targeted at U.S. and European actions taken in response to UN sanctions against Sudan due to Darfur and U.S. sanctions on Cuba.

The provision provides a right for Member States, not its citizens. Thus it did not provide any rights for citizens to access international telecommunication services. In addition, it is not clear what or whose international telecommunication service Member States have a right to. The implications of the provision were unclear, and delegations did not have time to consult their home countries before the end of the conference.

Several times during the debate the Chair of the WCIT and the Secretary-General of the ITU both tried to dissuade the proponents from pushing their proposal, to no avail. After extended debate, Iran called for a point of order and then called for a vote, the only official vote of the conference. After the text passed by majority vote, the Chair of the WCIT declared the ITRs approved. At that point the United States, followed by the United Kingdom, Sweden, and other countries, made statements that they would not sign the treaty. Supporters of the treaty read their statements in favor of the treaty. The conference was effectively over[11].

The uncertainty caused by the addition of this text at such a late date and the way it was added created a situation in which many countries that might have signed the treaty ended up not signing. This provision more than any other disagreement in the conference caused the conference to split to the extent that it did.

Looking Forward

Much of the long-term impact of the treaty will not be felt until the signing governments ratify the treaty and start enacting provisions into either law or regulation. It is likely that some of the countries that did not sign in Dubai will accede to the treaty at a later time, including countries that did not attend the WCIT.

WCIT is only one step (though an important one) in the long-term debate over Internet Governance and the appropriate role of governments (and intergovernmental organizations) in the Internet. The debate will continue in numerous international fora going forward such as:

- World Telecommunications Policy Forum (May 2013)
- World Summit on the Information Society Action Line Forum (May 2013)
- ITU Council Working Group on Internet Public Policy (ongoing)
- ITU-T Study Group meetings (ongoing)
- ITU Plenipotentiary Conference (2014)
- WSIS+10 Review (2013–2015)

It has already been seen that many of the same topics debated at WCIT will be debated in these venues; for example, IP addressing, naming, spam, and cybersecurity. The WCIT Resolutions (especially Res. Plen/3) will likely be used to promote a larger role of the ITU in the Internet Governance debate.

The ITU's Plenipotentiary Conference in 2014 will be the next important treaty conference where the ITU's Constitution and Convention (both treaty instruments) can be revised. In the hierarchy of treaties at ITU, the ITU Constitution takes precedence over the ITRs, and many of the terms used in the ITRs are defined in the Constitution. Therefore, changes to the ITU Constitution could affect the meaning of the ITRs. The ITU Plenipotentiary will provide an opportunity for the ITU Member States to come together and heal some of the differences coming out of the WCIT, but it is also an opportunity to widen the rift.

The WSIS+10 Review will be an important process because it is likely to set the agenda for the discussion of Internet Governance for the 5–10 years after 2015, much as the Tunis Agenda from 2005 set the agenda for the last 8 years. An important aspect of the WSIS+10 Review is that it involves other UN agencies (for example, UNESCO) in addition to the ITU. Many of the events involve stakeholders whose voices are not normally heard at ITU conferences.

Some of the disagreements exhibited at WCIT brought to light opportunities for the Internet community to engage with governments and other stakeholders by providing technical and thought leadership. Capacity building with many of the developing country governments will be an important part of the preparation leading up to the major international conferences such as the ITU Plenipotentiary and WSIS+10.

Much of the growth of the Internet going forward is likely to come in the countries that signed the ITRs. Many of these countries have started to develop multistakeholder consultations and processes when dealing with Internet topics. The fact that a government signed the ITRs does not mean that the country is somehow against the Internet. On the contrary, many of these countries are looking for ways to accelerate the Internet's development within their borders and to accelerate their international connectivity to the Internet. As the Internet grows and develops in these countries, the Internet communities in these countries will likely look to play a larger role in a consultative process regarding government positions on issues related to Internet Governance. Future growth of the Internet across ITR boundaries (signatories and non-signatories) will depend on cooperation amongst all stakeholders.

References

[0] Geoff Huston, "December in Dubai: Number Misuse, WCIT, and ITRs," *The Internet Protocol Journal*, Volume 15, No. 2, June 2012.

[1] World Conference on International Telecommunications (WCIT-12), **http://www.itu.int/en/wcit-12/Pages/default.aspx**

[2] International Telecommunication Regulations, **http://www.itu.int/en/wcit-12/Pages/itrs.aspx**

[3] "Discover ITU's History," **http://www.itu.int/en/history/Pages/DiscoverITUsHistory.aspx**

[4] "International Settlements Policy and U.S.–International Accounting Rates," **http://www.fcc.gov/encyclopedia/international-settlements-policy-and-us-international-accounting-rates**

[5] "WATTC-88 World Administrative Telegraph and Telephone Conference (Melbourne, 1988)," **http://www.itu.int/en/history/Pages/TelegraphAndTelephoneConferences.aspx?conf=33&dms=S0201000021**

[6] "Signatories of the Final Acts: 89 (in green)," **http://www.itu.int/osg/wcit-12/highlights/signatories.html**

[7] "Dr. Hamadoun I. Touré, ITU Secretary-General First Plenary of World Conference on International Telecommunications (WCIT-12)," **http://www.itu.int/en/wcit-12/Pages/speech-toure2.aspx**

[8] "Proposals Received from ITU Member States for the Work of the Conference," **http://www.itu.int/md/dologin_md.asp?lang=en&id=S12-WCIT12-121203-TD-0001!!MSW-E**

[9] "Tunis Agenda for the Information Society," *World Summit on the Information Society,* 2005. **http://www.itu.int/wsis/docs2/tunis/off/6rev1.html**

[10] Council Working Group to Prepare for the 2012 WCIT, **http://www.itu.int/council/groups/cwg-wcit12/index.html**

[11] Webcast and Captioning of the WCIT, **http://www.itu.int/en/wcit-12/Pages/webcast.aspx**

ROBERT PEPPER leads Cisco's Global Technology Policy team working with governments across the world in areas such as broadband, IP enabled services, wireless and spectrum policy, security, privacy, Internet governance and ICT development. He joined Cisco in July 2005 from the FCC where he served as Chief of the Office of Plans and Policy and Chief of Policy Development beginning in 1989 where he led teams developing policies promoting the development of the Internet, implementing telecommunications legislation, planning for the transition to digital television, and designing and implementing the first U.S. spectrum auctions. He serves on the board of the U.S. Telecommunications Training Institute (USTTI) and advisory boards for Columbia University and Michigan State University, and is a Communications Program Fellow at the Aspen Institute. He is a member of the U.S. Department of Commerce's Spectrum Management Advisory Committee, the UK's Ofcom Spectrum Advisory Board and the U.S. Department of State's Advisory Committee on International Communications and Information Policy. Pepper received his BA. and Ph.D. from the University of Wisconsin-Madison. E-mail: **rmpepper@cisco.com**

CHIP SHARP has 30 years in the communications industry and currently is a Director in the Research and Advanced Development Department at Cisco Systems, Inc. His current role is in Technology Policy focusing on Internet Governance issues. He participated in the US preparatory process for WCIT from 2010 including as Private Sector Advisor to the US Delegation to the ITU's Council Working Group to Prepare for the 2012 WCIT (CWG-WCIT12), mainly analyzing the impact of proposals on the Internet and Internet Governance. He helped develop many of the talking points and position papers related to the Internet for the US Delegation. He served in the same capacity on the US Delegation to WCIT. He continues to be active in many follow-on activities preparing for the World Telecommunication Policy Forum (WTPF), World Summit on the Information Society 10 year review (WSIS+10), ITU Plenipotentiary Conference 2014 and Internet Governance Forum. He also currently service on the FCC's Open Internet Advisory Committee. Prior to this role, he led a multinational, multidisciplinary team at Cisco helping drive various technologies such as LISP, DNSSEC, BGPSEC, ENUM, Lawful Intercept etc. He has also supported capacity building and development programs for developing countries, for example, deployment of Internet Exchange Points (IXPs). He started at Cisco in 1996 helping design dialup Internet access products to interface with legacy telco signaling systems. Prior to Cisco, Chip worked at Teleos Communications, AT&T Consumer Product Labs and NASA's Communications Division. E-mail: **chsharp@cisco.com**

Letters to the Editor

Dear Ole,
I am sorry that there is some delay (more than 1 second) between the arrival of *The Internet Protocol Journal* at my desk and this e-mail. In the December 2012 issue (Volume 15, No. 4), Geoff Houston discusses the extra second on the last minute of the 31st of June. There is no 31st of June in the calendar, at least not in old Europe, but maybe in the United States. It is funny to discuss the problem of a second at the end of a nonexistent day, isn't it?

Nevertheless I could take some new knowledge from this article.

Best regards,

—Richard Schuerger
richard.schuerger@gmx.de

Hi Geoff (and Ole)!
I am sitting comfortably in a chair on the terrace in a Tenerife house, reading the December 2012 issue of IPJ, which I received by mail today. Since I have been working many years with the *Network Time Protocol* (NTP), I started reading your article on the subject with great interest. Having read only a few sentences I jumped in my chair:

> "Back at the end of June 2012 there was a brief IT hiccup as the world adjusted the *Coordinated Universal Time* (UTC) standard by adding an extra second to the last minute of the 31st [!!] of June."

Of course you may have received numerous notices of this hiccup [ha, ha], but still I couldn't resist writing to you. Thank you for an [otherwise] well-written and clarifying article (as always).

—Truls Hjelle
truls@sund-hjelle.org

PS: Thanks to Ole for this anachronism on paper still available to us oldies who prefer sitting with a paper magazine in the sun instead of gazing at a poorly lit screen and struggling with the tiny letters.

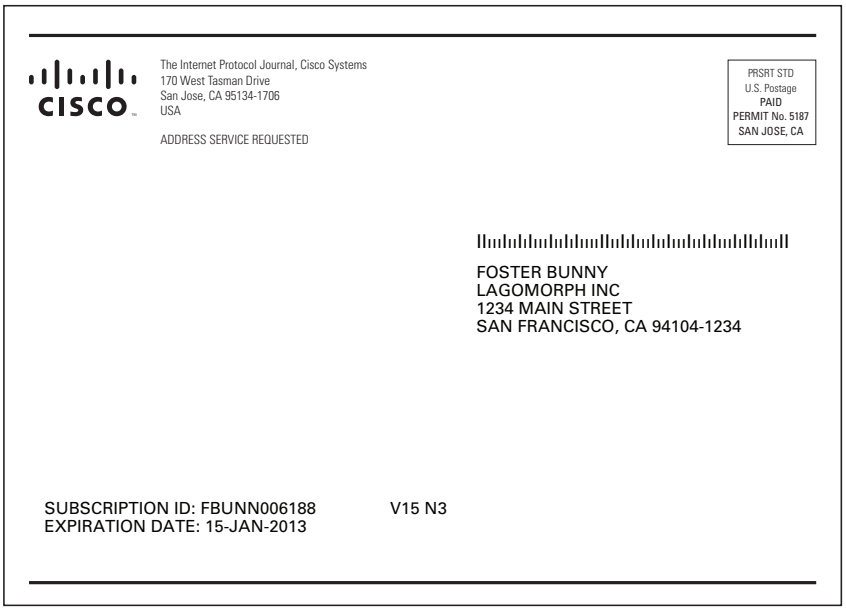
The author responds:

Back in 45 BC, Julius Caesar made same revolutionary changes to the Roman calendar, and the changes included adding one extra day to June (well not quite, as the letter "J" was not around until the 16th Century, and the letter "u" was also yet to makes its debut, so it is probably less of an anachronism to record that Gaivs Ivlivs Caesar added an extra day to the month of Ivnivs). Either way, this change brought the total number of days in the month of June to 30, which is where it has remained for 2058 years.

It is often said that Australia operates on a calendar all of its own, but while our isolation on a largish rock at the southern end of the Pacific Ocean has led to a number of revolutionary innovations that are easily on a par with fire and the wheel, including the world-renowned stump-jump plough and the sheep-shearing machine, we Australians have not yet turned our collective national genius to the calendar. Despite a pretty sensible suggestion from the latest meeting of the Grong Grong Shire Council for a year to be made up of 10 months of 30 days followed by a decent 65-day session at the pub, we have yet to get the blokes back from the pub after their last 65-day bender, so that plan needs some more work back at the shed before it gets another airing! Thus it looks like Australia uses the same calendar as everyone else, making the reference to the 31st of June one of those pesky brain-fade errors! Oops. Yes, it was meant to say 30th of June. Well spotted!

—Geoff Huston

gih@apnic.net



Don’t forget to renew and update your subscription. For details see the IPJ Subscription FAQ in our previous issue (Volume 15, No. 4).

Book Review

On Internet Freedom

On Internet Freedom, by Marvin Ammori, Elkat Books, January 2013, sold by: Amazon Digital Services, Inc., ASIN: B00B1MQZNW.

Marvin Ammori has written an important book about the threats to free speech and expression that we are not only privileged to conduct on the Internet today but have come to treat as basic human rights.

On Internet Freedom looks at the past, present, and future of the Internet as a speech technology. Ammori examines how the coordinated and determined efforts by Big Content to protect content and increasing efforts by governments to censor content threaten Internet use as we embrace it today. Ammori also explains how these acts were in fact anticipated by Clark, Sollins, Wroclawski, and Braden in a paper entitled "Tussle in Cyberspace: Defining Tomorrow's Internet,"[1] where the authors assert:

> "User empowerment, to many, is a basic Internet principle, but for this paper, it is the manifestation of the right to choose—to drive competition, and thus drive change."

Ammori cites only the first clause of this sentence—as a technologist, I believe the second is extremely important as well—but he makes clear that the end-to-end design of the Internet establishes a fundamental thesis:

> "If user choice is our design principle, then users should have the final say."

Unfortunately, Ammori explains that users do not have the final say but are increasingly challenged by lawyers, bureaucrats, commissioners, and others who are motivated to constrain their freedoms and who want to do so by altering the fundamental design of the Internet. Ammori's response, admittedly U.S.-centric, is simple: the Internet is a speech technology, and:

> "... the ultimate design principle for any speech technology, at least in the United States: the First Amendment, which protects freedom of speech. The *First Amendment* is not generally thought of as a design principle, but, by definition, it limits what Congress or any other government actor may or may not adopt in shaping the Internet's future."

This statement sets the context for the remainder of the book. In Part II, Ammori looks at events leading to the 18 January 2012 Internet Blackout in protest of the *Stop Online Piracy Act* (SOPA) and *PROTECT IP Act* (PIPA) and how these and possibly future legislation threaten "...the speech tools of the many while reshaping our speech environment for the benefit of the few."

Conveniently, Part II is largely about how the few benefit. Before judging whether you believe this theory is even-handed or not, remember that the litmus test throughout this book is the First Amendment of the U.S. Constitution. This part ought to make every Internet user or free speech advocate pause, or shiver. One of the most worrisome speculations Ammori offers is the extent to which legislation could stilt adoption of emerging technologies such as *three-dimensional* (3D) printing or stifle future innovations of this kind.

Part III looks at how the Internet as speech technology influences governments, how governments have attempted to exert influence, and how Internet users and dominant Internet forces (Google, Amazon, Facebook, and Twitter) respond. This part will probably be illuminating for most readers, because it explains situations where a *private conversation* between a government official and an *Internet Service Provider* (ISP) or hosting company can circumvent the First Amendment, and why *Terms of Service* are often more speech-restricting than the First Amendment as well.

Part IV focuses on net neutrality concerns. Ammori draws the lines of conflict: ISPs seek to differentiate, rate-control, block, or charge users differently for content that is transmitted on their networks. However, content includes speech, and if the Internet is speech technology, then ISPs should not be able to decide what you say or see, or they do so in violation of your First Amendment rights. Ammori also explains that net neutrality is not only a First Amendment concern but also an economic one: net neutrality violations can influence investments in or creation of new technology.

I began by saying that Marvin Ammori has written an important book. It is also an extremely readable book. Ammori does a commendable job explaining constitutional law and technology in easy to understand terms. I highly recommend the book not only for people who are interested in law or technology but for anyone who advocates freedom of expression.

On Internet Freedom is currently available as a Kindle download.

[1] David D. Clark, John Wroclawski, Karen R. Sollins, and Robert Braden, "Tussle in Cyberspace: Defining Tomorrow's Internet," *IEEE/ACM Transactions on Networking*, Volume 13, Issue 3, June 2005. Available from:
http://groups.csail.mit.edu/ana/Publications/PubPDFs/Tussle2002.pdf

—*Dave Piscitello,* **dave@corecom.com**

Reprinted with permission from *The Security Skeptic* blog:
http://securityskeptic.typepad.com/the-security-skeptic/

Fragments

Nominations Sought for 2013 Jonathan B. Postel Service Award

The Internet Society is soliciting nominations of qualified candidates for the 2013 *Jonathan B. Postel Service Award* by May 31, 2013. This annual award is presented to an individual or organization that has made outstanding contributions in service to the data communications community. The award is scheduled to be presented during the 87th IETF meeting in Berlin, Germany, July 28–August 2.

The award was established by the Internet Society to honor a person who has made outstanding contributions in service to the data communications community. The award is focused on sustained and substantial technical contributions, service to the community, and leadership. With respect to leadership, the award committee places particular emphasis on candidates who have supported and enabled others in addition to their own specific actions.

The award is named for Dr. Jonathan B. Postel to recognize and commemorate the extraordinary stewardship exercised by Jon over the course of a thirty-year career in networking. He served as the editor of the RFC series of notes from its inception in 1969 until 1998. He also served as the ARPANET "Numbers Czar" and *Internet Assigned Numbers Authority* (IANA) over the same period of time. He was a founding member of the Internet Architecture (nee Activities) Board and the first individual member of the Internet Society, which he also served as a Trustee.

For more information, see: **http://www.internetsociety.org/**

Upcoming Events

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in Beijing, China, April 7–11, 2013 and in Durban, South Africa, July 14–18, 2013. For more information, see: **http://icann.org/**

The *North American Network Operators' Group* (NANOG) will meet in New Orleans, Louisiana, June 3–5, 2013 and in Phoenix, Arizona, October 7–9, 2013. For more information see: **http://nanog.org**

The *Internet Engineering Task Force* (IETF) will meet in Berlin, Germany, July 28–August 2, 2013 and in Vancouver, Canada, November 3–8, 2013. For more information see: **http://www.ietf.org/meeting/**

The *Asia Pacific Regional Internet Conference on Operational Technologies* (APRICOT) will meet in Bangkok, Thailand, February 18–28, 2014. For more information see: **http://www.apricot.net**

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

The Internet Protocol Journal

June 2013 Volume 16, Number 2

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

Fifteen years ago we published the first edition of *The Internet Protocol Journal* (IPJ). This seems like a good time to reflect on where the Internet is today and where it might be going in the future, instead of looking back at earlier developments the way we did in the tenth anniversary issue of IPJ.

In our first article, Geoff Huston discusses network service models, comparing the Internet to the traditional *Public Switched Telephone Network* (PSTN) in both technical and business terms, and asks if the fundamental architectural differences between these networks might explain the rather slow deployment of IPv6. Although the number of IPv6-connected users has doubled in the last year (see page 35), IPv6 still represents a small percentage of total Internet traffic.

The mobile device dominates today's Internet landscape. Smartphones and tablets are starting to replace more traditional computers for Internet access. Many technical developments have made this possible, including high-resolution screens; powerful processors; and compact, long-lasting batteries. Combine such developments with numerous radio-based technologies (GPS, cellular, Wi-Fi, and Bluetooth) and you end up with a handheld device that is always connected to the network and can perform almost any task, using an appropriate "app." Improvements to communications technologies such as the deployment of *Long-Term Evolution* (LTE) cellular data networks and *Gigabit Wi-Fi* (IEEE 802.11ac) are already underway.

We asked Vint Cerf, known to many as one of the "Fathers of the Internet," to look beyond what is possible with today's Internet and today's devices and predict what the future might look like in a world where every imaginable appliance is "smart," connected to the network, and location-aware. His article takes us through some history and current trends, and then describes how the future Internet might shape many aspects of society such as business, science, and education.

According to Wikipedia, a *Data Center* is "a facility used to house computer systems and associated components, such as telecommunications and storage systems." In our final article, Alvaro Retana and Russ White discuss how developments in link-state protocols, usually associated with wide-area networks, can be applied to data center networks.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Network Service Models and the Internet

by Geoff Huston, APNIC

In recent times we've covered a lot of ground in terms of the evolution of telecommunications services, riding on the back of the runaway success of the Internet. We have taken the computer and applied a series of transformational changes in computing power and size, battery technology, and added radio capabilities to create a surprising result. We've managed to put advanced computation power in a form factor that fits in the palms of our hands, and have coupled it with a communications capability that can manage data flows of tens if not hundreds of megabits per second—all in devices that have as few as two physical buttons! And we have created these devices at such scale that their manufacturing cost is now down to just tens of dollars per unit. The Internet is not just at the center of today's mass market consumer service enterprise, it is now at the heart of many aspects of our lives. It's not just the current fads of the social networking tools, but so much more. How we work; how we buy and sell, even what we buy and sell; how we are entertained; how democracies function, even how our societies are structured; and so much more—all of these activities are mediated by the Internet.

But a few clouds have strayed into this otherwise sunny story of technological wonder. Perhaps the largest of these clouds is that the underlying fabric of the Internet, the numbering plan of the network, is now fracturing. We have run out of IP addresses in the Asia Pacific region, Europe, and the Middle East. At the same time, the intended solution, namely the transition to a version of the IP protocol with a massively larger number space, IPv6, is still progressing at an uncomfortably slow pace. Although the numbers look like a typical "up and to the right" Internet data series, the vertical axis tells a somewhat different story. The overall deployment of IPv6 in today's Internet currently encompasses around 1.3 percent[0] of the total user base of the Internet, and it is possible that the actions of the open competitive market in Internet-based service provision will not necessarily add any significant further impetus to this necessary transition.

We have gone through numerous phases of explanation for this apparently anomalous success-disaster situation for the Internet. Initially, we formed the idea that the slow adoption of IPv6 was due to a lack of widely appreciated knowledge about the imminent demise of IPv4 and the need to transition the network to IPv6. We thought that the appropriate response would be a concerted effort at information dissemination and awareness rising across the industry, and that is exactly what we did. But the response, as measured in terms of additional impetus for the uptake of IPv6 in the Internet, was not exactly overwhelming.

We then searched for a different reason as to why this IPv6 transition appeared to be stalling. There was the thought that this problem was not so much a technical one as a business or a market-based one, and there was the idea that a better understanding of the operation of markets and the interplay between markets and various forms of public sector initiatives could assist in creating a stronger impetus for IPv6 in the service market. The efforts at stimulation of the market to supply IPv6 goods and services through public sector IPv6 purchase programs have not managed to create a "tipping point" for adoption of IPv6.

Some have offered the idea that the realization of IPv4 exhaustion would focus our thinking and bring some collective urgency to our actions. But although IPv4 address exhaustion in the Asia Pacific region in 2011 has created some immediate interest in IPv4 address extension mechanisms, the overall numbers on IPv6 adoption have stubbornly remained under 1.5 percent of the 2 billion user base of the Internet.

Why has this situation occurred? How can we deliberately lead this prodigious network into the somewhat perverse outcomes that break to basic end-to-end IP architecture by attempting to continue to overload the IPv4 network with more and more connected devices? What strange perversity allows us to refuse to embrace a transition to a technology than can easily sustain the connection needs of the entire silicon industry for many decades to come and instead choose a path that represents the general imposition of additional cost and inefficiency?

Perhaps something more fundamental is going on here that reaches into the architectural foundations of the Internet and may explain, to some extent, this evident reluctance of critical parts of this industry to truly engage with this IPv6 transition and move forward.

Telephony Network Intelligence

Compared to today's "smart" phone, a basic telephone handset was a remarkably basic instrument. The entire telephone service was constructed with a model of a generic interface device that was little more than a speaker, a microphone, a bell, and a pulse generator. The service model of the telephone, including the call-initiation function of dialing and ringing, the real-time synchronous channel provision to support bidirectional speech, all forms of digital and analogue conversion, and of course the call-accounting function, were essentially all functions of the network itself, not the handset. Although the network was constructed as a real-time switching network, essentially supporting a model of switching time slots within each of the network switching elements, the service model of the network was a "full-service" model.

The capital investment in the telecommunications service was therefore an investment in the network—in the transmission, switching, and accounting functions.

Building these networks was an expensive undertaking in terms of the magnitude of capital required. By the end of the 20th century the equipment required to support synchronous time switching included high-precision atomic time sources, a hierarchy of time-division switches to support the dynamic creation of edge-to-edge synchronous virtual circuits, and a network of transmission resources that supported synchronous digital signaling. Of course although these switching units were highly sophisticated items of technology, most of this investment capital in the telephone network was absorbed by the last mile of the network, or the so-called "local loop."

Although the financial models to operate these networks varied from operator to operator, it could be argued that there was little in the way of direct incremental cost in supporting a "call" across such a network, but there is a significant opportunity or displacement cost. These networks have a fixed capacity, and the requirements for supporting a "call" are inelastic. When a time slot is being used by one call, this slot is unavailable for use by any other call.

Telephony Tariffs

Numerous models were used when a retail tariff structure for telephony was constructed. One model was a "subscription model," where, for a fixed fee, a subscriber could make an unlimited number of calls. In other words the operator's costs in constructing and operating the network were recouped equally from all the subscribers to the network, and no transaction-based charges were levied upon the subscriber. This model works exceptionally well where the capacity of the network to service calls is of the same order as the peak call demand that is placed on the network. In other words, where the capacity of the network is such that the marginal opportunity or displacement cost to support each call is negligible, there is no efficiency gain in imposing a transactional tariff on the user. In the United States' telephone network, for example, a common tariff structure was that the monthly telephone service charge also allowed the subscriber to make an unlimited number of local calls.

Another model in widespread use in telephony was of a smaller, fixed service charge and a per-transaction charge for each call made. Here a subscriber was charged a fee for each call (or "transaction") that the subscriber initiated. The components to determine the charge for an individual transaction included the duration of the call, the distance between the two end parties of the call, the time of day, and the day of the week. This model allowed a network operator to create an economically efficient model of exploitation of an underlying common resource of fixed capacity. This model of per-call accounting was widespread, used by some operators in local call zones, and more widely by telephone service operators in long distance and international calls.

This model allowed the operator to generate revenue and recoup its costs from those subscribers who used the service, and, by using the pricing function, the network operator could moderate peak demand for the resource to match available capacity.

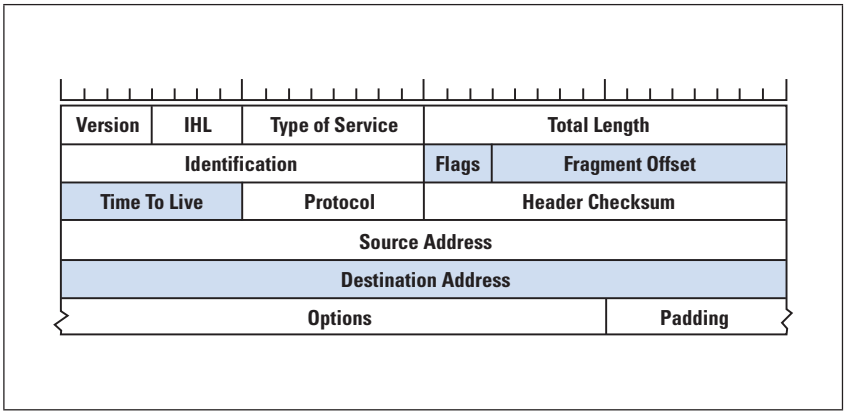
This per-transaction service model of telephony was available to the operator of the telephone service simply because the entire function of providing the telephone service was a network-based service. The network was aware of who initiated the transaction, who "terminated" the transaction, how long the transaction lasted, and what carriers were involved in supporting it. Initially this transactional service model was seen as a fair way to allocate the not inconsiderable costs of the construction and operation of the network to those who actually used it, and allocate these costs in proportion to the relative level of use. I suspect, though, that this fair cost allocation model disappeared many decades ago because these per-transaction service tariffs became less cost-based and more based on monopoly rentals.

IP Network Minimalism

The Internet is different. Indeed, the Internet is about as different from telephony as one could possibly imagine. The architecture of the Internet assumes that a network transaction is a transaction between computers. In this architecture the computers are highly capable signal processors and the network is essentially a simple packet conduit. The network is handed "datagrams," which the network is expected to deliver most of the time. However, within this architecture the network may fail to deliver the packets, may reorder the packets, or may even corrupt the content of the packets. The network is under no constraint as to the amount of time it takes to deliver the packet. In essence, the expectations that the architecture imposes on the network are about as minimal as possible. Similarly, the information that the edge-connected computers now expose to the network is also very limited. To illustrate this concept, it is useful to look at the fields that the Internet Protocol exposes to the network.

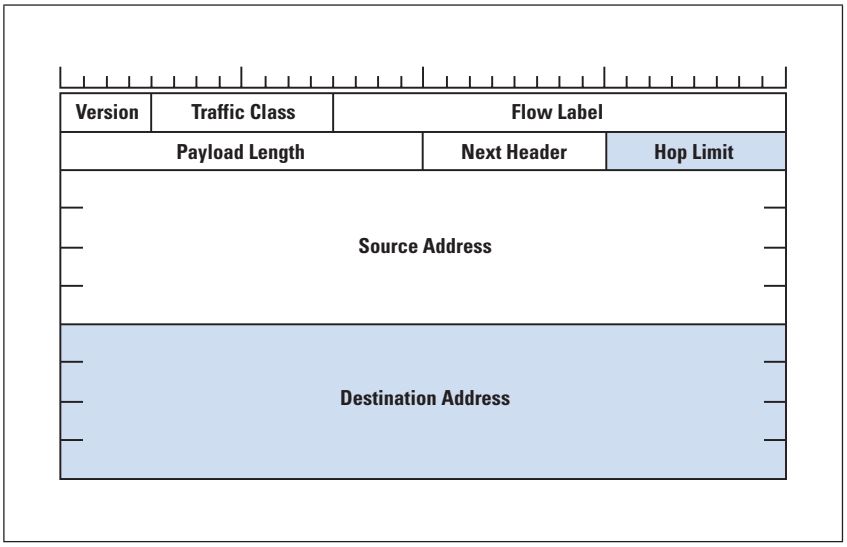
In IPv4 the fields of the Internet Protocol header are a small set, as shown in Figure 1. An IP packet header exposes the protocol *Version, Header Length* (IHL), *Total Length* of the IP packet, packet *Fragmentation Offset,* and *Type of Service* fields, a hop counter (*Time To Live* field), a *Header Checksum* field, and the *Source and Destination Address* fields. In practice, the Type of Service field is unused, and the Length and Checksum fields have information that is also contained in the data link frame header. What is left is the protocol Version field, packet length (Total Length field), the Fragmentation Offset field, a hop counter, and the Source and Destination Address fields. Of these fields, the Packet Length, Fragmentation Offset, hop counter, and Destination Address are the fields used by the network to forward the packet to its ultimate destination.

Figure 1: The IPv4 Packet Header



In IPv6 this minimal approach was further exercised with the removal of the Fragmentation Control fields and the Checksum fields (Figure 2). Arguably, the *Traffic Class* and *Flow Label* are unused, leaving only the *Protocol Version*, *Payload Length*, a *Hop Counter*, and the source and destination addresses exposed to the network. In IPv6 the minimal network-level information is now reduced to the packet length, the hop counter, and the destination address.

Figure 2: The IPv6 Packet Header



These fields represent the totality of the amount of information that the Internet Protocol intentionally exposes to the network. There are no transaction identifiers, no call initiation or call teardown signals, or even any reliable indication of relative priority of the packets. All the network needs to "see" in each carried packet is a hop counter, a packet length, and a destination address.

Within this model the actions of each of the network's switching elements are extremely simple, as shown in Figure 3.

Figure 3: IPv4 and IPv6 Packet Processing

```
for each received packet:
      decrement the hop counter
          if the counter value is zero then discard the packet, otherwise…
      look up the packet's destination address in a local table
          if the lookup fails then discard the packet, otherwise…
      look up the output queue from the located table entry
          if the queue is full discard the packet, otherwise…
          if the packet is too large for the outbound interface then
                 fragment the packet to fit, if permitted (IPv4)
                 or discard the packet (IPv6), otherwise…
      queue the packet for onward transmission
```

The Internet Service Model

What happened to "transactions" in this service model? What happened to network state? What happened to resource management within the network? What happened to all the elements of network-based communications services? The simple answer is that within the architecture of the Internet it is not necessary to expose such a detailed view of transactional state to the underlying network just to have the network deliver a packet. From a network perspective, IP has thrown all of that network level function away!

In the context of the Internet service architecture, a "transaction" is now merely an attribute of the application that is run on the end systems, and the underlying network is simply unaware of these transactions. All the network "sees" is IP packets, and each packet does not identify to the network any form of compound or multi-packet transaction.

Because a transaction is not directly visible to the IP network operator, the implication is that any effort for an IP service provider to use a transactional service tariff model becomes an exercise in frustration, given that there are no such network-visible interactions that could be used to create a transactional service model. In the absence of a network-based transactional service model, the *Internet Service Provider* (ISP) has typically used an access-based model as the basis of the IP tariff. Rather than paying a tariff per "call" the ISP typically charges a single flat fee independent of the number or nature of individual service transactions. Some basic differentiation is provided by the ability to apply price differentials to different access bandwidths or different volume caps, but this form of market segmentation is a relatively coarse one. Finer levels of transactional-based prices, such as pricing each individual video stream—or even pricing every individual webpage fetch—are not an inherent feature of such an access-based tariff structure.

The consequence for ISPs here is that within a single network access bandwidth class, this service model does not differentiate between heavy and light users, and is insensitive to the services operated across the network and to the average and peak loads imposed by these services. Like the flat-rate local telephone access model, the Internet pricing model is typically a flat-rate model that takes no account of individual network transactions. The ISP's service-delivery costs are, in effect, equally apportioned across the ISP's user base.

Interestingly, this feature has been a positive one for the Internet. With no marginal incremental costs for network usage, users are basically incented to use the Internet. In the same vein suppliers are also incented to use the Internet, because they can deliver goods and services to their customer base without imposing additional transaction costs to either themselves or their customers. For example, we have seen Microsoft and Apple move toward a software distribution model that is retiring the use of physical media, and moving to an all-digital Internet-based service model to support their user base. We have also seen other forms of service provision where the access-based tariff model has enabled services that would otherwise not be viable—here Netflix is a good example of such services that have been enabled by this flat-rate tariff structure. The attraction of cloud-based services in today's online world is another outcome of this form of incentive.

The other side effect of this shift in the architecture of the Internet is that it has placed the carriage provider—the network operator—into the role of a commodity utility. Without any ability to distinguish between various transactions, because the packets themselves give away little in terms of reliable information about the nature of the end-to-end service transaction, the carriage role is an undistinguished commodity utility function. The consequent set of competitive pressures in a market that is not strongly differentiated ultimately weans out all but the most efficient of providers from the service provider market—as long as competitive interests can be brought to bear on these market segments.

Invariably, consumers value the services that a network enables, rather than the network itself. In pushing the transaction out of the network and into the application, the architecture of the Internet also pushed value out of the network. Given that a service in the Internet model is an interaction between applications running on a content service provider's platform and on their clients' systems, it is clear that the network operator is not a direct party to the service transaction. An ISP may also provide services to users, but it is by no means an exclusive role, and others are also able to interact directly with customers and generate value through the provision of goods and services, without the involvement of the underlying network operators. It is not necessary to operate a network in order to offer a service on the Internet. Indeed, such a confusion of roles could well be a liability for such a carriage and content service provider.

The Content Business Model of the Internet

This unbundling of the service provision function from the network has had some rather unexpected outcomes. Those who made the initial forays of providing content to users believed that this function was no different from that of many retail models, where the content provider formed a set of relationships with a set of users. The direct translation of this model encountered numerous problems, not the least of which was reluctance on the part of individual users to enter into a panoply of service and content relationships. When coupled with considerations of control of secondary redistribution of the original service, this situation created some formidable barriers to the emergence of a highly valuable market for content and services on the Internet.

However, as with many forms of mass market media, the advertising market provides some strong motivation. With a traditional print newspaper, the full cost of the production of the newspaper is often borne largely by advertisers rather than by the newspaper readers. But newspaper advertising is a relatively crude exercise, in that the advertisement is visible to all readers, but it is of interest to a much smaller subset. The Internet provided the potential to customize the advertisement.

The greatest market value for advertisements is generated by those operations that gain the most information about their customers. These days it has a lot to do with knowledge of the consumer. It could be argued that Facebook's $1B purchase of Instagram was based on the observation that the combination of an individual's pictures and updates forms an amazingly rich set of real-time information about the behavior and preferences of individual consumers. It could also be argued that Google's business model is similarly based on forming a comprehensive and accurate picture of individual users' preferences, which is then sold to advertisers at a significant premium simply because of its tailored accuracy. And the mobile services are trying to merge users' current locations with the knowledge of their preferences to gain even greater value.

These developments are heading in the direction of a multiparty service model, where the relationship between a content provider and a set of users allows the content provider to resell names of these users to third parties through advertising. This on-selling of users' profiles and preferences is now a very sophisticated and significant market. As reported in [1], some 90 percent of Google's $37.9B income was derived from advertising revenue. The cost per click for "cheap car insurance" is reported in the same source to be $33.97!

The Plight of the Carrier

Although the content market with its associated service plane is now an extraordinarily valuable activity, the same is not true for the network operator—whose carriage function has been reduced from complete service-delivery management to a simple packet carrier without any residual visibility into the service plane of the network.

Obviously, network carriers look at these developments with dismay. Their own traditional value-added market has been destroyed, and the former model where the telcos owned everything from the handset onward has now been replaced by a new model that relegates them to a role similar to electricity or water reticulation—with no prospect of adding unique value to the content and service market. The highly valuable service-level transactions are effectively invisible to the carriage service providers of the Internet.

There is an evident line of thought in the carriage industry that appears to say: "If we could capture the notion of a service-level transaction in IP we could recast our service profile into a per-transaction profile, and if we can do that, then we could have the opportunity to capture some proportion of the value of each transaction."

Short of traffic interception, could the network operators working at the internet level of the network protocol stack have a means to identify these service-level transactions? The generic answer is "no," as we have already seen, but there are some other possibilities that could expose service-level transactions to the network operator.

QoS to the Rescue?

The recent calls by the The *European Telecommunications Network Operators' Association* (ETNO) advocating the widespread adoption of IP *Quality of Service* (QoS) appear to have some context from this perspective of restoring transaction visibility to the IP carriage provider. In the QoS model an application undertakes a QoS "reservation" with the network. The network is supposed to respond with a commitment to reserve the necessary resources for use by this transaction. The application then uses this QoS channel for its transaction, and releases the reservation when the transaction is complete.

From the network operator's perspective, the QoS-enabled network is now being informed of individual transactions, identifying the end parties for the transaction, the nature of the transaction and its duration, as well as the resource consumption associated with the transaction. From this information comes the possibility for the QoS IP network operator to move away from a now commonplace one-sided flat access tariff structure for IP services, and instead use a transactional service model that enables the network operator to impose transaction-based service fees on both parties to a network service if it so chooses. It also interposes the network operator between the content provider and the consumer, permitting the network operator to mediate the content service and potentially convert this gateway role into a revenue stream.

Of course the major problem in this QoS model is that it is based on a critical item of Internet mythology—the myth that inter-provider QoS exists on the Internet. QoS is not part of today's Internet, and there is no visible prospect that it will be part of tomorrow's Internet either!

Knotting up NATs

But QoS is not the only possible approach to exposing service-level transactions to the carriage-level IP network operator. Interestingly, the twin factors of the exhaustion of IPv4 addresses and the lack of uptake of IPv6 offers the IP network operator another window into what the user is doing, and, potentially, another means of controlling the quality of the user's experience by isolating individual user-level transactions at the network level.

When there are not enough addresses to assign each customer a unique IP address, the ISP is forced to use private addresses and operate a *Network Address Translator* (NAT)[2] within the carriage network.

However, NATs are not stateless passive devices. A NAT records every TCP and *User Datagram Protocol* (UDP) session from the user, as well as the port addresses the application uses when it creates a binding from an internal IP address and port to an external IP address and port. A new NAT binding is created for every user transaction: every conversation, every website, every streamed video, and literally everything else. If you were to look at the NAT logs that record this binding information, you would find a rich stream of real-time user data that shows precisely what each user is doing on the network. Every service transaction is now visible at the network level. How big is the temptation for the IP network operator to peek at this carrier-operated NAT log and analyze what it means?

Potentially, this transaction data could be monetized, because it forms a real-time data feed of every customer's use of the network. At the moment carriers think that they are being compelled to purchase and install this NAT function because of the IPv4 address situation. NATs offer a method for the carriage operator to obtain real-time feeds of customer behavior without actively intruding themselves into the packet stream. The NAT neatly segments the customer's traffic into distinct transactions that are directly visible to the NAT operator. I suspect that when they look at the business case for purchasing and deploying these *Carrier-Grade NAT* devices, they will notice a parallel business case that can be made to inspect the NAT logs and perhaps to either on-sell the data stream or analyze it themselves to learn about their customers' behavior.[3] And, as noted, there is already market evidence that such detailed real-time flows of information about individual users' activities can be worth significant sums.

But it need not necessarily be limited to a passive operation of stalking the user's online behavior. If the carriage provider were adventurous enough, it could bias the NAT port-binding function to even make some content work "better" than other content, by either slowing down the binding function for certain external sites or rationing available ports to certain less-preferred external sites. In effect, NATs provide many exploitable levers of control for the carriage operator, bundled with a convenient excuse of "we had no choice but to deploy these NATs!"

Where Now?

In contrast, what does an investment in IPv6 offer the carriage provider? An admittedly very bleak response from the limited perspective of the carriage service provider sector is that what is on offer with IPv6 is more of what has happened to the telecommunications carriage sector over the past 10 years, with not even the remote possibility of ever altering this situation. IPv6 certainly looks like forever, so if the carriers head down this path then the future looks awfully bleak for those who are entirely unused to, and uncomfortable with, a commodity utility provider role.

So should we just throw up our hands at this juncture and allow the carriage providers free rein? Are NATs inevitable? Should we view the introduction of transactional service models in the Internet as a necessary part of its evolution? I would like to think that these developments are not inevitable for the Internet, and that there are other paths that could be followed here. The true value for the end consumer is not in the carriage of bits through the network, but in the access to communication and services that such bit carriage enables. What does that reality imply for the future for the carriage role? I suspect that despite some evident misgivings, the carriage role is inexorably heading to that of a commodity utility operation.

This is not the first time an industry sector has transitioned from production of a small volume of highly valuable units to production of a massively larger volume of commodity goods, each of which has a far lower unit value, but generates an aggregate total that is much larger. The computing industry's transition from mainframe computers to mass market consumer electronics is a good example of such a transformation. As many IT sector enterprises have shown, it is possible to make such transitions. IBM is perhaps a classic example of an enterprise that has managed numerous successful transformations that have enabled it to maintain relevance and value in a rapidly changing environment.

The models for electricity distribution have seen a similar form of evolution in the last century. In the 1920s in the United Kingdom, electricity was a low-volume premium product. The prices for electricity were such that to keep just 5 light bulbs running for 1 day in a household cost the equivalent of an average week's wages. The consequent years saw public intervention in the form of nationalization of power generation and distribution that transformed electricity supply into a commonly available and generally affordable commodity.

The challenge the Internet has posed for the carriage sector is not all that different from these examples. The old carriage business models of relatively low-volume, high-value, transaction-based telecommunication services of telephony and faxes find no resonance within the service model of the Internet.

In the architecture of the Internet, it is the applications that define the services, while the demands from the underlying carriage network have been reduced to a simple stateless datagram-delivery service. Necessarily, the business models of carriage have to also change to adapt to this altered role, and one of the more fundamental changes is the dropping of the transaction-based model of the provision of telecommunications services for the carriage provider. What this situation implies for the carriage sector of the Internet is perhaps as radical as the transformation of the electricity supply industry during the period of the construction of the national grid systems in the first half of the 20th century.

The necessary change implied here is from a high-value premium service provider dealing in individual transactions across the network to that of a high-volume undistinguished commodity utility operator. The architectural concepts of a minimal undistinguished network carriage role and the repositioning of service management into end-to-end applications is an intrinsic part of the architecture of the Internet itself. It is not a universally acclaimed step—and certainly not one that is particularly popular in today's carriage industry—but if we want to see long-term benefits from the use of the Internet in terms of positive economic outcomes and efficient exploitation of this technology in delivering goods and services, then it is a necessary step in the broader long-term public interest.

References

[0] Google's IPv6 statistics:
http://www.google.com/ipv6/statistics.html

[1] Connor Livingston, "A breakdown of Google's top advertisers," **http://www.techi.com/2012/03/a-breakdown-of-googles-top-advertisers/**

[2] Geoff Huston, "Anatomy: A Look inside Network Address Translators," *The Internet Protocol Journal*, Volume 7, No. 3, September 2004.

[3] Geoff Huston, "All Your Packets Belong to Us," July 2012, **http://www.potaroo.net/ispcol/2012-07/allyourpackets.html**

GEOFF HUSTON, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of numerous Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005. He served on the Board of Trustees of the Internet Society from 1992 until 2001.
E-mail: **gih@apnic.net**

The Internet: Looking Forward

by Vint Cerf, Google

As I write, it is 2013 and 40 years have passed since the first drafts of the Internet design were written. The first published paper appeared in 1974[1] and the first implementations began in 1975. Much has happened since that time, but this essay is not focused on the past but, rather, on the future. Although the past is plainly prologue, our ability to see ahead is hampered by the unpredictable and the unknown unknowns that cloud and bedevil our vision. The exercise is nonetheless worth the effort, if only to imagine what might be possible.

Current trends reveal some directions. Mobile devices are accelerating access and applications. The economics of mobile devices have increased the footprint of affordable access to the Internet and the World Wide Web. Mobile infrastructure continues to expand on all inhabited continents. Speeds and functions are increasing as faster processors, more memory, and improved display technologies enhance the functions of these platforms. Cameras, microphones, speakers, sensors, multiple radios, touch-sensitive displays, and location and motion detection continue to evolve and open up new application possibilities. Standards and open source software facilitate widespread interoperability and adoption of applications. What is perhaps most significant is that these smart devices derive much of their power from access to and use of the extraordinary computing and memory capacity of the Internet. The Internet, cloud computing, and mobile devices have become hypergolic in their capacity to ignite new businesses and create new economic opportunities.

In the near term, the Internet is evolving. The *Domain Name System* (DNS) is expanding dramatically at the top level. Domain names can be written in non-Latin characters. The Internet address space is being expanded through the introduction of the IPv6 packet format, although the implementation rate among *Internet Service Providers* (ISPs) continues to be unsatisfactorily slow. This latter phenomenon may change as the so-called *Internet of Things*[2] emerges from its long incubation. Sensor networks, Internet-enabled appliances, and increasing application of artificial intelligence will transform the Internet landscape in ways that seem impossible to imagine. The introduction of IPv6 and the exhaustion of the older IPv4 address space have generated demand for application of the so-called *Network Address Translation* (NAT)[3] system. Geoff Huston has written and lectured extensively on this topic[4] and the potential futures involving their use. In some ways, these systems simultaneously interfere with the motivation to implement IPv6 and act as a bridge to allow both network address formats to be used concurrently.

Ironically, although most edge devices on the Internet today are probably IPv6-capable, as are the routers, firewalls, DNS servers, and other application servers, this advanced version of the Internet Protocol may not have been "turned on" by the ISP community. This situation is changing, but more slowly than many of us would like.

As the applications on the Internet continue to make demands on its capacity to transport data and to deliver low-latency services, conventional Internet technologies are challenged and new ideas are finding purchase in the infrastructure. The *OpenFlow*[5, 6] concept has emerged as a fresh look at packet switching in which control flow is segregated from data flow and routing is not confined to the use of address bits in packet headers for the formation and use of forwarding tables. Originally implemented with a central routing scheme to improve efficient use of network resources, the system has the flexibility to be made more distributed. It remains to be seen whether OpenFlow networks can be interconnected by using an extended form of the *Border Gateway Protocol* (BGP) so as to achieve end-to-end performance comparable to what has already been achieved in single networks.

Business models for Internet service play an important role here because end-to-end differential classes of service have not been realized, generally, for the current Internet implementations. Inter-ISP or edge-to-core commercial models also have not generally been perfected to achieve multiple classes of service. These aspirations remain for the Internet of the present day. Although it might be argued that increasing capacity in the core and at the edge of the Internet eliminates the need for differential service, it is fair to say that some applications definitely need lower delay, others need high capacity, and some need both (for example, for interactive video). Whether these requirements can be met simply through higher speeds or whether differential services must be realized at the edges and the core of the network is the source of substantial debate in the community. Vigorous experimentation and research continue to explore these topics.

Ubiquitous Computing

Mark Weiser[7] coined the term and concept of *Ubiquitous Computing*. He meant several things by this term, but among them was the notion that computers would eventually fade into the environment, becoming ever-present, performing useful functions, and operating for our convenience. Many devices would host computing capacity but would not be viewed as "computers" or even "computing platforms." Entertainment devices; cooking appliances; automobiles; medical, environmental, and security monitoring systems; our clothing; and our homes and offices would house many computing engines of various sizes and capacities. Many, if not all, would be interconnected in communication webs, responding to requirements and policies set by users or by their authorized representatives.

To this idyllic characterization, he implied there would be challenges: configurations of hundreds of thousands of appliances and platforms, privacy, safety, access control, information confidentiality, stability, resilience, and a host of other properties.

Even modest thought produces an awareness of the need for strong authentication to assure that only the appropriate devices and authorized parties are interacting, issuing instructions, taking data, etc. It is clear that multifactor authentication and some form of public key cryptography could play an important role in assuring limitations on the use and operation of these systems. Privacy of the information generated by these systems can be understood to be necessary to protect users from potential harm.

The scale of such systems can easily reach tens to hundreds of billions of devices. Managing complex interactions at such magnitudes will require powerful hierarchical and abstracting mechanisms. When it is also understood that our mobile society will lead to a constant background churn of combinations of devices forming subsets in homes, offices, automobiles, and on our persons, the challenge becomes all the more daunting. (By this I do not mean the use of mobile smartphones but rather a society that is geographically mobile and that moves some but not all its possessions from place to place, mixing them with new ones.) Self-organizing mechanisms, hierarchically structured systems, and systems that allow remote management and reporting will play a role in managing the rapidly proliferating network we call the Internet.

For further insight into this evolution, we should consider the position location capability of the *Global Positioning System* (GPS)[8]. Even small, low-powered devices (for example, mobile devices) have the ability to locate themselves if they have access to the proper satellite transmissions. Adding to this capability is geo-location using mobile cell towers and even known public Wi-Fi locations. In addition, we are starting to see appliances such as *Google Glass*[9] enter the environment. These appliances are portable, wearable computers that hear what we hear and see what we see and can respond to spoken commands and gestures. The Google self-driving cars[10] offer yet another glimpse into the future of computing, communication, and artificial intelligence in which computers become our partners in a common sensory environment—one that is not limited to the normal human senses. All of these systems have the potential to draw upon networked information and computing power that rivals anything available in history. The systems are potentially self-learning and thus capable of improvement over time. Moreover, because these devices may be able to communicate among themselves, they may be able to cooperate on a scale never before possible.

Even now we can see the outlines of a potential future in which virtually all knowledge can be found for the asking; in which the applications of the Internet continue to evolve; in which devices and appliances of all kinds respond and adapt to our needs, communicate with each other, learn from each other, and become part of an integrated and global environment.

Indeed, our day-to-day environment is very likely to be filled with information and data gathered from many sources and subject to deep analysis benefitting individuals, businesses, families, and governments at all levels. Public health and safety are sure to be influenced and affected by these trends.

Education

It is often noted that a teacher from the mid-19th century would not feel out of place in the classroom of the 21st, except, perhaps, for subject matter. There is every indication that this situation may be about to change. In 2012, two of my colleagues from Google, Peter Norvig and Sebastian Thrun, decided to use the Internet to teach an online class in artificial intelligence under the auspices of Stanford University. They expected about 500 students, but 160,000 people signed up for the course! There ensued a scramble to write or revise software to cope with the unexpectedly large scale of the online class. This phenomenon has been a long time in coming. Today we call such classes "MOOCs" (*Massive, Open, OnLine Classes*). Of the 160,000 who signed up, something like 23,000 actually completed the class. How many professors of computer science can say they have successfully taught 23,000 students?

The economics of this form of classroom are also very intriguing. Imagine a class of 100,000 students, each paying $10 per class. Even one class would produce $1,000,000 in revenue. I cannot think of any university that regularly has million dollar classes! There are costs, but they are borne in part by students (Internet access, equipment with which to reach the Internet, etc., for example) and in part by the university (Internet access, multicast or similar capability, and salaries of professors and teaching assistants). In some cases, the professors prepare online lectures that students can watch as many times as they want to—whenever they want to because the lectures can be streamed. The professors then hold classroom hours that are devoted to solving problems, in an inversion of the more typical classroom usage. Obviously this idea could expand to include nonlocal teaching assistants. Indeed, earlier experiments with video-taped lectures and remote teaching assistants were carried out with some success at Stanford University when I served on the faculty in the early 1970s.

What is potentially different about MOOCs is *scale*. Interaction and examinations are feasible in this online environment, although the form of exams is somewhat limited by the capabilities of the online platform used. Start-ups are experimenting with and pursuing these ideas (refer to **www.udacity.com** and **www.coursera.org**).

People who are currently employed also can take these courses to improve their skills, learn new ones, and position themselves for new careers or career paths. From young students to retired workers, such courses offer opportunities for personal expansion, and they provide a much larger customer base than is usually associated with a 2- or 4-year university or college program. These classes can be seen as re-invention of the university, the short course, the certificate program, and other forms of educational practice. It is my sense that this state of affairs has the potential to change the face of education at all levels and provide new options for those who want or need to learn new things.

The Information Universe

It is becoming common to speak of "big data" and "cloud computing" as indicators of a paradigm shift in our view of information. This view is not unwarranted. We have the ability to absorb, process, and analyze quantities of data beyond anything remotely possible in the past. The functional possibilities are almost impossible to fully fathom. For example, our ability to translate text and spoken language is unprecedented. With combinations of statistical methods, hierarchical hidden Markov models, formal grammars, and Bayesian techniques, the fidelity of translation between some language pairs approaches native language speaker quality. It is readily predictable that during the next decade, real-time, spoken language translation will be a reality.

One of my favorite scenarios: A blind German speaker and a deaf *American Sign Language* (ASL) signer meet, each wearing Google Glass. The deaf signer's microphone picks up the German speaker's words, translates them into English, and displays them as captions for the deaf participant. The blind man's Glass video camera sees the deaf signer's signs, translates the signs from ASL to English and then to German, and then speaks them through the bone conduction speaker of the Google Glass. We can do all of this now except for the correct interpretation of ASL. This challenge is not a trivial one, but it might be possible in the next 10 to 15 years.

The World Wide Web continues to grow in size and diversity. In addition, large databases of information are being accumulated, especially from scientific disciplines such as physics, astronomy, and biology. Telescopes (ground and space-based), particle colliders such as the Large Hadron Collider[11], and DNA sequencers are producing petabytes and more—in some cases on a daily basis!

We seem to be entering a time when much of the information produced by human endeavor will be accessible to everyone on the planet. Google's motto: "To organize the world's information and make it universally accessible and useful," might be nearly fulfilled in the decades ahead. Some tough problems lie ahead, however. One I call "bit rot."

By using this term, I do not mean the degradation of digital recordings on various media, although this is a very real problem. The more typical problem is that the readers of the media fall into disuse and disrepair. One has only to think about 8-inch Wang disks for the early Wang word processor, or 3.5-inch floppy disks or their 5 ¼-inch predecessors. Now we have CDs, DVDs, and Blu-Ray disks, but some computer makers—Apple for example—have ceased to build in readers for these media.

Another, more tricky problem is that much of the digital information produced requires software to correctly interpret the digital bits. If the software is not available to interpret the bits, the bits might as well be rotten or unreadable. Software applications run over operating systems that, themselves, run on computer hardware. If the applications do not work on new versions of the operating systems, or the applications are upgraded but are not backward-compatible with earlier file and storage formats, or the maker of the application software goes out of business and the source code is lost, then the ability to interpret the files created by this software may be lost. Even when open source software is used, it is not clear it will be maintained in operating condition for thousands of years. We already see backward-compatibility failures in proprietary software emerging after only years or decades.

Getting access to source code for preservation may involve revising notions of copyright or patent to allow archivists to save and make usable older application software. We can imagine that "cloud computing" might allow us to emulate hardware, run older operating systems, and thus support older applications, but there is also the problem of basic input/output and the ability to emulate earlier media, even if the physical media or their readers are no longer available. This challenge is a huge but important one.

Archiving of important physical data has to be accompanied by archiving of metadata describing the conditions of collections, calibration of instruments, formatting of the data, and other hints at how to interpret it. All of this work is extra, but necessary to make information longevity a reality.

The Dark Side

To the generally optimistic and positive picture of Internet service must be added a realistic view of its darker side. The online environment and the devices we use to exercise it are filled with software. It is an unfortunate fact that programmers have not succeeded in discovering how to write software of any complexity that is free of mistakes and vulnerabilities.

Despite the truly remarkable and positive benefits already delivered to us through the Internet, we must cope with the fact that the Internet is not always a safe place.

The software upon which we rely in our access devices, in the application servers, and in the devices that realize the Internet itself (routers, firewalls, gateways, switches, etc.) is a major vulnerability, given the apparently inescapable presence of bugs.

Not everyone with access to the Internet has other users' best interests at heart. Some see the increasing dependence of our societies on the Internet as an opportunity for exploitation and harm. Some are motivated by a desire to benefit themselves at the expense of others, some by a desire to hurt others, some by nationalistic sentiments, some by international politics. That Shakespeare's plays are still popular after 500 years suggests that human frailties have not changed in the past half millennium! The weaknesses and vulnerabilities of the Internet software environment are exploited regularly. What might the future hold in terms of making the Internet a safer and more secure place in which to operate?

It is clear that simple usernames and passwords are inadequate to the task of protecting against unauthorized access and that multifactor and perhaps also biometric means are going to be needed to accomplish the desired effect. We may anticipate that such features might become a part of reaching adulthood or perhaps a rite of passage at an earlier age. Purely software attempts to cope with confidentiality, privacy, access control, and the like will give way to hardware-reinforced security. Digitally signed *Basic Input/Output System* (BIOS), for example, is already a feature of some new chipsets. Some form of trusted computing platform will be needed as the future unfolds and as online and offline hazards proliferate.

Governments are formed that are, in principle, kinds of social contracts. Citizens give up some freedoms in exchange for safety from harm. Not all regimes have their citizens' best interests at heart, of course. There are authoritarian regimes whose primary interest is staying in power. Setting these examples aside, however, it is becoming clear that the hazards of using computers and being online have come to the attention of democratic as well as authoritarian regimes. There is tension between law enforcement (and even determination of what the law should be) and the desire of citizens for privacy and freedom of action. Balancing these tensions is a nontrivial exercise. The private sector is pressed into becoming an enforcer of the law when this role is not necessarily an appropriate one. The private sector is also coerced into breaching privacy in the name of the law.

"Internet Governance" is a broad term that is frequently interpreted in various ways depending on the interest of the party desiring to define it for particular purposes. In a general sense, Internet Governance has to do with the policies, procedures, and conventions adopted domestically and internationally for the use of the Internet. It has not only to do with the technical ways in which the Internet is operated, implemented, and evolved but also with the ways in which it is used or abused.

In some cases it has to do with the content of the Internet and the applications to which the Internet is put. It is evident that abuse is undertaken through the Internet. Fraud, stalking, misinformation, incitement, theft, operational interference, and a host of other abuses have been identified. Efforts to defend against them are often stymied by lack of jurisdiction, particularly in cases where international borders are involved. Ultimately, we will have to reach some conclusions domestically and internationally as to which behaviors will be tolerated and which will not, and what the consequences of abusive behavior will be. We will continue to debate these problems well into the future.

Our societies have evolved various mechanisms for protecting citizens. One of these mechanisms is the Fire Department. Sometimes volunteer, this institution is intended to put out building or forest fires to minimize risks to the population. We do not have a similar institution for dealing with various forms of "cyberfires" in which our machines are under attack or are otherwise malfunctioning, risking others by propagation of viruses, worms, and Trojan horses or participation in botnet denial-of-service or other forms of attacks. Although some of these matters may deserve national-level responses, many are really local problems that would benefit from a "Cyber Fire Department" that individuals and businesses could call upon for assistance. When the cyber fire is put out, the question of cause and origin could be investigated as is done with real fires. If deliberately set, the problem would become one of law enforcement.

Intellectual property is a concept that has evolved over time but is often protected by copyright or patent practices that may be internationally adopted and accepted. These notions, especially copyright, had origins in the physical reproduction of content in the form of books, films, photographs, CDs, and other physical things containing content. As the digital and online environment penetrates more deeply into all societies, these concepts become more and more difficult to enforce. Reproduction and distribution of digital content gets easier and less expensive every day. It may be that new models of compensation and access control will be needed in decades ahead.

Conclusion

If there can be any conclusion to these ramblings, it must be that the world that lies ahead will be immersed in information that admits of extremely deep analysis and management. Artificial intelligence methods will permeate the environment, aiding us with smart digital assistants that empower our thought and our ability to absorb, understand, and gain insight from massive amounts of information.

It will be a world that is also at risk for lack of security, safety, and privacy—a world in which demands will be made of us to think more deeply about what we see, hear, and learn. While we have new tools with which to think, it will be demanded of us that we use them to distinguish sound information from unsound, propaganda from truth, and wisdom from folly.

References

[1] Vinton G. Cerf and Robert E. Kahn, "A Protocol for Packet Network Intercommunication," *IEEE Transactions on Communications,* Vol. Com-22, No. 5, May 1974.

[2] David Lake, Ammar Rayes, and Monique Morrow, "The Internet of Things," *The Internet Protocol Journal,* Volume 15, No. 3, September 2012.

[3] Geoff Huston, "Anatomy: A Look inside Network Address Translators," *The Internet Protocol Journal,* Volume 7, No. 3, September 2004.

[4] Geoff Huston and Mark Kosters, "The Role of Carrier Grade NATs in the Near-Term Internet," TIP 2013 Conference, **http://events.internet2.edu/2013/tip/agenda.cfm?go=session&id=10002780**

[5] **http://www.openflow.org/**

[6] William Stallings, "Software-Defined Networks and OpenFlow," *The Internet Protocol Journal,* Volume 16, No. 1, March 2013.

[7] **http://en.wikipedia.org/wiki/Mark_Weiser**

[8] **http://en.wikipedia.org/wiki/Global_Positioning_System**

[9] **http://www.google.com/glass/start/**

[10] **http://en.wikipedia.org/wiki/Google_driverless_car**

[11] **home.web.cern.ch**

[12] Cerf, V., "Looking Toward the Future," *The Internet Protocol Journal,* Volume 10, No. 4, December 2007.

[13] Vint Cerf, "A Decade of Internet Evolution," *The Internet Protocol Journal,* Volume 11, No. 2, June 2008.

[14] Geoff Huston, "A Decade in the Life of the Internet," *The Internet Protocol Journal,* Volume 11, No. 2, June 2008.

VINTON G. CERF is vice president and chief Internet evangelist for Google. Cerf has held positions at MCI, the Corporation for National Research Initiatives, Stanford University, UCLA, and IBM. He served as chairman of the board of the Internet Corporation for Assigned Names and Numbers (ICANN) and was founding president of the Internet Society. Cerf was appointed to the U.S. National Science Board in 2013. Widely known as one of the "Fathers of the Internet," he received the *U.S. National Medal of Technology* in 1997, the *Marconi Fellowship* in 1998, and the *ACM Alan M. Turing Award* in 2004. In November 2005, he was awarded the *Presidential Medal of Freedom,* in April 2008 the *Japan Prize,* and in March 2013 the *Queen Elizabeth II Prize for Engineering*. He is a Fellow of the IEEE, ACM, and AAAS, the American Academy of Arts and Sciences, the American Philosophical Society, the Computer History Museum, and the National Academy of Engineering. Cerf holds a Bachelor of Science degree in Mathematics from Stanford University and Master of Science and Ph.D. degrees in Computer Science from UCLA, and he holds 21 honorary degrees from universities around the world.
E-mail: **vint@google.com**

Optimizing Link-State Protocols for Data Center Networks

by Alvaro Retana, Cisco Systems, and Russ White, Verisign

With the advent of cloud computing[6, 7], the pendulum has swung from focusing on wide-area or global network design toward a focus on *Data Center* network design. Many of the lessons we have learned in the global design space will be relearned in the data center space before the pendulum returns and wide-area design comes back to the fore.

This article examines three extensions to the *Open Shortest Path First* (OSPF) protocol that did not originate in the data center field but have direct applicability to efficient and scalable network operation in highly meshed environments. Specifically, the application extensions to OSPF to reduce flooding in *Mobile Ad Hoc Networks* (MANET)[1], demand circuits designed to support on-demand links in wide-area networks[2], and OSPF stub router advertisements designed to support large-scale *hub and spoke* networks[3] are considered in a typical data center network design to show how these sorts of protocol improvements could affect the scaling of data center environments.

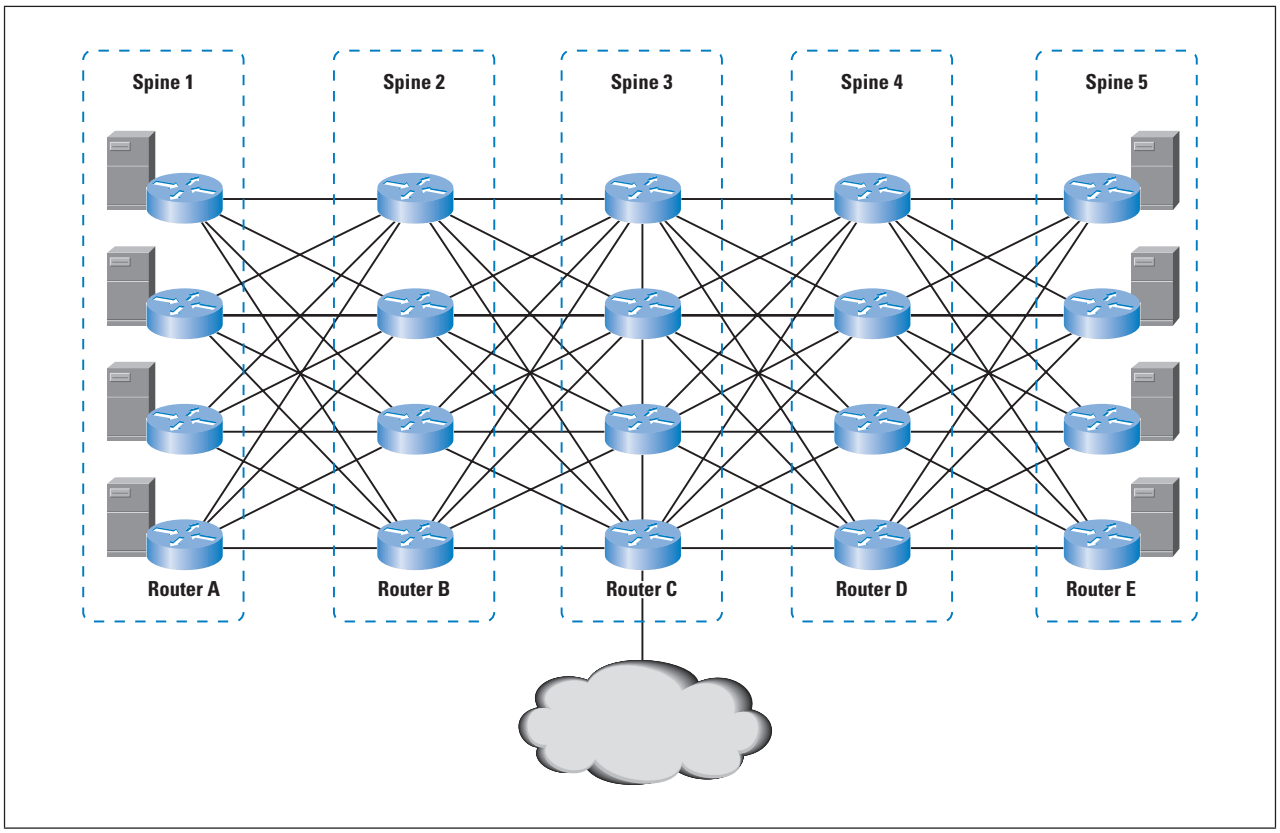
Each of the improvements examined has the advantage of being available in shipping code from at least one major vendor. All of them have been deployed and tested in real-world networks, and have proven effective for solving the problems they were originally designed to address. Note, as well, that OSPF is used throughout this article, but each of these improvements is also applicable to *Intermediate System-to-Intermediate System* (IS-IS), or any other link-state protocol.

Defining the Problem

Figure 1 illustrates a small Clos[0] fabric, what might be a piece of a much larger network design. Although full-mesh fabrics have fallen out of favor with data center designers, Clos and other styles of fabrics are in widespread use. A Clos fabric configured with edge-to-edge Layer 3 routing has three easily identifiable problems.

The flooding rate is the first problem a link-state protocol used in this configuration must deal with. Router B (and the other routers in spine 2), for instance, will receive four type 1 *Link State Advertisements* (LSAs) from the four routers in spine 1. Each of the routers in spine 2 will reflood each of these type 1 LSAs into spine 3, so the other routers in spines 3, 4, and 5 will each receive four copies of each type 1 LSA originated by routers in spine 1, a total of 16 type 1 LSAs in all.

Figure 1: A Clos Fabric with Layer 3 to the Top of Rack



To make matters worse, OSPF is designed to time out every LSA originated in the network once every 20 to 30 minutes. This feature was originally put in OSPF to provide for recovery from bit and other transmission errors in older transport mechanisms with little or no error correction. So a router in spine 5 will receive 16 copies of each type 1 LSA generated by routers in spine 1 every 20 minutes. A single link failure and recovery can also cause massive reflooding. The process of bringing the OSPF adjacency back into full operation requires a complete exchange of local link-state databases. If the link between router A and router B fails and then is recovered, the entire database must be transferred between the two routers, even though router B clearly has a complete copy of the database from other sources.

Finally, the design of this network produces some challenges for the *Shortest Path First* (SPF) algorithm, which link-state protocols use to determine the best path to each reachable destination in the network. Every router in spine 1 appears to be a transit path to every other destination in the network. This outcome might not be the intent of the network designer, but SPF calculations deal with available paths, not intent.

This set of problems has typically swayed network designers away from using link-state protocols in such large-scale environments. Some large cloud service providers use the *Border Gateway Protocol* (BGP) (see [4]), with each spine being a separate Autonomous System, so they can provide scalable Layer 3 connectivity edge-to-edge in large Clos network topologies. Others have opted for simple controls, such as removing all control-plane protocols and relying on reverse-path-forwarding filters to prevent loops.

The modifications to OSPF discussed in this article, however, make it possible for a link-state protocol to not only scale in this type of environment, but also to be a better choice.

Reducing Flooding Through MANET Extensions

MANET networks are designed to be "throw and forget;" a collection of devices is deployed into a quickly fluid situation on the ground, where they connect over short- and long-haul wireless links, and "just work." One of the primary scaling (and operational) factors in these environments is an absolute reduction of link usage wherever possible, including for the control plane.

The "Extensions to OSPF to Support Mobile Ad Hoc Networking,"[1] were developed to reduce flooding in single-area OSPF networks to the minimal necessary, while providing fast recovery and guaranteed delivery of control-plane information. The idea revolves around the concept of an overlapping relay, which reduces flooding by accounting for the network topology, specifically groups of overlapping nodes.

Let's examine the process from the perspective of router A shown in Figure 2.

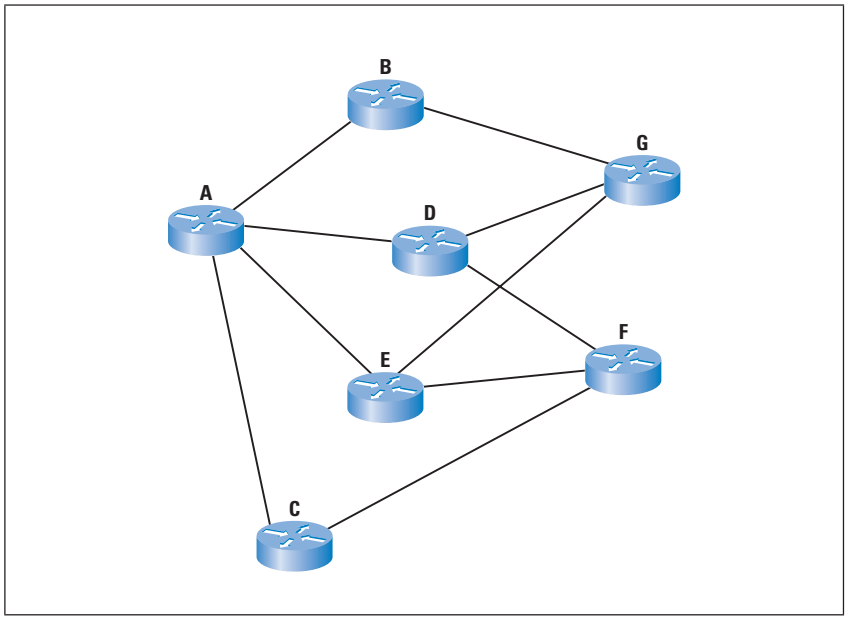


Figure 2: Ad Hoc Extensions to OSPF

Router A begins the process by not only discovering that it is connected to routers B, C, D, and E, but also that its *two-hop neighborhood* contains routers F and G. By examining the list of two-hop neighbors, and the directly connected neighbors that can reach each of those two-hop neighbors, router A can determine that if router D refloods any LSAs router A floods, every router in the network will receive the changes. Given this information, router A notifies routers B, C, and E to delay the reflooding of any LSAs received from router A itself.

When router A floods an LSA, router D will reflood the LSA to routers F and G, which will then acknowledge receiving the LSA to routers B, C, D, and E. On receiving this acknowledgement, routers B, C, and E will remove the changed LSA from their reflood lists.

Routers F and G, then, will receive only one copy of the changed LSA, rather than four.

Applying this process to the Clos design in Figure 1 and using this extension would dramatically reduce the number of LSAs flooded through the network in the case of a topology change. If router A, for instance, flooded a new type 1 LSA, the routers in spine 2 would each receive one copy. The routers in spines 3, 4, and 5 would also receive only one copy each, rather than 4 or 16.

Reducing Flooding Through Demand Circuits

Network engineers have long had to consider links that are connected only when traffic is flowing in their network and protocol designs. Dial-up links, for instance, or dynamically configured *IP Security* (IPsec) tunnels, have always been a part of the networking landscape. Part of the problem with such links is that the network needs to draw traffic to destinations reachable through the link even though the link is not currently operational.

With protocols that rely on neighbor adjacencies to maintain database freshness, such as OSPF, links that can be disconnected in the control plane and yet still remain valid in the data plane pose a unique set of difficulties. The link must appear to be available in the network topology even when it is, in fact, not available.

To overcome this challenge, the OSPF working group in the IETF extended the protocol to support demand links. Rather than attacking the problem at the adjacency level, OSPF attacks the problem at the database level. Any LSA learned over a link configured as a demand link is marked with the *Do Not Age* (DNA) bit; such LSAs are exempt from the normal aging process, causing LSAs to be removed from the link-state database periodically.

How does this situation relate to scaling OSPF in data center network design?

Every 20 minutes or so, an OSPF implementation will time out all the locally generated LSAs, replacing them with newly generated (and identical) LSAs. These newly generated LSAs will be flooded throughout the network, replacing the timed-out copy of the LSA throughout the network. In a data center network, these refloods are simply redundant; there is no reason to refresh the entire link-state database periodically.

To reduce flooding, then, data center network designers can configure all the links in the data center as demand circuits. Although these links are, in reality, always available, configuring them as demand circuits causes the DNA bit to be set on all the LSAs generated in the network. This process, in turn, disables periodic reflooding of this information, reducing control-plane overhead.

Reducing Control-Plane Overhead by Incremental Database Synchronization

When a link fails and then recovers, the OSPF protocol specifies a lengthy procedure through which the two newly adjacent OSPF processes must pass to ensure their databases are exactly synchronized. In the case of data center networks, however, there is little likelihood that a single link failure (or even multiple link failures) will cause two adjacent OSPF processes to have desynchronized databases.

For instance, in Figure 1, if the link between routers A and B fails, routers A and B will still receive any and all link-state database updates from some other neighbor they are still fully adjacent with. When the link between routers A and B is restored, there is little reason for routers A and B to exchange their entire databases again.

This situation is addressed through another extension suggested through the MANET extensions to OSPF called *Unsynchronized Adjacencies*. Rather than sending an entire copy of the database on restart and waiting until this exchange is complete to begin forwarding traffic on link recovery, this extension states that OSPF processes do not need to synchronize their databases if they are already synchronized with other nodes in the network. If needed, the adjacency can be synchronized out of band at a later time.

The application of the MANET OSPF extensions[1] to a data center network means links can be pressed into service very quickly on recovery, and it provides a reduction in the amount of control-plane traffic required for OSPF to recover.

Reducing Processing Overhead Through Stub Routers

The SPF calculation that link-state protocols use to determine the best path to any given destination in the network treats all nodes and all edges on the graph as equal. Returning to Figure 2, router B will calculate a path through router A to routers D, E, and C, even if router A is not designed to be a transit node in the network. This failure to differentiate between transit and nontransit nodes in the network graph increases the number of paths SPF must explore when calculating the shortest-path tree to all reachable destinations.

Although modern implementations of SPF do not suffer from problems with calculation overhead or processor usage, in large-scale environments, such as a data center network with tens of thousands of nodes in the shortest-path tree and virtualization requirements that cause a single node to run SPF hundreds or thousands of times, small savings in processing power can add up.

The "OSPF Stub Router Advertisement"[3] mechanism allows network administrators to mark an OSPF router as nontransit in the shortest-path tree. This mechanism would, for instance, prevent router A in Figure 1 from being considered a transit path between router B and some other router in spine 2. You would normally want to consider this option only for any actual edge routers in the network, such as the top-of-rack routers shown here. Preventing these routers from being used for transit can reduce the amount of redundancy available in the network, and, if used anyplace other than a true edge, prevent the network from fully forming a shortest-path tree.

Advantages and Disadvantages of Link-State Protocols in the Data Center

Beyond the obvious concerns of convergence speed and simplicity, there is one other advantage to using a link-state protocol in data center designs: equal-cost load sharing. OSPF and IS-IS both load share across all available equal-cost links automatically (subject to the limitations of the forwarding table in any given implementation). No complex extensions (such as [5]), are required to enable load sharing across multiple paths.

One potential downside to using a link-state protocol in a data center environment must be mentioned, however—although BGP allows route filtering at any point in the network (because it is a path vector–based protocol)—link-state protocols can filter or aggregate reachability information only at flooding domain boundaries. This limitation makes it more difficult to manage traffic flows through a data center network using OSPF or IS-IS to advertise routing information. This problem has possible solutions, but this area is one of future, rather than current, work.

Conclusion

Many improvements have been made to link-state protocols over the years to improve their performance in specific situations, such as MANETs, and when interacting with dynamically created links or circuits. Many of these improvements are already deployed and tested in real network environments, so using them in a data center environment is a matter of application rather than new work. All of these improvements are applicable to link-state control planes used for Layer 2 forwarding, as well as Layer 3 forwarding, and they are applicable to OSPF and IS-IS.

These improvements, when properly applied, can make link-state protocols a viable choice for use in large-scale, strongly meshed data center networks.

References

[0] **http://en.wikipedia.org/wiki/Clos_network**

[1] Roy, A., "Extensions to OSPF to Support Mobile Ad Hoc Networking," RFC 5820, March 2010.

[2] Abhay Roy and Sira Panduranga Rao, "Detecting Inactive Neighbors over OSPF Demand Circuits (DC)," RFC 3883, October 2004.

[3] Alvaro Retana, Danny McPherson, Russ White, Alex D. Zinin, and Liem Nguyen, "OSPF Stub Router Advertisement," RFC 3137, June 2001.

[4] Petr Lapukhov and Ariff Premji, "Using BGP for routing in large-scale data centers," Internet Draft, work in progress, April 2013, **draft-lapukhov-bgp-routing-large-dc-04**

[5] Daniel Walton, John Scudder, Enke Chen, and Alvaro Retana, "Advertisement of Multiple Paths in BGP," Internet Draft, work in progress, December 2012, **draft-ietf-idr-add-paths-08**

[6] T. Sridhar, "Cloud Computing—A Primer, Part 1: Models and Technologies," *The Internet Protocol Journal,* Volume 12, No. 3, September 2009.

[7] T. Sridhar, "Cloud Computing—A Primer, Part 2: Infrastructure and Implementation Topics," *The Internet Protocol Journal,* Volume 12, No. 4, December 2009.

RUSS WHITE is a Principle Research Engineer at Verisign, where he works on the intersection of naming and routing. In the more than 20 years since he first began working in computer networking, he has co-authored 8 technical books and more than 30 patents; he has participated in the writing, editing, and guiding of numerous Internet Standards, and he has written a fiction novel. He is currently working on *The Art of Network Architecture,* to be published by Cisco Press in 2013. Russ splits his time between the Raleigh, N.C., area and Oak Island, N.C.; he teaches in a local homeschool coop and attends Shepherds Theological Seminary. He is a regular blogger and guest on the *Packet Pushers* podcast.
E-mail: **riwhite@verisign.com**

ALVARO RETANA is a Distinguished Engineer in Cisco Technical Services, where he works on strategic customer enablement. Alvaro is widely recognized for his expertise in routing protocols and network design and architecture; he has CCIE® and CCDE® certifications, and he is one of a handful of people who have achieved the CCAR® certification. Alvaro is an active participant in the IETF, where he co-chairs the Routing Area Working Group (rtgwg), is a member of the Routing Area Directorate, and has authored several RFCs on routing technology. Alvaro has published 4 technical books and has been awarded more than 35 patents by the U.S. Patent and Trademark Office. His current interests include software-defined networking, energy efficiency, infrastructure security, routing protocols, and other related topics. E-mail: **aretana@cisco.com**

Letter to the Editor

Dear Editor,

I enjoyed reading the article on "Address Authentication" in the March 2013 edition of *The Internet Protocol Journal* (Volume 16, No. 1), but I couldn't help thinking to myself how the widespread adoption of the use of *IPv6 Privacy Addresses* (RFC 4941) would affect some of the assertions in the article about the relative merits of using IPv6 addresses for authentication. With both Microsoft and Apple operating systems now implementing IPv6 Privacy Addresses, it is now effectively impossible for any user authentication service to assume that a presented IPv6 address is going to remain constant over time. It is probably safer to assume that such IPv6 addresses are in fact not constant at all, and not to use them in any context of authentication. Given that the widespread use of NATs in IPv4 leads one to the same basic conclusion about using IPv4 addresses for authentication, isn't the best advice these days to avoid "Address Authentication" as it is applied to Internet end users?

Regards,

—*Geoff Huston*
gih@apnic.net

The author responds:

I agree with Geoff's comments. My article explores the idea that IPv6 may be more "trustworthy," but it concludes by recommending against using any IP address as a form of authentication.

IPv4 addresses will be far less "trustworthy" with the introduction of *Carrier-Grade NATs* or *Large-Scale NATs*. We will not be able to trust IPv6 addresses if the interface identifier changes frequently. My expectation is that most enterprises would prefer *Dynamic Host Configuration Protocol for IPv6* (DHCPv6) with randomized interface identifiers, but most broadband Internet access subscribers will use a *Customer Premises Equipment* (CPE) that uses *Stateless Address Autoconfiguration* (SLAAC) and Stateless DHCPv6. IPv6 offers the ability to perform traceback to the /64 subnet level. That feature is only slightly better than IPv4 traceback.

—*Scott Hogg*
scott@hoggnet.com

Book Review

Network Geeks

Network Geeks: How They Built the Internet, by Brian E. Carpenter, Copernicus Books, ISBN 978-1-4471-5024-4, 2013.

The movie opens on a familiar scene, toward the end of a congenial dinner party at the plush home of an august personage. Conversation has been casual and wide-ranging. The group retires to the library for brandy, cigars, and more conversation. Because you are new to your profession and the august personage was involved in its early years, you ask him what it was like. As he begins his recitation, the scene fades to an earlier time... "My great-grandfather, John Winnard, was born in Wigan..."

Such is the style of Brian Carpenter's book, *Network Geeks: How They Built the Internet*. Although indeed many other people are cited, the book really is Brian's personal memoir, complete with his own photographs. It explores his background and work, providing a fascinating travelogue of one person's arc through recent history. Given the breadth and scale of the 50-year process of invention and development of the global Internet, we need perhaps a thousand more such reminiscences to provide sufficiently rich detail about the many actors and acts that contributed to its success.

Brian's experiences within that global history are certainly worthy of note. His writing paints pictures of places and topics such as the forces and attractions that drew him to computer networking; in those days, it was an outlier technical topic and people often happened into it, rather than setting out with a plan. Indeed, Brian's doctoral work was in computer speech understanding—not networking. However, he has played a key role in many significant Internet activities. His frequent employer, the Swiss CERN[0], was a focal point for much of the early European networking activity—as well as being the birthplace of the World Wide Web—and Brian's various leadership roles in the *Internet Engineering Task Force* (IETF) came at pivotal times. Other popular references to Internet history tend to emphasize its American basis, making Brian's primarily European perspective refreshing and helpful.

The book is short, just 150 pages. Although Brian makes some terse references early in the book, he does not get fully into gear talking about the Internet until a third of the way through it. He started in physics, coming fully to computer science only in graduate school. Over the course of the memoir, we hear quite a bit about his physics work at CERN and elsewhere, as well as his activities with the early European deployment of Internet services, his eventual work with Internet standards, and the like.

The IETF

Brian's reference to his great-grandfather does appear, but not until page 10 in a chapter that extensively details his family history and his own upbringing—how many other books on Internet history are likely to include an inset distinguishing the English Baptist church from the American Southern Baptist? Rather, the book begins with a description of a prototypical IETF plenary session at the thrice-annual standards meeting, and he paints the picture well enough to have prompted a guessing game about the person he was describing. IETF meetings, including the plenaries, have a great deal of audience participation, because these meetings are working meetings, not conferences. I particularly enjoyed Brian's turn of phrase when describing one participant, "...who had given several articulate but incomprehensible arguments at the microphone." Later in the book he also equitably describes a colleague as "a wise leader, decisive or even pig-headed, but willing to listen..."

After its opening sequences, the book follows Brian's life chronology, including extended periods in England, Switzerland, the United States, and New Zealand, most recently landing at the last. His employment has variously been university, research, and corporate, including roles as researcher, manager, chair, and teacher.

This book is a memoir, so Brian casually and regularly moves between discussion of personal and professional developments. From one paragraph to the next, he might describe structural aspects of an Internet organization, insulation of housing in New Zealand, the next effort at particle physics, optimizing travel when flying out of southeast England, the nature of a computer networking technology, or the personal style of a co-worker.

In particular, this work is not a tutorial on Internet technology or on its invention. Although Brian does discuss many aspects of the technologies, the pedagogy suits an after-dinner evening's reminiscences, not a classroom lecture. Some concepts are explained in great detail, while others are merely cited. For example, his early discussion of computer networking references the fact that it enables mesh topologies, in contrast to then-common star configurations, but he doesn't give much sense of what "mesh" means in technical terms. Also, the core technology of networking is *packet-switching* and although his discussion on the page after the mesh reference cites queuing theory, he never introduces the motivating design construct of "store and forward."

His discussion of addressing suggests his hardware background, and misses the essence that a name at one level of architecture is often an address at the next level up. So although `www.example.com` is the "name" of a host system attached to the Internet, it has the role of "address" in a URL, because it specifies where to go to resolve the remainder of the URL.

That said, quibbling with such an issue in a tutorial might be reasonable, but it is entirely inappropriate for a memoir. These are Brian's recollections. If they prompt the reader to explore things later, so much the better; but arguing his view will not do. Perhaps reflexively, it is convenient that the Internet makes such exploration quite easy...

NATs

Except that I remain sorry to see that Brian still has such a strikingly purist view about *Network Address Translation* (NAT)[1, 2] devices, which map between internal (private) IP addresses and public ones. The purist view is that they are an abomination that breaks the elegance of the "end-to-end" design principle of the Internet. The principle is powerful, because it tends to greatly simplify the communications infrastructure and greatly enable innovation at the endpoints. The problem is that the real world imposes organizational and operational models that are more complex than easily supported by the basic end-to-end construct, at the least needing to include enterprise-level policies. NATs do cause problems, by replacing one IP address for another, and some mechanisms do cease to work because of these replacements. However, the operational world views NATs as being useful against multiple problems. One is address space constraints, which is the formal justification for creating the mechanism: an enterprise uses far fewer public IP addresses—a reality that is now essential as IPv4 addresses have grown scarce. Another justification is the misguided view that they improve enterprise security, and the other is the legitimate view that they simplify enterprise network administration. After more than 20 years of extensive deployment, these devices might be expected to have become tolerable to a pragmatist, possibly even forcing consideration of a more elaborate architectural model for the Internet. Yet Brian suffers no such weakness; NATs are evil.

One of the technical points that intrigued me was Brian's repeated discussion of the *Remote Procedure Call* (RPC). This mechanism makes network interaction for an application look like little more than a subroutine invocation. It was hoped that it would greatly simplify network-oriented programming and make it accessible to any software developer, rather than requiring the developer to have a deep understanding of networking interfaces and dynamics. Brian cites the mechanism as having been "invented by the ARPANET community in the mid-1970s..." and used at CERN in a programming language shortly after that. But my own recollection is of hearing a Xerox *Palo Alto Research Center* (PARC) manager in 1980 proudly announce that one of his summer interns had just developed the idea. Indeed, Wikipedia credits the late Bruce Jay Nelson, a Carnegie Mellon University graduate student who was working at PARC.[3]

And that is the essence of a memoir. It is the remembrances of the speaker, not the formal work of a historian or journalist. It is not the diligent unfolding of a researched history, such as in *Where Wizards Stay up Late*[4], nor the tourist approach of *Exploring the Internet: A Technical Travelogue*[5] that seeks to name every possible person active at the time—although Brian does sometimes invoke that latter template. Instead it shares one person's sense of what happened—what he remembers doing and seeing.

Railing against architectural biases or historical nuances is essential when evaluating formal professional writing, and we do need such judicious efforts to capture the history of the Internet. But had Brian sought to produce such a tome, it would not have been as rich or as personal.

References

[0] European Organization for Nuclear Research, Geneva, **http://home.web.cern.ch/**

[1] Network Address Translation, **https://en.wikipedia.org/wiki/Network_address_translation**

[2] Geoff Huston, "Anatomy: A Look inside Network Address Translators," *The Internet Protocol Journal*, Volume 7, No. 3, September 2004.

[3] Remote Procedure Call, **http://en.wikipedia.org/wiki/Remote_procedure_call**

[4] Katie Hafner and Matthew Lyon, *Where Wizards Stay Up Late: The Origins of the Internet*, Simon & Schuster, ISBN 0-684-81201-0, 1996.

[5] Carl Malamud, *Exploring the Internet: A Technical Travelogue*, Prentice-Hall, Inc., ISBN 0-13-296898-3 1992/1997, **http://museum.media.org/eti/**

—Dave Crocker,
dcrocker@bbiw.net

Fragments

Number of IPv6-Connected Internet Users Doubles

The *Internet Society* (ISOC) recently reported that the number of IPv6-connected users has doubled since *World IPv6 Launch* began on June 6, 2012, when thousands of *Internet Service Providers* (ISPs), home networking equipment manufacturers, and Web companies around the world came together to permanently enable the next generation of *Internet Protocol Version 6* (IPv6) for their products and services. This marks the third straight year IPv6 use on the global Internet has doubled. If current trends continue, more than half of Internet users around the world will be IPv6-connected in less than 6 years.

"The year since World IPv6 Launch began has cemented what we know will be an increasing reality on the Internet: IPv6 is ready for business," said Leslie Daigle, the Internet Society's Chief Internet Technology Officer. "Forward-looking network operators are successfully using IPv6 to reduce their dependency on expensive, complex network address translation systems (*Carrier Grade Network Address Translators*) to deal with a shortage of IPv4 addresses. Leaders of organizations that aspire to reach all Internet users must accelerate their IPv6 deployment plans now, or lose an important competitive edge."

As IPv6 adoption continues to grow, members of the worldwide Internet community are contributing to its deployment. Statistics reported by World IPv6 Launch participants underscore the increasing deployment of IPv6 worldwide:

- Google reports the number of visitors to its sites using IPv6 has more than doubled in the past year.
- The number of networks that have deployed IPv6 continues to grow, with more than 100 worldwide reporting significant IPv6 traffic.
- Australian ISP Internode reports that 10 percent of its customers now use IPv6 to access the Internet.
- Akamai reports that it is currently delivering approximately 10 billion requests per day over IPv6, which represents a 250 percent growth rate since June of last year.
- KDDI measurement shows that the number of IPv6 users of KDDI has doubled and that IPv6 traffic has increased approximately three times from last year.

World IPv6 Launch participants have worked together to help drive adoption, leading to the creation of *World IPv6 Day* in 2011, in which hundreds of websites joined together for a successful global 24-hour test flight of IPv6.

This was followed by World IPv6 Launch in 2012, in which more than a thousand participants permanently enabled IPv6 for their products and services, including four of the most visited websites: Google, Facebook, YouTube, and Yahoo!.

As a platform for innovation and economic development, the Internet plays a critical role in the daily lives of billions. This momentum has not slowed—IPv6 adoption continues to skyrocket, fast establishing itself as the "new normal" and a must-have for any business with an eye towards the future.

For more information about companies that have deployed IPv6, as well as links to useful information for users and how other companies can participate in the continued deployment of IPv6, please visit: **http://www.worldipv6launch.org**

IPv4 has approximately four billion IP addresses (the sequence of numbers assigned to each Internet-connected device). The explosion in the number of people, devices, and web services on the Internet means that IPv4 is running out of space. IPv6, the next-generation Internet protocol which provides more than 340 trillion, trillion, trillion addresses, will connect the billions of people not connected today and will help ensure the Internet can continue its current growth rate indefinitely.

The Internet Society is the trusted independent source for Internet information and thought leadership from around the world. With its principled vision and substantial technological foundation, the Internet Society promotes open dialogue on Internet policy, technology, and future development among users, companies, governments, and other organizations. Working with its members and Chapters around the world, the Internet Society enables the continued evolution and growth of the Internet for everyone. For more information, visit: **http://www.internetsociety.org**

RIPE NCC Report on ITU WTPF-13

The RIPE NCC has published a report on the recent *ITU World Telecommunications/ICT Policy Forum* (WTPF-13). The report is available from the following URL:

https://www.ripe.net/internet-coordination/news/ripe-ncc-report-on-the-itu-wtpf-13

Any comments or questions are welcome on the RIPE Cooperation Working Group mailing list:

https://www.ripe.net/ripe/mail/wg-lists/cooperation

Google.org Awards Grant to ISOC to Advance IXPs in Emerging Markets

The *Internet Society* (ISOC) recently announced that it has been awarded a grant by Google.org to extend its *Internet Exchange Point* (IXP) activities in emerging markets. The grant will build on the Internet Society's previous efforts and will establish a methodology to assess IXPs, provide training for people to operate the IXPs, and build a more robust local Internet infrastructure in emerging markets.

IXPs play an important role in Internet infrastructure that allows *Internet Service Providers* (ISPs) and other network operators to exchange traffic locally and more cost effectively, which can help lower end-user costs, speed-up transmissions, increase Internet performance, and decrease international Internet connectivity costs. The Internet Society and Internet technical experts have been working for several years to bring IXPs to emerging markets. These efforts have resulted in locally trained experts and facilitated the development of local and regional technical infrastructures. An additional benefit of IXP development is the expansion of community governance models as well as building local Internet expertise.

Google.org, a team within Google focused on social impact, develops and supports technology solutions that can address global challenges, such as expanding Internet access to more of the world's seven billion people.

"The Internet Society has proved to be one of the most effective institutions in the Internet community," said Vint Cerf, vice president and Chief Internet Evangelist at Google. "I am confident that they will apply their grant wisely to extend their work to increase Internet access for everyone, including those in emerging markets."

Lynn St. Amour, President and CEO of the Internet Society, stated, "We are very excited to receive this grant from Google.org. With support to extend our IXP development and improvement projects, we can more quickly bring core Internet infrastructure to underserved countries and assist in building key human and governance capabilities. We will also be able to extend the Internet Society's mission to ensure the open development, evolution, and use of the Internet for the benefit of people everywhere. We look forward to working with Google.org, and we are committed to collaborating with Internet community partners around the world on this important project."

IPJ Subscription FAQ

What is my "Subscription ID" for The Internet Protocol Journal (IPJ) and where do I find it?

Your Subscription ID is a unique combination of letters and numbers used to locate your subscription in our database. It is printed on the back of your IPJ issue or on the envelope. You will also find information about your subscription expiration date near your Subscription ID. Here is an example:

How do I renew or update my subscription?

From the IPJ homepage (**www.cisco.com/ipj**) click "Subscriber Service" and then enter your Subscription ID and your e-mail address in the boxes. After you click "Login" the system will send you an e-mail message with a unique URL that allows access to your subscription record. You can then update your postal and e-mail details, change delivery options, and of course *renew* your subscription.

What will you use my e-mail address and postal address for?

This information is used *only* to communicate with you regarding your subscription. You will receive renewal reminders as well as other information about your subscription. We will never use your address for any form of marketing or unsolicited e-mail.

I didn't receive the special URL that allows me to renew or update my Subscription. Why?

This is likely due to some form of spam filtering. Just send an e-mail message to **ipj@cisco.com** with your Subscription ID and any necessary changes and we will make the changes for you.

Do I need my Subscription ID to read IPJ online? What is my username and password?

Your Subscription ID is used *only* for access to your subscription record. No username or password is required to read IPJ. All back issues are available for online browsing or for download at **www.cisco.com/ipj**

I can't find my Subscription ID and I have since changed e-mail address anyway; what do I do now?

Just send a message to **ipj@cisco.com** and we will take care of it for you.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com