IPv6

IPv6 History, Principles, Addressing, Plug and Play,
Routing, Facts and Myths,
Migration & Transition Ideas

Agenda

- **History**
  - **The initial problem in the 1990s**
  - The first decade (decision and prototyping)
  - The second decade (maturity level)
- **IPv6**
- **ICMPv6 and Plug&Play**
- **Routing**
- **Transition**
- **Miscellaneous**

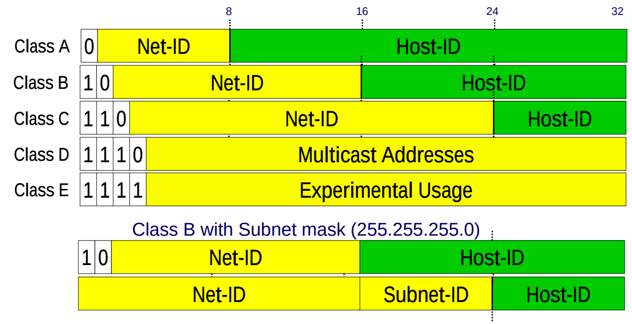
Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

The Internet at 1990 (1)

- **Only classful IP addresses (32 bit) were used**
  - A, B, C (unicast), D (multicast), E (experimental)
  - There are only
    - 126 class A nets with 16.777.214 hosts
    - 16.384 class B nets with 65.534 hosts
    - 2.097.152 class C nets with 254 hosts
- **In order to communicate over the Internet**
  - You have to use an official IP address range assigned by your ISP or Local/**R**egional **I**nternet **R**egistry (RIR) for all your IP systems
  - This could be only a class A or B or C address range
- **Subnetting was used**
  - To divide a given class A, B or C address into subnets in order to structure your local IP network with IP routers

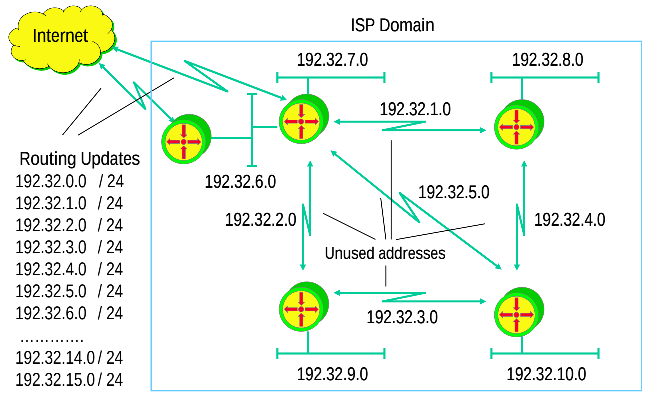
Review: IP Address Classes / Subnetting



First octet rule:

- class A range: 1-126
- class B range: 128-191
- class C range: 192-223
- class D range: 224-239

Review: ISP Routing Updates with Classful Behavior



The Internet at 1990 (2)

- **Address classes were inflexible**
  - IP address with 2^{32} bits
    - 4.294.967.296 IP hosts could be distinguished in theory
      - minus Class D and E (536.870.912)
      - minus Net 0 and 127 (33.554.432)
      - minus RFC 1918 (17.891.328)
    - result: 3.706.650.624 usable addresses
  - Class B to large and Class C to small for many companies
  - Constant subnet masks wasted a lot of (unused) addresses
- **Exponential growth of Internet connected devices**
  - Prediction of the exhaustion of IPv4 addresses by 2005-2011
- **But the real problem was**
  - Class B addresses were nearly exhausted
    - Prediction of the exhaustion of IPv4 class B addresses by 1994
  - Routing table explosion at Internet core routers
    - Caused by assigning multiple class C addresses for bigger companies

© 2012, D.I. Lindner

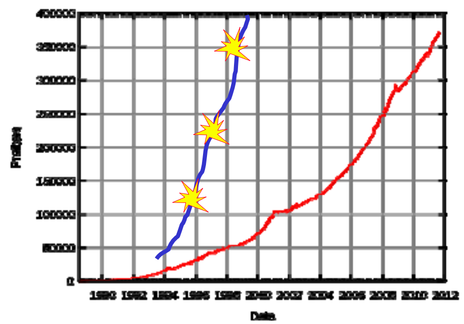
Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

BGP Routing Table Growth – Internet Core

Growth without CIDR + Private Addresses + NAT)

Actual Growth Source: http://en.wikipedia.org/wiki/Border_Gateway_Protocol



The Way to IPng / IPv6 History (2)

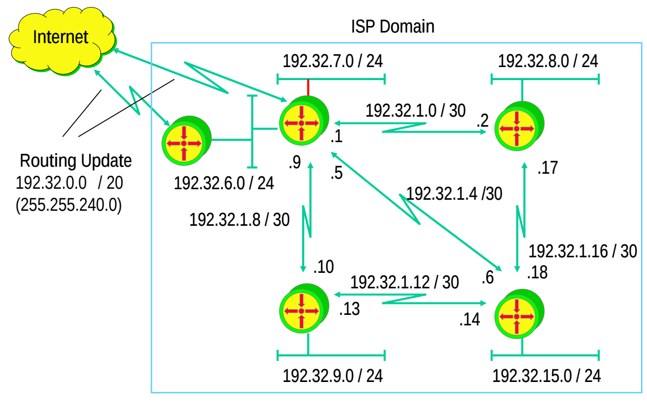
- **The Need for a new IP was identified**
- **November 1991**
  - Routing and Addressing (ROAD) workgroup formed
- **March 1992**
  - ROAD report
    - Do CIDR (Classless Interdomain Routing)
    - Issue a call for **IPng** (next generation) proposals
- **CIDR, private IP addressing and network address translation (NAT) as temporary (short-term) solution**
  - RFC 1518 An Architecture for IP Address Allocation with CIDR (Status: HISTORIC)
  - RFC 1519 Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy (obsoleted by RFC 4632) (Status: PROPOSED STANDARD)
  - **RFC 4632** Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan (Obsoletes RFC1519) BCP0122)
    - http://www.potaroo.net/ (BGP table, CIDR report – Geoff Huston)
  - RFC 1631 “The IP Network Address Translator (NAT)” may 1994 (obsoleted by RFC 3022) (Status: INFORMATIONAL)
  - **RFC 3022** Traditional IP Network Address Translator (Traditional NAT) (Obsoletes RFC1631) (Status: INFORMATIONAL)
  - RFC 1918 “Address Allocation for Private Internets” February 1996 (Status: BCP0005)

© 2012, D.I. Lindner

Classless Inter-Domain Routing (CIDR)

- Address assignment and aggregation (= route summarization) strategy for blocks of networks with contiguous addresses
- Creative IP address allocation
  - addressing plan for class C addresses by continents
  - provider based addressing strategy
- Classless routing (prefix, length)
  - Classful routing protocols advertize only net-id (= prefix) of networks but no subnetmask
- Usage of VLSM (**V**ariable **L**ength **S**ubnet **M**ask)
  - For better usage of address space
- Supernetting of class C network numbers blocks

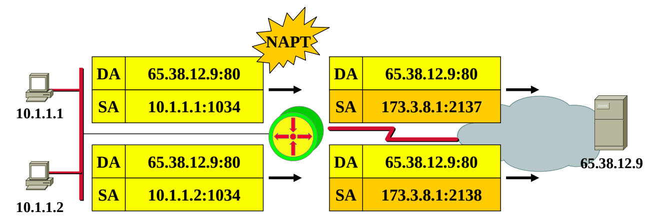
ISP Routing Updates with Classless Behavior (Route Summarization with Supernetting), VLSM



Private Address Range – NAT

- **Three blocks of address ranges are reserved for addressing of private networks (RFC 1918)**
  - 10.0.0.0 - 10.255.255.255 (10/8 prefix)
  - 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
  - 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)
- **Translation between private addresses and globally unique addresses**
  - NAT (Network Address Translation)
  - Address saving
    - Actually achieved by many-to-one translation
      - NAPT (Network Address Port Translation)
      - PAT (Port Address Translation
    - Usable in context of TCP and UDP sessions

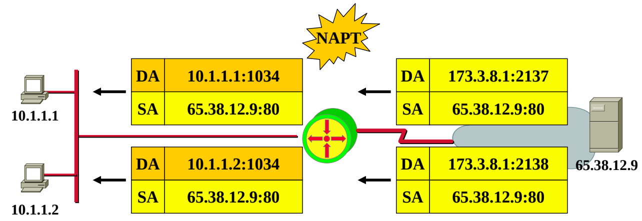
Review: NAPT (1)



Prot.	Local	Global
TCP	10.1.1.1:1034	173.3.8.1:2137
TCP	10.1.1.2:1034	173.3.8.1:2138

Extended NAT Translation Table

Review: NAPT (2)



Prot.	Local	Global
TCP	10.1.1.1:1034	173.3.8.1:2137
TCP	10.1.1.2:1034	173.3.8.1:2138

Extended NAT Translation Table

Some NAT Drawbacks

- **No end-to-end IP anymore possible**
  - Sufficient for client/server model
  - Not sufficient for peer-to-peer networking
- **Network component (NAT device)**
  - Needs to maintain state of connections in case of dynamic mapping (PAT, NAPT)
- **Fast rerouting difficult if NAT router fails**
- **Makes development of new application difficult (“NAT friendly?”)**
- **Security (end-to-end)**
  - IPsec needs NAT-Transversal (encapsulating into UDP or TCP)
- **Manageability**

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Agenda

- **History**
  - The initial problem in the 1990s
  - **The first decade (decision and prototyping)**
  - The second decade (maturity level)
- **IPv6**
- **ICMPv6 and Plug&Play**
- **Routing**
- **Transition**
- **Miscellaneous**

The Way to IPng / IPv6 History (2)

- **July 1992**
  - IAB issues “IP version 7” (TP/IX, RFC 1475)
    - Intention for new IP and TCP, 64 bit addresses, admin domain number, forward route identifier (flow), new routing protocol RAP
  - IETF issues the call for IPng proposals
- **July 1993**
  - IPv7 refused by IESG
  - Short-term solutions postpones exhaustion of IP addresses and gives enough time for development of new IP
  - New IP should not covering address issues only but also other known weaknesses of protocol suite
    - Security (authentication, privacy, integrity)
    - Auto-configuration (plug and play)
    - Support for High speed networks
    - Quality of service (QoS)
    - Mobility
    - Real time traffic and multimedia
- **August 1993**
  - IETF area formed to consolidate IPng activity
    - Allison Markin and Scott Bradner area co-directors
- **December 1993**
  - RFC 1550 “IP: Next Generation (IPng) White Paper Solicitation”
    - Input and answers: RFC 1667-1680, 1682/83, 1686-88, 1705, 1707, 1710, 1715

The Way to IPng / IPv6 History (3)

- **Three proposals**
  - CATNIP
    - Common Architecture for next-generation IP (RFC 1707)
    - Common ground between Internet, OSI and Novell protocols
    - Developed from TP/IX working group
    - Cache handles
  - SIPP
    - Simple Internet Protocol Plus (RFC 1710)
    - Complete new version of IP (merge of SIP (Simple IP) and PIP)
    - 64 bit addresses
  - TUBA
    - **T**CP and **U**DP with **B**ig-**A**ddresses (RFC 1347, 1561)
    - TCP/UDP over CNLP-Addressed Networks
    - Migration to OSI NSAP address space (20 byte addresses)
    - Replacement of IP by CNLP

The Way to IPng / IPv6 History (4)

- **July 1994**
  - After review of proposals a recommendation was given for next generation IP by IPng area co-directors
- **October 1994**
  - Recommendation approved by IESG
- **December 1994 / January 1995**
  - RFC 1726
    - Technical criteria for IPng
      - At least 10^9 networks , 10^{12} end-systems
      - **Datagram service**, **conservative routing**, topologically flexible
      - High performance, transition plan from IPv4
      - Robust service, media independent
      - **Auto-configuration**, secure operation, globally unique names
      - Access to standards, **extensible**, include control protocol
      - Support of **mobility**, of multicasting, of service classes and of private networks (tunneling)
  - RFC 1752
    - Merging of proposals and revised proposal based on SIPP
- **April 1995**
  - Base documents ready for proposed standard
  - Decision to give IPng the version number 6 (IPv6)

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

The Way to IPng / IPv6 History (5)

- **December 1995:**
  - RFC 1825 Security Architecture for the Internet Protocol (Obsoleted by RFC2401)
  - RFC 1826 IP Authentication Header (Obsoleted by RFC2402) **IPsec 0.8**
  - RFC 1827 IP Encapsulating Security Payload (ESP) (Obsoleted by RFC2406)
  - **RFC 1881** IPv6 Address Allocation Management (Status: INFORMATIONAL)
  - RFC 1883 Internet Protocol, Version 6 (IPv6) Specification (Obsoleted by RFC2460) **IPv6 0.9**
  - RFC 1884 IP Version 6 Addressing Architecture (Obsoleted by RFC2373)
  - RFC 1885 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) (Obsoleted by RFC2463) **ICMPv6 0.8**
  - RFC 1886 DNS Extensions to support IP version 6 (Obsoleted by RFC3596)
  - **RFC 1887** An Architecture for IPv6 Unicast Address Allocation (Status: INFORMATIONAL)

The Way to IPng / IPv6 History (6)

- **1996-1998:**
  - RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers (Obsoleted by RFC2893)
  - RFC 1970 Neighbor Discovery for IP Version 6 (IPv6)(Obsoleted by RFC2461) **ND 0.8**
  - RFC 1971 IPv6 Stateless Address Autoconfiguration (Obsoleted by RFC2462) **SLAAC 0.8**
  - RFC 1972 A Method for the Transmission of IPv6 Packets over Ethernet Networks (Obsoleted by RFC2464)
  - **RFC 1981** Path MTU Discovery for IP version 6 (Status: DRAFT STANDARD)
  - RFC 2023 IP Version 6 over PPP (Obsoleted by RFC2472)
  - RFC 2073 An IPv6 Provider-Based Unicast Address Format (Obsoleted by RFC2374)

The Way to IPng / IPv6 History (7)

- **1996-1998 cont.:**
  - **RFC 2080** RIPng for IPv6 (Status: PROPOSED STANDARD)
  - RFC 2133 Basic Socket Interface Extensions for IPv6 (Obsoleted by RFC2553)
  - RFC 2147 TCP and UDP over IPv6 Jumbograms (Obsoleted by RFC2675)
  - **RFC 2185** Routing Aspects of IPv6 Transition (Status: INFORMATIONAL)
  - RFC 2292 Advanced Sockets API for IPv6 (Obsoleted by RFC3542)
  - RFC 2374 An IPv6 Aggregatable Global Unicast Address Format (Obsoletes RFC2073) (Obsoleted by RFC3587)
  - **RFC 2375** IPv6 Multicast Address Assignments (Status: INFORMATIONAL)

© 2012, D.I. Lindner IPv6, v4.7 21

The Way to IPng / IPv6 History (7)

- **1996-1998 cont.:**
  - RFC 2401 Security Architecture for the Internet Protocol (Obsoletes RFC1825) (Obsoleted by RFC4301)
  - RFC 2402 IP Authentication Header (Obsoletes RFC1826) (Obsoleted by RFC4302, RFC4305) — **IPsec 0.9**
  - **RFC 2403** The Use of HMAC-MD5-96 within ESP and AH (Status: PROPOSED STANDARD)
  - **RFC 2404** The Use of HMAC-SHA-1-96 within ESP and AH (Status: PROPOSED STANDARD)
  - **RFC 2405** The ESP DES-CBC Cipher Algorithm With Explicit IV (Status: PROPOSED STANDARD)
  - RFC 2406 IP Encapsulating Security Payload (ESP (Obsoletes RFC1827) (Obsoleted by RFC4303, RFC4305)
  - RFC 2407 The Internet IP Security Domain of Interpretation for ISAKMP (Obsoleted by RFC4306)
  - RFC 2408 Internet Security Association and Key Management Protocol (ISAKMP) (Obsoleted by RFC4306)
  - RFC 2409 The Internet Key Exchange (IKE) (Obsoleted by RFC4306) — **IKEv1 1.0**
  - **RFC 2410** The NULL Encryption Algorithm and Its Use With IPsec (Status: PROPOSED STANDARD)
  - RFC 2411 IP Security Document Roadmap (Obsoleted by RFC6071)
  - **RFC 2412** The OAKLEY Key Determination Protocol (Status: INFORMATIONAL)

© 2012, D.I. Lindner IPv6, v4.7 22

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

The Way to IPng / IPv6 History (8)

- **December 1998:** — **IPv6 1.0**
  - **RFC 2460** Internet Protocol, Version 6 (IPv6) Specification (Obsoletes RFC1883) (Updated by RFC5095, RFC5722, RFC5871) (Status: DRAFT STANDARD)
  - RFC 2461 Neighbor Discovery for IP Version 6 (IPv6 (Obsoletes RFC1970) (Obsoleted by RFC4861) — **ND 0.9**
  - RFC 2462 IPv6 Stateless Address Autoconfiguration (Obsoletes RFC1971) (Obsoleted by RFC4862) — **SLAAC 0.9**
  - RFC 2463 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification (Obsoletes RFC1885) (Obsoleted by RFC4443) — **ICMPv6 0.9**
  - **RFC 2464** Transmission of IPv6 Packets over Ethernet Networks. (Obsoletes RFC1972) (Updated by RFC6085) (Status: PROPOSED STANDARD)
  - RFC 2465 Management Information Base for IP Version 6: Textual Conventions and General Group (Obsoleted by RFC4293)
  - RFC 2466 Management Information Base for IP Version 6: ICMPv6 Group (Obsoleted by RFC4293)
  - RFC 2472 IP Version 6 over PPP. D. Haskin, E. Allen (Obsoletes RFC2023) (Obsoleted by RFC5072, RFC5172)
  - **RFC 2473** Generic Packet Tunneling in IPv6 Specification (Status: PROPOSED STANDARD)
  - **RFC 2474** Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 (Obsoletes RFC1455, RFC1349) (Updated by RFC3168, RFC3260) (Status: PROPOSED STANDARD)

© 2012, D.I. Lindner IPv6, v4.7 23

The Way to IPng / IPv6 History (9)

- **1999:**
  - **RFC 2491** IPv6 over Non-Broadcast Multiple Access (NBMA) networks (Status: PROPOSED STANDARD)
  - **RFC 2492** IPv6 over ATM Networks (Status: PROPOSED STANDARD)
  - **RFC 2529** Transmission of IPv6 over IPv4 Domains without Explicit Tunnels (Status: PROPOSED STANDARD)
  - **RFC 2545** Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing (Status: PROPOSED STANDARD)
  - RFC 2546 6Bone Routing Practice (Obsoleted by RFC2772)
  - RFC 2553 Basic Socket Interface Extensions for IPv6 (Obsoletes RFC2133) (Obsoleted by RFC3493) (Updated by RFC3152)
  - **RFC 2590** Transmission of IPv6 Packets over Frame Relay Networks Specification (Status: PROPOSED STANDARD)
  - **RFC 2675** IPv6 Jumbograms (Obsoletes RFC2147) (Status: PROPOSED STANDARD)
  - **RFC 2710** Multicast Listener Discovery (MLD) for IPv6 (Updated by RFC3590, RFC3810) (Status: PROPOSED STANDARD)
  - **RFC 2711** IPv6 Router Alert Option (Status: PROPOSED STANDARD)
  - RFC 2732 Format for Literal IPv6 Addresses in URL's (Obsoleted by RFC3986)
  - RFC 2740 OSPF for IPv6 (Obsoleted by RFC5340) — **OSPFv3 0.9**

© 2012, D.I. Lindner IPv6, v4.7 24

Summary History of IPv6 in the 1990´s

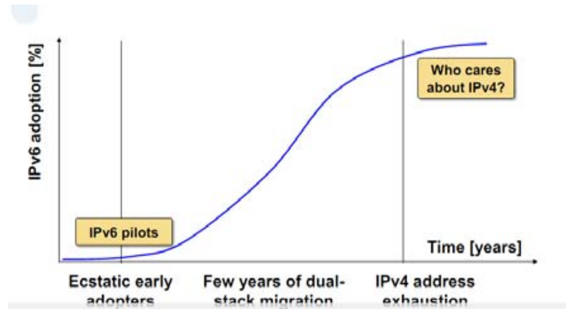
1990 Prediction of the exhaustion of IPv4 Class B by 1994.
1991 ROAD group formed to address routing.
1992 Prediction of the exhaustion of IPv4 addresses by 2005-2011.
1993 IPng Proposals solicitation (RFC 1550).
1994 CATNIP, SIPP, TUBA analyzed. SIPP+ chosen. IPng wg started.
1995 First specification: RFC 1883.
1996 6bone started.
1997 First attempt for provider-based address format.
1998 First IPv6 exchange: 6tap.
1999 Registries assign IPv6 prefixes. IPv6 Forum formed.
2000 Major vendors bundle IPv6 in their mainstream product line.

IPv5

BTW: What happened to IPv5?

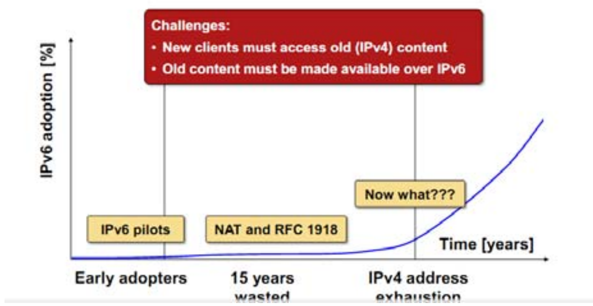
0	IP	March 1977 version (deprecated)
1	IP	January 1978 version (deprecated)
2	IP	February 1978 version A (deprecated)
3	IP	February 1978 version B (deprecated)
4	IPv4	September 1981 version (current widespread)
5	ST	Stream Transport (not a new IP, little use)
6	IPv6	December 1998 version (formerly SIP, SIPP)
7	CATNIP	IPng evaluation (formerly TP/IX; deprecated)
8	Pip	IPng evaluation (deprecated)
9	TUBA	IPng evaluation (deprecated)
10		unassigned
11		unassigned
--		
15		unassigned

Expected IPv6 Transition



Source: Webinar "Enterprise_IPv6_-_The_First_Steps" Ivan Pepelnjak 2011

Real IPv6 Transition



Source: Webinar "Upcoming_Internet_Challanges" Ivan Pepelnjak 2011

Expectations on IPv6 in the 1990s (1)
(compared to what has happened)

- More addresses -> ok
  - But we have waited until IPv4 addresses were exhausted because of private addresses, NAT and CIDR
  - Dual-stack strategy good until now, but makes no sense if there are no new IPv4 addresses are available
- Multihoming -> nok
  - You have to do BGP routing instead of static NAT for small environments
  - Maybe Shim6 or HIP or LISP will help in the future
- Stop explosion of core routing table entries -> nok
  - Multihoming solved only by PI and BGP
  - Provider independent IPv6 addresses will still be a pain for ISPs
- Location / ID separation -> nok
  - Maybe LISP in the future

Expectations on IPv6 in the 1990s (2)
(compared to what has happened)

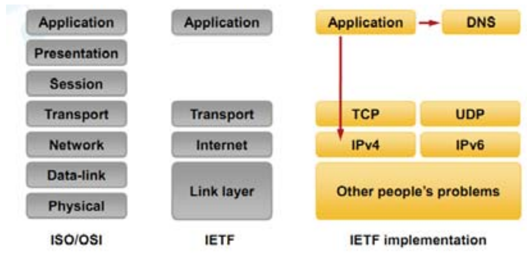
- End-to-End Security -> nok
  - IPsec also usable for IPv4, SSL as alternative widely used
  - IPsec for Site-Site- or Client-Site-VPNs with preshared secrets works fine
  - IPsec for communication with unknown peer with mutual authentication based on PKI still not solved (key distribution issues and trust issues)
- Better QoS -> nok
  - Flow label nice for “Integrated Services Qos Model”
  - But not necessary for “Differentiated Services Qos Model”
  - Traffic class in IPv6 = DSCP in IPv4
- IP Mobility -> nok
  - Also available in IPv4
  - Maybe easier handling in IPv6 in the future because of new IPv6 mobile extension header

© 2012, D.I. Lindner

Expectations on IPv6 in the 1990s (3)
(compared to what has happened)

- Needs no change in the application because UDP and TCP are still there -> nok
  - Protocol stack is broken
    - Session layer is not there
    - Application are established between IP addresses
    - DNS is an optional add-on application
  - Socket (API) is broken
    - You have to asks DNS for giving you IP addresses
    - You will get a list of addresses (IPv4 and IPv6)
    - Applications must be aware of that
    - Applications must be tested if they do address handling correctly

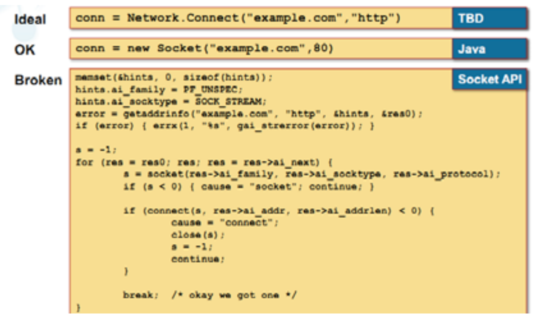
Broken Protocol Stack



Source: Webinar “Upcoming_Internet_Challanges” Ivan Pepelnjak 2011

© 2012, D.I. Lindner

Broken Socket



Source: Webinar "Upcoming_Internet_Challanges" Ivan Pepelnjak 2011 33

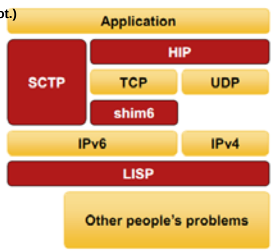
Proposed Fixes for Multihoming, Location/ID Separation

SCTP (Stream Control Transportation Prot.)
- New transport protocol
- Supports multihoming & streams

LISP (Locator/ID Separation Protocol)
- Global directory-driven mGRE/NHRP-like solution

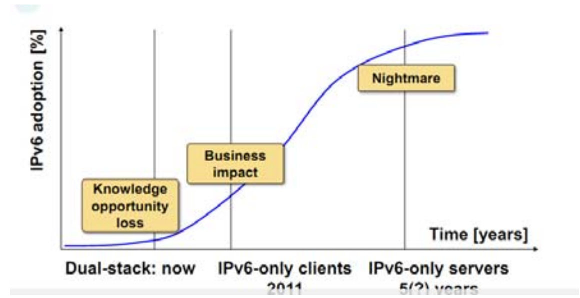
shim6
- Add-on for TCP over IPv6 and UDP streams over IPv6

HIP (Host Identity Protocol)
- Replaces IP address with signed host identifiers



Source: Webinar "Upcoming_Internet_Challanges" Ivan Pepelnjak 2011 34

Expected Results of IPv6 Denial Strategy



Source: Webinar "Enterprise_IPv6_-_The_First_Steps" Ivan Pepelnjak 2011 35

Agenda

- **History**
  - The initial problem in the 1990s
  - The first decade (decision and prototyping)
  - **The second decade (maturity level)**
- **IPv6**
- **ICMPv6 and Plug&Play**
- **Routing**
- **Transition**
- **Miscellaneous**

© 2012, D.I. Lindner IPv6, v4.7 36

IPv6 Evolution (1)

- **2000-2002:**
  - RFC 2765 Stateless IP/ICMP Translation Algorithm (SIIT) (Obsoleted by RFC6145)
  - RFC 2766 Network Address Translation - Protocol Translation (NAT-PT) (Obsoleted by RFC4966)
  - RFC 2767 Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS (Status: INFORMATIONAL)
  - RFC 2772 6Bone Backbone Routing Guidelines (Obsoletes RFC2546) (Updated by RFC3152) (Status: INFORMATIONAL)
  - RFC 2893 Transition Mechanisms for IPv6 Hosts and Routers) (Obsoletes RFC1933) (Obsoleted by RFC4213)
  - RFC 2894 Router Renumbering for IPv6 (Status: PROPOSED STANDARD)
  - RFC 3019 IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol (Obsoleted by RFC5519)
  - RFC 3041 Privacy Extensions for Stateless Address Autoconfiguration in IPv6 (Obsoleted by RFC4941) (Status: PROPOSED STANDARD)
  - RFC 3053 IPv6 Tunnel Broker (Status: INFORMATIONAL)
  - RFC 3056 Connection of IPv6 Domains via IPv4 Clouds (Status: PROPOSED STANDARD)
  - RFC 3068 An Anycast Prefix for 6to4 Relay Routers (Status: PROPOSED STANDARD)

IPv6 Evolution (2)

- **2000-2002 (cont.):**
  - RFC 3089 A SOCKS-based IPv6/IPv4 Gateway Mechanism (Status: INFORMATIONAL)
  - RFC 3122 Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification (Status: PROPOSED STANDARD)
  - RFC 3142 An IPv6-to-IPv4 Transport Relay Translator (Status: INFORMATIONAL)
  - RFC 3152 Delegation of IP6.ARPA. (Obsoleted by RFC3596)
  - RFC 3266 Support for IPv6 in Session Description Protocol (SDP) Obsoleted by RFC4566)
  - RFC 3306 Unicast-Prefix-based IPv6 Multicast Addresses (Updated by RFC3956, RFC4489) (Status: PROPOSED STANDARD)
  - RFC 3307 Allocation Guidelines for IPv6 Multicast Addresses. (Status: PROPOSED STANDARD)

© 2012, D.I. Lindner

IPv6 Evolution (3)

- **2002-2003:**
  - RFC 3307 Allocation Guidelines for IPv6 Multicast Addresses (Status: PROPOSED STANDARD)
  - RFC 3315 Dynamic Host Configuration Protocol for IPv6 (DHCPv6 (Updated by RFC4361, RFC5494, RFC6221) (Status: PROPOSED STANDARD)
  - RFC 3319 Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers (Status: PROPOSED STANDARD)
  - RFC 3338 Dual Stack Hosts Using "Bump-in-the-API" (BIA (Status: EXPERIMENTAL)
  - RFC 3484 Default Address Selection for Internet Protocol version 6 (IPv6 (Status: PROPOSED STANDARD)
  - RFC 3493 Basic Socket Interface Extensions for IPv6 (Obsoletes RFC2553) (Status: INFORMATIONAL)
  - RFC 3513 Internet Protocol Version 6 (IPv6) Addressing Architecture (Obsoletes RFC2373) (Obsoleted by RFC4291)
  - RFC 3542 Advanced Sockets Application Program Interface (API) for IPv6 (Obsoletes RFC2292) (Status: INFORMATIONAL)
  - RFC 3582 Goals for IPv6 Site-Multihoming Architectures (Status: INFORMATIONAL)
  - RFC 3587 IPv6 Global Unicast Address Format (Obsoletes RFC2374) (Status: INFORMATIONAL)
  - RFC 3595 Textual Conventions for IPv6 Flow Label (Status: PROPOSED STANDARD)
  - RFC 3596 DNS Extensions to Support IP Version 6 (Obsoletes RFC3152, RFC1886) (Status: DRAFT STANDARD)

IPv6 Evolution (4)

- **2003-2005:**
  - RFC 3633 IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6 (Status: PROPOSED STANDARD)
  - RFC 3646 DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6) (Status: PROPOSED STANDARD)
  - RFC 3697 IPv6 Flow Label Specification (Status: PROPOSED STANDARD)
  - RFC 3750 Unmanaged Networks IPv6 Transition Scenarios (Status: INFORMATIONAL)
  - RFC 3756 IPv6 Neighbor Discovery (ND) Trust Models and Threats (Status: INFORMATIONAL)
  - RFC 3769 Requirements for IPv6 Prefix Delegation (Status: INFORMATIONAL)
  - RFC 3775 Mobility Support in IPv6 (Obsoleted by RFC6275)
  - RFC 3776 Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents (Updated by RFC4877) (Status: PROPOSED STANDARD)
  - RFC 3810 Multicast Listener Discovery Version 2 (MLDv2) for IPv6 (Updates RFC2710) (Updated by RFC4604) (Status: PROPOSED STANDARD)
  - RFC 3901 DNS IPv6 Transport Operational Guidelines (Also BCP0091) (Status: BEST CURRENT PRACTICE)
  - RFC 3904 Evaluation of IPv6 Transition Mechanisms for Unmanaged Networks (Status: INFORMATIONAL)
  - RFC 3971 SEcure Neighbor Discovery (SEND) (Status: PROPOSED STANDARD)

© 2012, D.I. Lindner

IPv6 Evolution (5)

- **2005-2006:**
  - RFC 4007 IPv6 Scoped Address Architecture (Status: PROPOSED STANDARD))
  - RFC 4057 IPv6 Enterprise Network Scenarios (Status: INFORMATIONAL)
  - RFC 4068 Fast Handovers for Mobile IPv6 (Obsoleted by RFC5268) (Status: EXPERIMENTAL)
  - RFC 4074 Common Misbehavior Against DNS Queries for IPv6 Addresses (Status: INFORMATIONAL)
  - RFC 4075 Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6 (Status: PROPOSED STANDARD)
  - RFC 4076 Renumbering Requirements for Stateless Dynamic Host Configuration Protocol for IPv6 (DHCPv6) (Status: INFORMATIONAL)
  - RFC 4213 Basic Transition Mechanisms for IPv6 Hosts and Routers (Obsoletes RFC2893) (Status: PROPOSED STANDARD)
  - RFC 4219 Things Multihoming in IPv6 (MULTI6) Developers Should Think About (Status: INFORMATIONAL)
  - RFC 4241 A Model of IPv6/IPv4 Dual Stack Internet Access Service (Status: INFORMATIONAL)
  - RFC 4242 Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6) (Status: PROPOSED STANDARD)

IPv6 Evolution (6)

- **2006:**
  - RFC 4285 Authentication Protocol for Mobile IPv6 (Status: INFORMATIONAL)
  - RFC 4291 IP Version 6 Addressing Architecture (Obsoletes RFC3513) (Updated by RFC5952, RFC6052) (Status: DRAFT STANDARD)
  - RFC 4294 IPv6 Node Requirements (Updated by RFC5095) (Status: INFORMATIONAL)
  - RFC 4295 Mobile IPv6 Management Information Base (Status: PROPOSED STANDARD)
  - RFC 4301 Security Architecture for the Internet Protocol (Obsoletes RFC2401) (Updates RFC3168) (Updated by RFC6040) (Status: PROPOSED STANDARD)

IPsec 1.0

  - RFC 4302 IP Authentication Header. S. Kent (Obsoletes RFC2402) (Status: PROPOSED STANDARD)
  - RFC 4303 IP Encapsulating Security Payload (ESP (Obsoletes RFC2406) (Status: PROPOSED STANDARD)
  - RFC 4306 Internet Key Exchange (IKEv2) Protocol (Obsoletes RFC2407, RFC2408, RFC2409) (Obsoleted by RFC5996) (Updated by RFC5282)
  - RFC 4307 Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2 (Status: PROPOSED STANDARD)

IKEv2 0.9

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

IPv6 Evolution (7)

- **2006-2007:**
  - **RFC 4308 Cryptographic Suites for IPsec (Status: PROPOSED STANDARD)**
  - **RFC 4311 IPv6 Host-to-Router Load Sharing (Updates RFC2461) (Status: PROPOSED STANDARD)**
  - RFC 4339 IPv6 Host Configuration of DNS Server Information Approaches (Status: INFORMATIONAL)
  - RFC 4380 Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs (Updated by RFC5991, RFC6081) (Status: PROPOSED STANDARD)

ICMPv6 1.0

  - RFC 4443 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification (Obsoletes RFC2463) (Updates RFC2780) (Updated by RFC4884) (Status: DRAFT STANDARD)
  - RFC 4449 Securing Mobile IPv6 Route Optimization Using a Static Shared Key (Status: PROPOSED STANDARD)
  - RFC 4541 Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches) (Status: INFORMATIONAL)
  - RFC 4552 Authentication/Confidentiality for OSPFv3 (Status: PROPOSED STANDARD)
  - RFC 4477 Dynamic Host Configuration Protocol (DHCP): IPv4 and IPv6 Dual-Stack Issues (Status: INFORMATIONAL)
  - RFC 4487 Mobile IPv6 and Firewalls: Problem Statement (Status: INFORMATIONAL)
  - RFC 4489 A Method for Generating Link-Scoped IPv6 Multicast Addresses (Updates RFC3306) (Status: PROPOSED STANDARD)
  - RFC 4649 Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option (Status: PROPOSED STANDARD)

IPv6 Evolution (8)

- **2006-2007 (cont.):**
  - RFC 4604 Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast (Updates RFC3376, RFC3810) (Status: PROPOSED STANDARD)
  - RFC 4651 A Taxonomy and Analysis of Enhancements to Mobile IPv6 Route Optimization (Status: INFORMATIONAL)
  - RFC 4692 Considerations on the IPv6 Host Density Metric. (Status: INFORMATIONAL)
  - RFC 4704 The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option (Status: PROPOSED STANDARD)
  - RFC 4727 Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers (Status: PROPOSED STANDARD)
  - RFC 4776 Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information (Obsoletes RFC4676) (Updated by RFC5774) (Status: PROPOSED STANDARD)
  - RFC 4779 ISP IPv6 Deployment Scenarios in Broadband Access Networks. (Status: INFORMATIONAL)
  - RFC 4798 Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE) (Status: PROPOSED STANDARD)
  - RFC 4864 Local Network Protection for IPv6 (Status: INFORMATIONAL)
  - RFC 4852 IPv6 Enterprise Network Analysis - IP Layer 3 Focus) (Status: INFORMATIONAL)

IPv6 Evolution (9)

- **2007:**
  - **RFC 4861** Neighbor Discovery for IP version 6 (IPv6) (Obsoletes RFC2461) (Updated by RFC5942) (Status: DRAFT STANDARD)
  - **RFC 4862** IPv6 Stateless Address Autoconfiguration (Obsoletes RFC2462) (Status: DRAFT STANDARD)
  - **RFC 4864** Local Network Protection for IPv6 (Status: INFORMATIONAL)
  - **RFC 4866** Enhanced Route Optimization for Mobile IPv6 (Status: PROPOSED STANDARD)
  - **RFC 4890** Recommendations for Filtering ICMPv6 Messages in Firewalls (Status: INFORMATIONAL)
  - **RFC 4891** Using IPsec to Secure IPv6-in-IPv4 Tunnels (Status: INFORMATIONAL)
  - **RFC 4941** Privacy Extensions for Stateless Address Autoconfiguration in IPv6 (Obsoletes RFC3041) (Status: DRAFT STANDARD)
  - **RFC 4942** IPv6 Transition/Co-existence Security Considerations (Status: INFORMATIONAL)
  - **RFC 4943** IPv6 Neighbor Discovery On-Link Assumption Considered Harmful (Status: INFORMATIONAL)
  - **RFC 4944** Transmission of IPv6 Packets over IEEE 802.15.4 Networks (Updated by RFC6282) (Status: PROPOSED STANDARD)
  - **RFC 4966** Reasons to Move the Network Address Translator – Protocol Translator (NAT-PT) to Historic Status. (Obsoletes RFC2766) (Status: INFORMATIONAL)
  - **RFC 4968** Analysis of IPv6 Link Models for 802.16 Based Networks (Status: INFORMATIONAL)

ND 1.0

SLAAC 1.0

© 2012, D.I. Lindner IPv6, v4.7 45

IPv6 Evolution (10)

- **2007-2008:**
  - **RFC 4994** DHCPv6 Relay Agent Echo Request Option (Status: PROPOSED STANDARD)
  - **RFC 5007** DHCPv6 Lease query (Status: PROPOSED STANDARD))
  - **RFC 5014** IPv6 Socket API for Source Address Selection (Status: INFORMATIONAL)
  - **RFC 5072** IP Version 6 over PPP (Obsoletes RFC2472) (Status: DRAFT STANDARD)
  - RFC 5075 IPv6 Router Advertisement Flags Option (Obsoleted by RFC5175)
  - **RFC 5094** Mobile IPv6 Vendor Specific Option. (Status: PROPOSED STANDARD)
  - **RFC 5095** Deprecation of Type 0 Routing Headers in IPv6 (Updates RFC2460, RFC4294) (Status: PROPOSED STANDARD)
  - **RFC 5156** Special-Use IPv6 Addresses (Status: INFORMATIONAL)
  - **RFC 5157** IPv6 Implications for Network Scanning (Status: INFORMATIONAL)
  - **RFC 5158** 6to4 Reverse DNS Delegation Specification (Status: INFORMATIONAL)
  - **RFC 5172** Negotiation for IPv6 Datagram Compression Using IPv6 Control Protocol (Obsoletes RFC2472) (Status: PROPOSED STANDARD)
  - **RFC 5175** IPv6 Router Advertisement Flags Option (Obsoletes RFC5075) (Status: PROPOSED STANDARD)
  - **RFC 5187** OSPFv3 Graceful Restart (Status: PROPOSED STANDARD)

© 2012, D.I. Lindner IPv6, v4.7 46

IPv6 Evolution (11)

- **2008:**
  - 5201 Host Identity Protocol (Updated by RFC 6253) (Status: EXPERIMENTAL)
  - 5202 Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP) (Status: EXPERIMENTAL)
  - 5203 Host Identity Protocol (HIP) Registration Extension (Status: EXPERIMENTAL)
  - 5204 Host Identity Protocol (HIP) Rendezvous Extension (Status: EXPERIMENTAL)
  - 5205 Host Identity Protocol (HIP) Domain Name System (DNS) Extensions (Status: EXPERIMENTAL)
  - 5206 End-Host Mobility and Multihoming with the Host Identity Protocol (Status: EXPERIMENTAL)
  - 5207 NAT and Firewall Traversal Issues of Host Identity Protocol (HIP) Communication (Status: INFORMATIONAL)

© 2012, D.I. Lindner IPv6, v4.7 47

IPv6 Evolution (12)

- **2008-2009:**
  - RFC 5268 Mobile IPv6 Fast Handovers (Obsoletes RFC4068) (Obsoleted by RFC5568)
  - **RFC 5308** Routing IPv6 with IS-IS (Status: PROPOSED STANDARD)
  - **RFC 5340 OSPF** for IPv6 (Obsoletes RFC2740) (Status: PROPOSED STANDARD)
  - **RFC 5350** IANA Considerations for the IPv4 and IPv6 Router Alert Options (Updates RFC2113, RFC3175) (Status: PROPOSED STANDARD)
  - **RFC 5375** IPv6 Unicast Address Assignment Considerations (Status: INFORMATIONAL)
  - **RFC 5453** Reserved IPv6 Interface Identifiers (Status: PROPOSED STANDARD)
  - **RFC 5533** Shim6: Level 3 Multihoming Shim Protocol for IPv6 (Status: PROPOSED STANDARD)
  - **RFC 5534** Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming (REAP for Shim6) (Status: PROPOSED STANDARD)
  - **RFC 5555** Mobile IPv6 Support for Dual Stack Hosts and Routers (Status: PROPOSED STANDARD)
  - **RFC 5568** Mobile IPv6 Fast Handovers (Obsoletes RFC5268) (Status: PROPOSED STANDARD)

OSPFv3 1.0

© 2012, D.I. Lindner IPv6, v4.7 48

IPv6 Evolution (14)

- **2009-2010 cont:**
  - **RFC 5569** IPv6 Rapid Deployment on IPv4 Infrastructures (6rd (Status: INFORMATIONAL)
  - RFC 5572 IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP) (Status: EXPERIMENTAL)
  - RFC 5643 Management Information Base for OSPFv3 (Status: PROPOSED STANDARD)
  - **RFC 5701** IPv6 Address Specific BGP Extended Community Attribute (Status: PROPOSED STANDARD)
  - RFC 5747 4over6 Transit Solution Using IP Encapsulation and MP-BGP Extensions (Status: EXPERIMENTAL)
  - RFC 5722 Handling of Overlapping IPv6 Fragments (Updates RFC2460) (Status: PROPOSED STANDARD)
  - 5770 Basic Host Identity Protocol (HIP) Extensions for Traversal of Network Address Translators (Status: EXPERIMENTAL)
  - **RFC 5798** Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6 (Obsoletes RFC3768) (Status: PROPOSED STANDARD)
  - **RFC 5838** Support of Address Families in OSPFv3 (Status: PROPOSED STANDARD)
  - **RFC 5844** IPv4 Support for Proxy Mobile IPv6 (Status: PROPOSED STANDARD)
  - RFC 5871 IANA Allocation Guidelines for the IPv6 Routing Header (Format: TXT=4000 bytes) (Updates RFC2460 (Status: PROPOSED STANDARD)

IPv6 Evolution (15)

- **2009-2010 cont:**
  - **RFC 5881** Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop (Status: PROPOSED STANDARD)
  - **RFC 5902** IAB Thoughts on IPv6 Network Address Translation (Status: INFORMATIONAL)
  - **RFC 5908** Network Time Protocol (NTP) Server Option for DHCPv6 (Status: PROPOSED STANDARD)
  - **RFC 5942** IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes (Updates RFC4861) (Status: PROPOSED STANDARD)
  - **RFC 5949** Fast Handovers for Proxy Mobile IPv6 (Status: PROPOSED STANDARD)
  - **RFC 5952** A Recommendation for IPv6 Address Text Representation (Updates RFC4291) (Status: PROPOSED STANDARD)
  - **RFC 5954** Essential Correction for IPv6 ABNF and URI Comparison in RFC 3261 (Updates RFC3261) (Status: PROPOSED STANDARD)
  - **RC 5963** IPv6 Deployment in Internet Exchange Points (IXPs (Status: INFORMATIONAL)

IPv6 Evolution (16)

- **2010-2011:**
  - **RFC 5969** IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) – Protocol Specification (Status: PROPOSED STANDARD)
  - **RFC 5970** DHCPv6 Options for Network Boot (Status: PROPOSED STANDARD)
  - **RFC 5991** Teredo Security Updates (Updates RFC4380) (Status: PROPOSED STANDARD)
  - **RFC 5996** Internet Key Exchange Protocol Version 2 (IKEv2) (Obsoletes RFC4306, RFC4718) (Updated by RFC5998) (Status: PROPOSED STANDARD)

    IKEv2 1.0
  - **RFC 6018** IPv4 and IPv6 Greynets (Status: INFORMATIONAL)
  - **RFC 6052** IPv6 Addressing of IPv4/IPv6 Translators (Updates RFC4291) (Status: PROPOSED STANDARD)
  - **RFC 6081** Teredo Extensions. (Updates RFC4380) (Status: PROPOSED STANDARD)
  - **RFC 6085** Address Mapping of IPv6 Multicast Packets on Ethernet (Updates RFC2464) (Status: PROPOSED STANDARD)
  - **RFC 6104** Rogue IPv6 Router Advertisement Problem Statement (Status: INFORMATIONAL)
  - **RFC 6105** IPv6 Router Advertisement Guard (Status: INFORMATIONAL)

IPv6 Evolution (17)

- **2011:**
  - **RFC 6106** IPv6 Router Advertisement Options for DNS Configuration. (Obsoletes RFC5006) (Status: PROPOSED STANDARD)
  - **RFC 6119** IPv6 Traffic Engineering in IS-IS (Status: PROPOSED STANDARD)
  - **RFC 6127** IPv4 Run-Out and IPv4-IPv6 Co-Existence Scenarios (Status: INFORMATIONAL)
  - **RFC 6145** Stateless IP/ICMP Translation Algorithm (SIIT) (Obsoletes RFC2765) (Status: PROPOSED STANDARD)
  - **RFC 6146** Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers (Status: PROPOSED STANDARD)
  - **RFC 6147** DNS64: DNS Extensions for Network Address Translation from IPv6Clients to IPv4 Servers (Status: PROPOSED STANDARD)
  - **RFC 6153** DHCPv4 and DHCPv6 Options for Access Network Discovery and Selection Function (ANDSF) Discovery (Status: PROPOSED STANDARD)
  - **RFC 6164** Using 127-Bit IPv6 Prefixes on Inter-Router Links (Status: PROPOSED STANDARD)
  - **RFC 6180** Guidelines for Using IPv6 Transition Mechanisms during IPv6 Deployment (Status: INFORMATIONAL)
  - **RFC 6221** Lightweight DHCPv6 Relay Agent (Updates RFC3315) (Status: PROPOSED STANDARD)

IPv6 Evolution (18)

- **2011 cont.:**
  - RFC 6275 Mobility Support in IPv6 (Obsoletes RFC3775) (Status: PROPOSED STANDARD)
  - RFC 6296 IPv6-to-IPv6 Network Prefix Translation (Status: EXPERIMENTAL)
  - RFC 6311 Protocol Support for High Availability of IKEv2/IPsec (Status: PROPOSED STANDARD)
  - RFC 6324 Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations (Status: INFORMATIONAL)
  - RFC 6333 Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion (Status: PROPOSED STANDARD)
  - RFC 6334 Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite (Status: PROPOSED STANDARD)
  - RFC 6343 Advisory Guidelines for 6to4 Deployment (Status: INFORMATIONAL)
  - RFC 6346 The Address plus Port (A+P) Approach to the IPv4 Address Shortage (Status: EXPERIMENTAL)

Agenda

- **History**
- **IPv6**
  - **IPv6 Facts**
  - Review IPv4 Header
  - IPv6 Main Header
  - IPv6 Extension Headers
  - Security
  - Addressing
- **ICMPv6 and Plug&Play**
- **Routing**
- **Transition**

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Why IPv6?

- **Official IPv4 addresses are depleted**
  - IANA address pool is empty -> all given to RIR (2011-02)
  - APNIC down to last /8 (2011-04); each ISP will gets a /22
  - RIPE address pool will be exhausted by June 2012
  - ARIN address pool will be exhausted in 2013
  - LACNIC, AfriNIC will be exhausted by end of 2014
- **Growth can not be handled by IPv4 anymore**
  - Huge demand for IP addresses by smart-phones (connected to the Internet all the time -> IP address can not be shared), consumer electronics (televisions, game consoles), new devices like power meters etc.
- **There is not any alternative to IPv6**
  - To grow the Internet

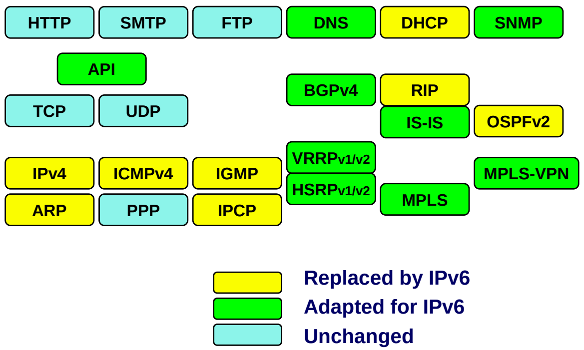
Who Needs IPv6?

- **The Internet Service Providers (ISPs)**
  - In order to connect new customers
  - In order to support growing amount of mobile devices
- **Some very large enterprises**
  - Private Address 10.0.0.0/8 is not sufficient enough
- **Autonomous devices**
  - Who be operated in a a large-scale global addressing
- **Peer-to-peer application developers**
  - No need for NAT
- **Product development**
  - For emerging markets with IPv6 from the beginning
  - For supporting new IPv6 only customers without any sick IPv4 to IPv6 transition / translation gateways

IPv6 Facts (1)

- IPv6 is a replacement for IPv4
- IPv6 technology
  - Still connectionless packet switching (datagram service)
  - Replacement for IPv4 with longer addresses (128 bit)
  - Changes of L2/L3 protocols including routing protocols
  - Upper layers and applications should not change
- IPv6 is handled as separate protocol family
- IPv6 changes concentrate
  - On network layer (mainly longer addresses)
- Upper layers and application should not change
  - Unfortunately not completely true

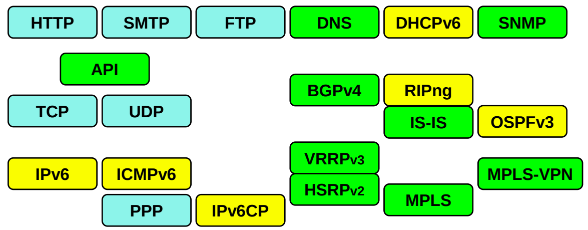
IPv4 Protocol Stack



Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

IPv6 Protocol Stack



- ICMPv6 replaces ARP, ICMPv4, IGMP and introduces SLAAC (Serverless Automatic Address Configuration)
- DNS is adapted by AAAA records
- BGP supports IPv6 with address family feature
- SNMP is adapted for support of v6-MIBs
- API is adapted for handling larger addresses
- IPv6 over MPLS (6PE): IPv4 in the core, BGP to announce IPv6 prefixes
- IPv6 MPLS VPN (6VPE): IPv4 in the core; like IPv4 MPLS VPN

Adaptation Protocols / Sockets

- Protocols to be replaced / adapted
  - IPv4 -> IPv6
  - ICMP, IGMP, ARP -> ICMPv6
  - UDP / TCP -> UDPv6 / TCPv6
    - That was the original idea but was never done !!!
  - RIP, OSPF -> RIPng, OSPFv3
    - Note: IS-IS, BGP supports IPv6 as new address family
  - VRRPv2 -> VRRPv3
  - DNS (-> new AAAA resource record)
  - DHCP -> DHCPv6
- Standard programming interfaces need to be adapted
  - Address data structure
  - Name-to-address translation functions
  - Address conversion functions
- Adaptation to IPv6 should be straightforward
  - Address aspects must be handled mainly

IPv6 Facts (2)

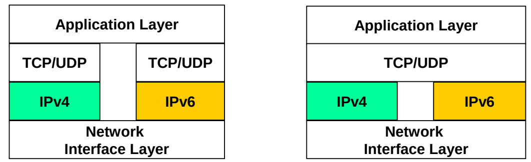
- **IPv6 in the network**
  - Multiprotocol routing like in the old days of parallel Novell-, Appletalk-, Decnet- and IP-routing
    - Therefore new routing protocols to support IPv6
  - But in the core we still may have IPv4 only networks
    - Tunneling IPv6 across IPv4 domains
  - IPv6 only networks
    - For network regions where IPv4 addresses are completely exhausted
    - Tunneling IPv4 over IPv6 domains
    - Address translation between IPv6 and IPv4 (NAT64)
- **IPv6 needs no NAT anymore**
  - Good for peer-to-peer application development
  - But address hiding and sharing dynamic addresses is not feasible anymore (privacy and security aspects of SOHO)
    - Security should be achieved anyway by a stateful firewall solution

IPv6 Facts (3)

- **IPv6 in the end system**
  - Dual IP layer in case both (IPv4 and IPv6) should be supported
  - Some logic necessary to decide which layer to be used (DNS plays a major role)
  - IPv6 only stack
    - How to reach IPv4 content -> Translation NAT-PT -> NAT64
  - IPv4 only stack
    - How to reach IPv6 content -> ???
- **IPv6 implementation**
  - You have to forget certain IPv4 habits
  - You have to do already IPv4-known things in a new different way

Dual-Stack versus Dual-IP Layer

Dual-Stack **Dual-IP Layer**



Separate driver implementation of TCP/UDP for IPv4 and IPv6

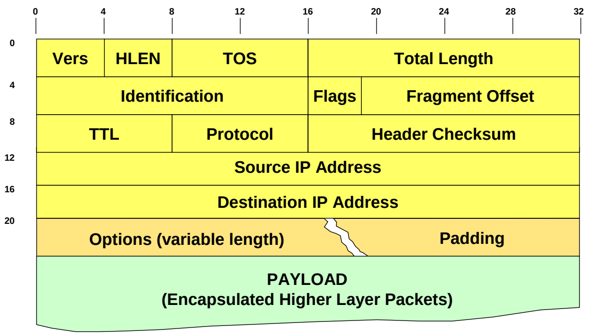
Summary of Changes in IPv6 Protocol Stack

- **128 bit address**
- **Host interfaces have multiple IPv6 addresses**
  - Link-Local-, Unique-Local-, Global-Addresses
  - Unique-Local-Addresses (ULA) replaces private addresses
  - Global-Addresses avoid usage of NAT
- **ARP**
  - Replaced by ICMPv6 Neighbor Discovery on all media types
  - Note: IPCPv6 (PPP) lost address assignment feature known by IPCPv4 for dial-in clients
- **Default gateway configuration**
  - Replaced by ICMPv6 Router Advertisements
- **Stateless Address Autoconfiguration (SLAAC)**
  - Replaces need for DHCP and is available on all media types
  - DHCP may still be necessary for DNS server discovery

Agenda

- History
- **IPv6**
  - IPv6 Facts
  - **Review IPv4 Header**
  - IPv6 Main Header
  - IPv6 Extension Headers
  - Security
  - Addressing
- ICMPv6
- Routing
- Transition

Review IPv4 Header



Review IPv4 Header Entries

- **Version (4 bits)**
  - version of the IP protocol = 4
- **HLEN (4 bits)**
  - length of the header in 32 bit words
- **Type of Service (TOS) (8 bits)**
  - priority of a datagram (precedence bits)
  - preferred network characteristics (D, T, R and C bits)
  - Nowadays replaced by DSCP (Differentiated Services Code Point) to indicate a service class in Diff-Serv-QoS networks
- **Total Length (16 bits)**
  - total length of the IP datagram (header + data) in octets

Review IPv4 Header Entries

- **Identification (16 bits)**
  - unique identification of a datagram, used for fragmentation and reassembling
- **Flags (for fragmentation) (3 bits)**
  - Reserved
  - DF (do not fragment)
  - MF (more fragments)
- **Fragment Offset (13 bits)**
  - position of a fragment relative to the beginning of the original datagram, Offset is measured in multiples of 8 octets (64 bits)

Review IPv4 Header Entries

- **Time To Live (TTL) (8 bits)**
  - limits the lifetime of a datagram in the network (units are seconds, range 0-255)
  - is set by an end system and decremented by every intermediate router
  - If TTL reaches 0, the datagram is discarded
- **Protocol (8 bits)**
  - indicates the higher layer protocol
- **Header Checksum (16 bits)**
  - change of TTL means recalculation at every hop
- **Source- and Destination IP address**
  - 32 bit each
- **Options with variable length**

Original TOS Field (RFC 1349)

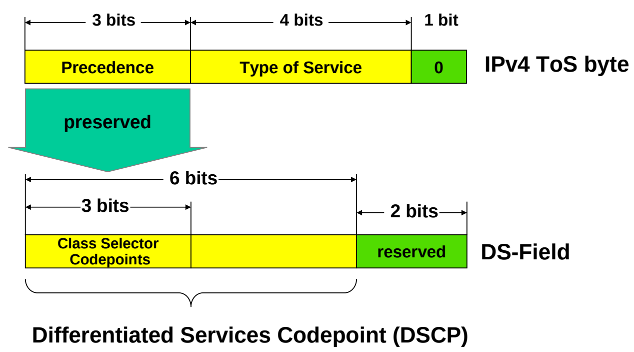
Precedence	D	T	R	C	"0"

Precedence (Priority):	DTRC bits:		
111 Network Control	0 0 0 0 . .	. . . .	normal service
110 Internetwork Control	1 0 0 0 D	Delay	min. delay
101 Critic/ECP	0 1 0 0 T	Throughput	max. throughput
100 Flash Override	0 0 1 0 R	Reliability	max. reliability
011 Flash	0 0 0 1 C	Cost	min. cost
010 Immediate			
001 Priority	No other values are defined but have to be accepted (ignored) by a router or host.		
000 Routine			

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

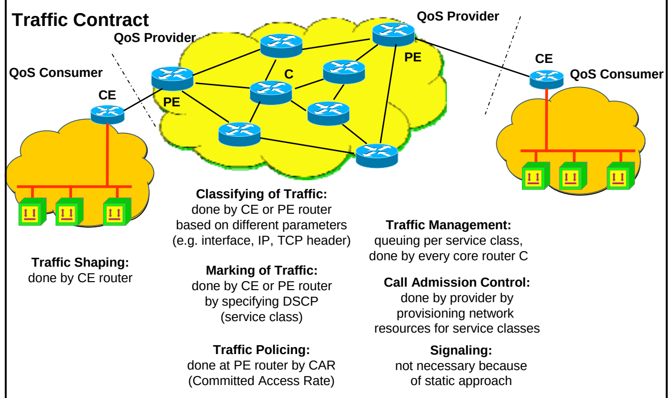
IPv4 TOS Recycling



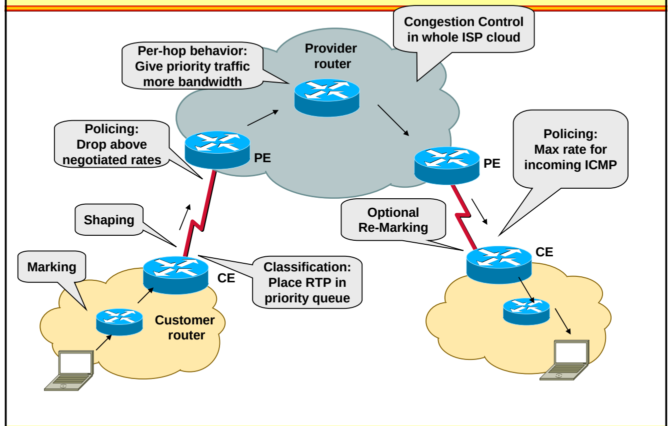
DSCP Usage

- **Important for IP QoS (Quality of Service)**
  - IP QoS Differentiated Services Model
    - RFC 2474: "Definition of the Differentiated Service Field in the IPv4 and IPv6 Headers"
    - RFC 2475: "An Architecture for Differentiated Services"
  - Remember
    - IP is basically a Best Effort Service, therefore not suited for interactive real-time traffic like voice and video
  - Using DSCP a IP datagram can be labelled at the border of IP QoS domain
    - with a certain traffic class
  - Traffic class will receive a defined handling within in IP QoS Domain
    - e.g. limited delay, guaranteed throughput

Differentiated Services Model



Differentiated Services In Action



© 2012, D.I. Lindner

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

14 Recommended Code Points

- **Expedited Forwarding (EF)**
  - DSCP 46 = 101 110 binary
  - For low delay, low loss, and low jitter
  - Defined in RFC 3246
- **Assured Forwarding (AF)**
  - 12 codepoints: 4 classes and 3 drop precedence each
  - Defined in RFC 2597
- **Best Effort (BE)**
  - 000000 binary

Assured Forwarding (AF)

- **Guarantees a certain bandwidth to a traffic class**
  - If the traffic exceeds the committed bandwidth the drop probability is raised according to the specified drop precedence
- **There are 12 different AF behavior code points**
  - Consisting of 4 classes (AF1y to AF4y)
  - And 3 drop probabilities (AFx1 to AFx3) for each class (low/med/hi)

Drop:	Class 1			Class 2			Class 3			Class 4		
Low	AF11	10	001010	AF21	18	010010	AF31	26	011010	AF41	34	100010
Medium	AF12	12	001100	AF22	20	010100	AF32	28	011100	AF42	36	100100
High	AF13	14	001110	AF23	22	010110	AF33	30	011110	AF43	38	100110

decimal | binary

© 2012, D.I. Lindner

Class Selector Values

```
Level Meaning
===== ===================================================
7     unchanged (link layer and routing protocol keep alive)
6     unchanged (used for IP routing protocols)
5     Express Forwarding (EF)
4     AF Class 4
3     AF Class 3
2     AF Class 2
1     AF Class 1
0     Best effort
```

- **The legacy IP Precedence values (0-7) can be directly mapped into the three Class Selector bits (0,1,2) *with the three LSBs (3,4,5) set to zero***
- **This results in the seven CSx values**
  - CS0 = DSCP 00 = 000000
  - ...
  - CS7 = DSCP 56 = 111000

DSCP Values Overview

Code Point Name	DSCP hex	DSCP dec	Whole IP TOS byte binary	Whole IP TOS byte hex	Whole IP TOS byte dec
EF	0x2e	46	10111000	0xb8	184
AF41	0x22	34	10001000	0x88	136
AF42	0x24	36	10010000	0x90	144
AF43	0x26	38	10011000	0x98	152
AF31	0x1a	26	01101000	0x68	104
AF32	0x1c	28	01110000	0x70	112
AF33	0x1e	30	01111000	0x78	120
AF21	0x12	18	01001000	0x48	72
AF22	0x14	20	01010000	0x50	80
AF23	0x16	22	01011000	0x18	24
AF11	0x0a	10	00101000	0x28	40
AF12	0x0c	12	00110000	0x30	48
AF13	0x0e	14	00111000	0x38	56
CS7	0x38	56	11100000	0xe0	224
CS6	0x30	48	11000000	0xc0	192
CS5	0x28	40	10100000	0xa0	160
CS4	0x20	32	10000000	0x80	128
CS3	0x18	24	01100000	0x60	96
CS2	0x10	16	01000000	0x40	64
CS1	0x08	8	00100000	0x20	32
CS0 = BE	0x00	0	00000000	0x00	0

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Agenda

- **History**
- **IPv6**
  - IPv6 Facts
  - Review IPv4 Header
  - **IPv6 Main Header**
  - IPv6 Extension Headers
  - Security
  - Addressing
- **ICMPv6 and Plug&Play**
- **Routing**
- **Transition**

IPv6 Header Overview

- **128 bit addresses**
  - Written in hexadecimal notation
- **Simple header**
  - Only basic functions for (fast hardware) switching of IPv6 packets
- **Extension headers**
  - For advanced or optional functions
- **Support of**
  - Auto-configuration
  - Authentication and privacy (encryption)
  - Source routes
  - Flow identification (QoS)
- **Daisy-chain of headers**

Original IPv6 Basic Header (RFC 1883)

	0	4	8	16	24	31
0	Version	Priority	Flow Label			
4	Payload Length			Next Header	Hop Limit	
8	Source IP Address					
24	Destination IP Address					
40						

Actual IPv6 Basic Header (RFC 2460)

	0	4	12	16	24	31
0	Version	Traffic Class	Flow Label			
4	Payload Length			Next Header	Hop Limit	
8	Source IP Address					
24	Destination IP Address					
40						

Traffic Class = IPv4 DSCP Diff-Srv Code Point

© 2012, D.I. Lindner

IPv6 Basic Header Entries (1)

- **Version (4 bits)**
  - Version of the IP protocol = 6
    - note: new Ethernet-type for IPv6: 0x86DD (IPv4: 0x0800)
- **Priority (4 / 8 bits) and Flow Label (24 /20 bits)**
  - Facilitates support of real-time service or non default quality of service
- **Payload Length (16 bits)**
  - Length of the payload (data only)
  - Max. length 65.535 byte
  - Larger length than 65.535 is possible
    - **Jumbogram**
      - Payload length = all zeros
      - Actual length in hop-by-hop options extension header

IPv6 Basic Header Entries (2)

- **Next Header (8 bits)**
  - Indicates the next header following the IPv6 header
  - Same values allowed as protocol field in IPv4 header
    - IP in IP (4), TCP (6), UDP (17), ICMPv6 (58), OSPF (89), etc
    - new values reserved for extension headers
- **Hop Limit (8 bits)**
  - Same function as TTL in IPv4
    - With the exception that this field is decremented by one by each node
- **Source- and Destination IP address**
  - 16 bytes (128 bit) each
  - Destination address need not be address of destination end-system
    - Could be address of intermediate systems in case of routing extension header

© 2012, D.I. Lindner

Comparison of IPv6 and IPv4 (1)

- **IPv6 simplifications**
  - Fixed format of all headers
    - Initial 64 bit plus 128 bit IPv6 addresses
    - Therefore IPv4 HLEN not necessary in IPv6
    - Next Header field implicitly indicates length of next header
  - Header checksum removed
    - Processing overhead reduced
    - No header checksum in IPv6
  - Hop-by-hop fragmentation procedure removed
    - IPv4 fragmentation fields (Identification, Fragment Offset, Flags) are not necessary in IPv6
    - IPv6 host must use MTU path discovery (RFC 1981)
      - Note: RFC 1191 describes MTU path discovery IPv4

Comparison of IPv6 and IPv4 (2)

- **Renamed/redefined fields**
  - IPv4 Protocol field replaced by IPv6 Next Header field
  - IPv6 Payload Length versus IPv4 Total Length
  - IPv4 TTL renamed in IPv6 Hop Limit
    - IPv6 counts number of hops instead number of seconds (IPv4)
- **New fields**
  - Priority
    - Originally role (RFC 1983) could be compared with ToS precedence field, facilitates queuing strategy in a router
    - Nowadays used as traffic class (DSCP)
  - Flow Label can be used to implement QoS support
    - Is used to distinguish a flow of packets that require the same treatment
      - E.g. packets that are sent by a given source to a given destination belonging to a special traffic class reserved by RSVP

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Comparison of IPv6 and IPv4 (3)

- **Suppressed fields**
  - Header length HLEN
  - ToS (D, T, R and C bits)
  - Identification, Flags, Fragment Offset
- **Some IPv4 options moved to extension headers**
  - Remember: IPv4 options allow special-case treatment of some packets
    - Security, source routing, record route, timestamp, etc.
    - Leads to performance penalty
  - Normal (and hence fastest) packet forwarding based on basic IPv6 header only
  - Processing of normal IPv6 packets need not take care of options

Priority (RFC 1883 Old)

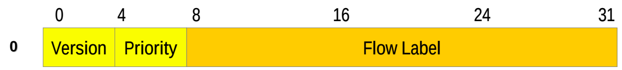
	0	4	8 16 24 31
0	Version	Priority	Flow Label

Values 0 - 7 are used to specify the priority of traffic for which the source is providing congestion control, e.g. traffic that "backs off" in response to congestion such as TCP traffic.

0 - uncharacterized traffic
1 - "filler" traffic (e.g., netnews)
2 - unattended data transfer (e.g., email)
3 - (reserved)
4 - attended bulk transfer (e.g., FTP, NFS)
5 - (reserved)
6 - interactive traffic (e.g., telnet, X, database access)
7 - internet control traffic (e.g., routing protocols, SNMP)

Values 8 - 15 are used to specify the priority of traffic that does not back off in response to congestion, e.g. "real-time" packets being sent at a constant rate.

Flow Label (RFC 1883 Old)

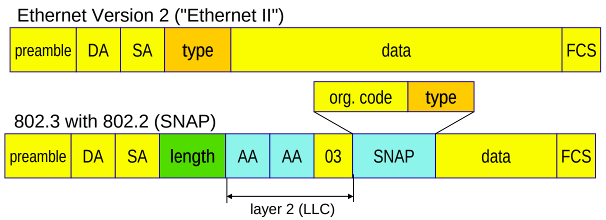


The 24-bit Flow Label field in the IPv6 header may be used by a source to label those packets for which it requests special handling by the IPv6 routers, such as non-default quality of service or "real-time" service. Hosts or routers that do not support the functions of the Flow Label field are required to set the field to zero when originating a packet, pass the field on unchanged when forwarding a Packet and ignore the field when receiving a packet

A flow is a sequence of packets sent from a particular source to a particular (unicast or multicast) destination for which the source desires special handling by the intervening routers. The nature of that special handling might be conveyed to the routers by a control protocol, such as a resource reservation protocol, or by information within the flow's packets themselves, e.g., in a hop-by-hop option.

IPv6 over Ethernet

- **Type = 0x86DD**



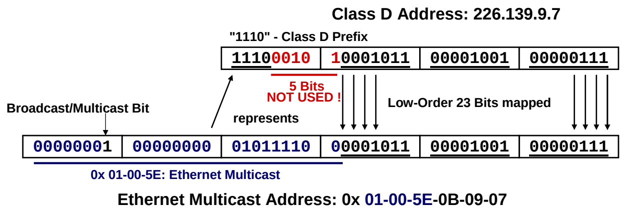
- **Multicast mapping L3->L2:**
  - Prefix 3333
  - Resulting MAC=3333<last 32 bits of IPv6 address>

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Class D IPv4 ⇨ LAN Multicast Address

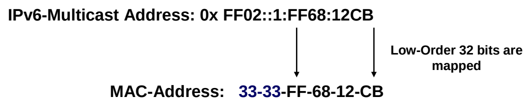
- **Example of**
  - Mapping Class D IPv4 address to Ethernet multicast address
    - Static mapping, no ARP
    - IANA has reserved range 0x 01005E000000- 0x 01005E7FFFFF



Class D IPv4 ⇨ LAN Multicast Address

- **5 bits cannot be mapped**
- **32:1 address ambiguity**
  - 32 different IP-Multicast-Groups have the same multicast MAC-Address
  - Filtering is needed by taking IP-Address into account

- **In IPv6 very similar:**



IPv6 over PPP

- **HDLC framing and encapsulation (RFC 1662)**
- **Protocol = 0x0057 for IPv6**
- **New IPv6CP Protocol 0x8075 (RFC 5072)**
  - No IP address assignment anymore possible like in IPCP (= DHCP over WAN) for Dial-In Clients
  - If address assignment is still necessary – SLAAC (**S**erverless **A**ddress **A**uto **C**onfiguration) plus DHCPv6

Flag	Address	Control	Protocol	Information	FCS	Flag

Flag	= 01111110		Protocol	=	see RFC 1700 (assigned numbers)
Address	= 11111111		Information	=	Network Layer PDU
Control	= 00000011 (UI frame)		FCS	=	16 bit

IPv6 MTU (RFC 2460) (1)

- **Minimum IPv6 MTU = 1280 octets**
  - In RFC 1883 minimum MTU = 576
- **On any link**
  - That cannot convey a 1280-octet packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below IPv6
- **Recommendations:**
  - Links with configurable MTU (e.g., PPP links) should be configured to have an MTU of at least 1500 octets
  - From each link to which a node is directly attached, the node must be able to accept packets as large as that link's MTU

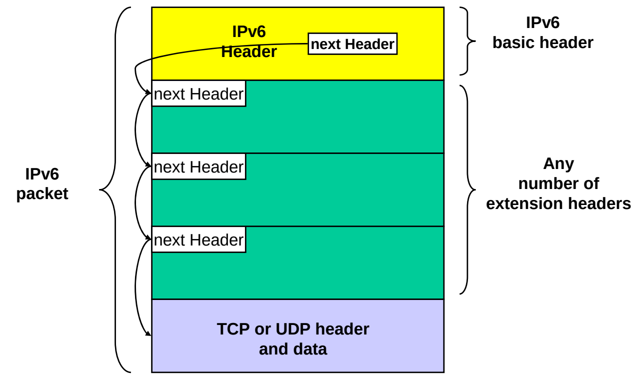
IPv6 MTU (RFC 2460) (2)

- **Recommendations cont.:**
  - IPv6 nodes should implement Path MTU Discovery (RFC 1981) in order to discover and take advantage of path MTUs greater than 1280 octets
  - However, a minimal IPv6 implementation (e.g., in a boot ROM) may simply restrict itself to sending packets no larger than 1280 octets, and omit implementation of Path MTU Discovery
  - In order to send a packet larger than a path's MTU, a node may use the IPv6 Fragment header to fragment the packet at the source and have it reassembled at the destination
  - However, the use of such fragmentation is discouraged in any application that is able to adjust its packets to fit the measured path MTU (i.e., down to 1280 octets)

Agenda

- **History**
- **IPv6**
  - IPv6 Facts
  - Review IPv4 Header
  - IPv6 Main Header
  - **IPv6 Extension Headers**
  - Security
  - Addressing
- **ICMPv6 and Plug&Play**
- **Routing**
- **Transition**

Daisy Chain of Extension Headers



Important IPv4 Protocol-Types and IPv6 Extension Header Types (1)

0	HOPOPT/ HBH	Hop by hop options (IPv6)
1	ICMP	Internet Control Message (IPv4)
2	IGMP	Internet Group Management (IPv4)
4	IPv4	IPv4 encapsulation (used e.g. by Mobile IP)
6	TCP	Transmission Control
17	UDP	User Datagram
41	IPv6	IPv6 encapsulation (used e.g. by Mobile IP)
43	IPv6-Route / RH	Routing Header (IPv6)
44	IPv6-Frag / FH	Fragmentation Header (IPv6)
46	RSVP	Resource Reservation Protocol
47	GRE	Generic Route Encpsulation
50	ESP	Encrypted Security Payload
51	AH	Authentication Header
55	Mobile	Mobility in IPv4
58	IPv6-ICMP	ICMPv6
59	IPv6-NoNxt	No next Header for IPv6
60	IPv6-Opts / DO	Destination Options for IPv6

© 2012, D.I. Lindner

Important IPv4 Protocol-Types and IPv6 Extension Header Types (2)

89	OSPF	Open Shortest Path First for IPv4 and IPv6
103	PIM	Protocol Independent Multicast Routing
112	VRRP	Virtual Router Redundancy Protocol Version 3 for IPv4 and IPv6
115	L2TP	Layer Two Tunneling Protocol
136	Mobility Header	Mobility in IPv6
137	MPLS in IP	Encapsulating MPLS in IP or GRE (RFC 4023)
139	HIP	Host Identity Protocol (RFC 5201)
140	Shim6	Shim6: Level 3 Multihoming Shim Protocol for IPv6 (RFC 5553)
253		Use for experimentation and testing (RFC 3692)
254		Use for experimentation and testing (RFC 3692)
255	Reserved	by IANA

For a complete list look to:
http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml

note: RFC 1700 Assigned Numbers -> Historic -> moved to Online database www.iana.org/protocols

Extension Header Handling (1)

- Extension headers are not examined or processed by any node along a packet's delivery path, until the packet reaches the node identified in the Destination Address field of the IPv6 header.
  - Only exception -> Hop-by-Hop Options Header
- The contents and semantics of each extension header determine whether or not to proceed to the next header. Therefore, extension headers must be processed strictly in the order they appear in the packet.
- The Hop-by-Hop Options header carries information that must be examined and processed by every node along a packet's delivery path, including the source and destination nodes. The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header. Its presence is indicated by the value zero in the Next Header field of the IPv6 header.
- If, as a result of processing a header, a node is required to proceed to the next header but the Next Header value in the current header is unrecognized by the node
  - it should discard the packet and send an ICMP Parameter Problem message to the source of the packet, with an ICMP Code value of 1 ("unrecognized Next Header type encountered") and the ICMP Pointer field containing the offset of the unrecognized value within the original packet.
  - The same action should be taken if a node encounters a Next Header value of zero in any header other than an IPv6 header.

© 2012, D.I. Lindner

Extension Header Handling (2)

- **A full implementation of IPv6 includes implementation of the following extension headers:**
  - Hop-by-Hop Options
  - Routing (Type 0)
  - Fragment
  - Destination Options
  - Authentication
  - Encapsulating Security Payload
- **When more than one extension header is used in the same packet, it is recommended that those headers appear in the following order:**
  - IPv6 header
  - Hop-by-Hop Options header
  - Destination Options header (note 1)
  - Routing header
  - Fragment header
  - Authentication header (note 2)
  - Encapsulating Security Payload header (note 2)
  - Destination Options header (note 3)
  - Upper-layer header

Extension Header Handling (3)

- **note 1: for options to be processed by the first destination that appears in the IPv6 Destination Address field plus subsequent destinations listed in the Routing header**
- **note 2: additional recommendations regarding the relative order of the Authentication and Encapsulating Security Payload headers are given in [RFC-4302 AH, RFC 4303 ESP]**
- **note 3: for options to be processed only by the final destination of the packet**
- **Each extension header should occur at most once, except for the Destination Options header which should occur at most twice (once before a Routing header and once before the upper-layer header)**
- **If the upper-layer header is another IPv6 header (in the case of IPv6 being tunneled over or encapsulated in IPv6), it may be followed by its own extension headers, which are separately subject to the same ordering recommendations**
- **IPv6 nodes must accept and attempt to process extension headers in any order and occurring any number of times in the same packet, except for the Hop-by-Hop Options header which is restricted to appear immediately after an IPv6 header only**
- **Nonetheless, it is strongly advised that sources of IPv6 packets adhere to the above recommended order until and unless subsequent specifications revise that recommendation**

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Routing Header (RH) (RFC 2460)

- **Routing Extension Header:**
  - Lists one or more intermediate nodes to be visited
  - Designed to support SDRP (source demand routing protocol
    - Policy routing between Internet Routing Domains
  - Designed to support Mobile IP
    - A host can keep his home-IP address when connected to a foreign network
  - Very similar to old source routing option of IPv4
    - <u>Loose source routing combined with record route</u>
  - A node will only look at RH if one of its own IP addresses is recognized in the IPv6 destination address field
  - **<u>Next header value</u>** of immediately preceding header = **<u>43</u>**

Routing Header (RFC 2460)

	0	8	16	24 31
0	Next Header	Header Ext. Length	Routing Type = 0	Segment Left
4	reserved			
8	Address [0]			
24	Address [1]			
40				

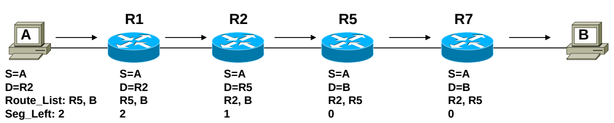
………..
up to Address [Header Ext. Length/2 - 1]

Routing Header (RFC 2460)

- **Extension header length**
  - Number of 64-bit words
  - Two times the number of addresses in the list
  - Up to 24 nodes could be specified as segments in the list
- **“Segment Left” is used as pointer to the next to be visited node**
  - This address is used as next IPv6 destination address
    - The corresponding address of RH and the current IPv6 destination address are swapped
  - Segment Left is decremented
  - Number of listed nodes that still have to be traversed before reaching the final destination
    - Note: Segment Left acts as pointer from the end of the segment list

Example for RH Type-0 Usage

- **Contains “segments” (= next hops) and “counter” (= segments left)**
- **Next-hop list is decremented**
- **New DA = next hop (“segment”)**
- **DA seen by receiving hop stored in segment list**



© 2012, D.I. Lindner

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Attention !!!

- **RH Type-0 is deprecated in RFC 5095 because of security aspects**
  - A single RH0 may contain multiple intermediate node addresses, and the same address may be included more than once in the same RH0
  - This allows a packet to be constructed such that it will oscillate between two RH0-processing hosts or routers many times and hence may act as a denial-of-service mechanism.
- **For Mobile IPv6 (RFC 6275) a new RH Type 2 was created**
  - To allow the packet to be routed directly from a correspondent to the mobile node's care-of address. The mobile node's care-of address is inserted into the IPv6 Destination Address field. Once the packet arrives at the care-of address, the mobile node retrieves its home address from the routing header, and this is used as the final destination address for the packet. This routing header type (type 2) is restricted to carry only one IPv6 address.

Mobility Support for IPv6

- **RFC 6275**
  - Creates a new IPv6 protocol Mobility Header (next header =135) carrying the following messages depending on MH Type
    - Home Test Init (Type 1)
    - Home Test (Type 3)
    - Care-of Test Init (Type 2)
    - Care-of Test (Type 4)
    - Binding Update (Type 5)
    - Binding Acknowledgement (Type 6)
    - Binding Refresh Request (Type 0)
    - Binding Error (Type 7)
  - New IPv6 destination option
    - Home Address Option (Option Type 201 = 0xC9)
  - New IPv6 ICMP messages
    - Home Agent Address Discovery Request
    - Home Agent Address Discovery Reply,
    - Mobile Prefix Solicitation
    - Mobile Prefix Advertisement

© 2012, D.I. Lindner

Extension Header for Options

- **Two ways to encode optional destination information in IPv6**
  - Destination option header (DO)
  - Hop-by-hop option header (HBH)
- **These headers carry a variable number of type-length-value (TLV) encoded "options“ of the following format:**

Option Type	Option Data Length	Option Data ....

Option Type:

0 1	2	3 7
Action Bits	C	Number

Action Bits: 0 0 ... skip over this option
0 1 ... discard the packet, no ICMPv6 report
1 0 ... discard the packet, send ICMPv6 report even if multicast
1 1 ... discard the packet, send ICMPv6 report if not multicast

C Bit (change en route): 0 ... option does not change en-route
1 ... option may change en-route

DO / HBH Extension Header

	0	8	16	24 31
0	Next Header	Header Ext. Length	Option Type	Option Data Length
4	Option Data			
8	Option Type	Option Data Length	Option Data	

- A particular option is identified by a full 8-bit option type, not just the low-order 5 bits of an option type
- The same option type numbering space is used for both the HBH and DO header
- Individual options may have specific alignment requirements, to ensure that multi-octet values within option Data fields fall on natural boundaries
- Currently defined options are for padding only to align gap between options properly (32 bit alignment)
  - PAD1 (type = 0) … null byte to be included
  - PADN (type = 1) … specifies number of null bytes to be included

© 2012, D.I. Lindner

Destination Options (DO)

- **Destination Options Extension Header:**
  - Specifies one or more options which are processed only by the end-system specified in IP destination address field
  - Used for adding functionality to IPv6 later
    - E.g. Mobile IP together with routing extension header
  - Option may be known or not know to receiver
    - if unknown ⇨ action-bits specify what to do
  - Extension header length
    - number of 64-bit words
  - **Next header value** of immediately preceding header = **60**

Hop-by-Hop Options (HBH)

- **Hop-by-Hop Extension Header:**
  - Same format as the DO header
  - Specifies one or more options which are processed by every intermediate system (router)
  - May be used for adding management/debugging functions later to IPv6
  - Option may be known or not know to receiver
    - if unknown ⇨ action-bits specify what to do
  - Here C-bits are used by every hop
  - **Next header value** of immediately preceding header = **0**

© 2012, D.I. Lindner

IPv6 Jumbo Payload Option (HBH)

- **A "Jumbogram"**
  - Is an IPv6 packet containing a payload longer than 65,535 octets
  - Payload length in IPv6 basic header only 16 bits
  - Note: Jumbograms are relevant only to IPv6 nodes that may be attached to links with a link MTU greater than 65,575 octets, and need not be implemented or understood by IPv6 nodes that do not support attachment to links with such large MTU
- **Originally defined in RFC 1883 as one valid HBH option**
- **Currently defined in separate RFC 2675**
  - Option type = 194 (**110** 00010 or 0xC2)
  - Length of Jumbo specified in option data field
  - TCP / UDP extension to make use of jumbos

Example for Jumbogram

	0	8	16	24 31
0	Next Header	Header Ext. Length	194 (**110** 00010)	4
4	Jumbo Payload Length			

Option Type:

0 1	2	3 7
Action Bits	C	Number

Action Bits: 0 0 ... skip over this option
0 1 ... discard the packet, no ICMP report
1 0 ... discard the packet, send ICMP report even if multicast packet
<u>1 1 ... discard the packet, send ICMP report if not multicast packet</u>

C Bit: <u>0 ... option does not change en-route</u>
1 ... option may change en-route

© 2012, D.I. Lindner

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

IPv6 Router Alert Option (HBH)

- **One base idea of IPv6 header**
  - Is rapidly forwarding of regular datagrams
- **Buts some protocols (e.g. RSVP)**
  - Use control datagrams which, while addressed to a particular destination, contain information that needs to be examined, and in some case updated, by routers along the path between the source and destination.
  - Currently, however, the only way for a router to determine if it needs to examine a datagram is to at least partially parse upper layer data in all datagrams; this parsing is expensive and slow
- **Router Alert option**
  - HBH option defined in RFC 2711
  - The presence of this option in an IPv6 datagram informs the router that the contents of this datagram needs to be examined / handled by routers along the path to the destination although not addressed to any of the routers
  - The absence of this option in an IPv6 datagram informs the router that the datagram does not contain information needed by the router and hence can be safely routed without further datagram parsing.
  - option-type = 5 (**000** 00101) or 0x5
  - protocol like RSVP can benefit from this option
  - no additional performance penalty on forwarding normal datagrams

Example for Router Alert Option (HBH)

	0	8	16	24 31
0	Next Header	Header Ext. Length	5 (**000** 00101)	2
4	Value			

Action Bits: <u>0 0 ... skip over this option</u>
0 1 ... discard the packet, no ICMP report
1 0 ... discard the packet, send ICMP report even if multicast packet
1 1 ... discard the packet, send ICMP report if not multicast packet

C Bit: <u>0 ... option does not change en-route</u>
1 ... option may change en-route

Value: **0 ... Datagram contains a Multicast Listener Discovery message [RFC-2710]**
1 ... Datagram contains RSVP message
2 ... Datagram contains an Active Networks message
3-65535 Reserved to IANA for future use

© 2012, D.I. Lindner

Fragment Header (FH)

- **Fragment Extension Header:**
  - IPv6 routers do not fragment oversized packets
    - Implicitly act like IPv4 routers on receiving a datagram with “Do not Fragment” bit set to one
  - IPv6 host must use MTU path discovery in order to select correct MTU size
  - But IPv6 host (source node) can fragment packets before they are sent in the network using FH
    - Identification field, fragment offset field, “More” fragment-bit used in the same way as in IPv4
  - Each fragment may be routed independently
  - Destination must reassemble fragments
  - **Next header value** of immediately preceding header = **44**

Fragment Header (FH)

	0	8	16	24	29	31
0	Next Header	Reserved	Fragment Offset		Res.	M
4	Identification					

Fragment Offset (13 bits) ... pointer to location of fragment in original packet

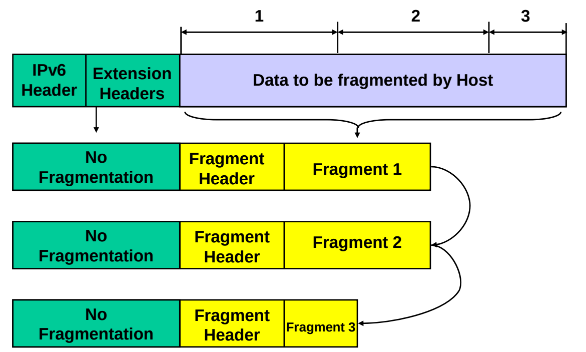
Identification (32 bits) ... unique ID for the original packet, same value in every fragment of original packet

M Bit: 0 ... last fragment
1 ... more fragments

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Host Fragmentation



No Fragmentation – contain the IPv6 basis header + optional extension headers HPH, (DO) up to RH

Agenda

- **History**
- **IPv6**
  - IPv6 Facts
  - Review IPv4 Header
  - IPv6 Main Header
  - IPv6 Extension Headers
  - **Security**
  - Addressing
- **ICMPv6 and Plug&Play**
- **Routing**
- **Transition**

IP Security Discussion Raise with IPv6

- **End-to-end security**
  - Will become more and more important when Internet goes to the commercial world
- **Question was**
  - If the next generation IP protocol (IPv6) should provide end-to-end security as integral part of itself
- **Basic building blocks for end-to-end security**
  - Authentication and integrity
    - Provides identity of sender
    - Senders message was not changed on the way through the network
  - Confidentiality or Privacy
    - Message cannot be read by others than authorized receiver
  - Non-repudiation
    - The sender cannot later repudiate the contents of the message
    - Protection of the receiver

IPv6 Security Aspects

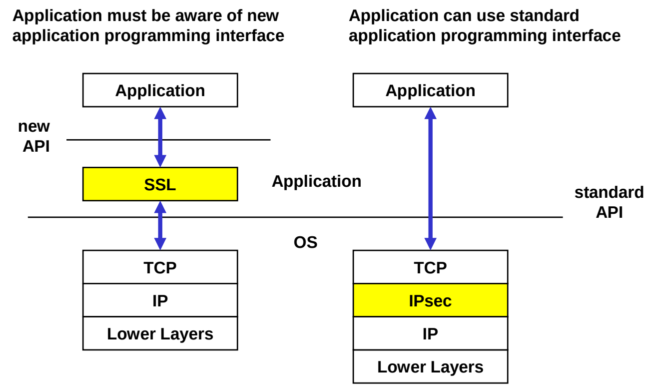
- **After heated discussions IESG decided**
  - Basic building blocks (without non-repudiation) of network security should be part of IPv6 functionality
  - A vendor of an IPv6 implementation must include support of these basic building blocks in order to be standard-compliant
    - Does not mean that the use of authentication and encryption blocks is required; only support must be guaranteed
  - IPv6 security follows the general IPsec recommendations
    - RFC 3401 (obsoletes RCF 2401 obsoletes RFC 1825) Security Architecture for IP (IPv4 and IPv6)
  - Difference of security aspects between IPv4 and IPv6
    - Security in IPv6 is an integral part of it
    - Security in IPv4 is an add on

© 2012, D.I. Lindner

Security Architecture for IP

- **The goal of the IPsec architecture**
  - Provision of various security services for traffic at the IP layer in both IPv4 and IPv6 environments
  - In a standardized and universal way
  - „Security Framework“
- **Before IPsec**
  - Existing solutions were mostly on the application layer (SSL, S/MIME, ssh, …)
  - Existing solutions on the network layer were all propriety
    - E.g. it was complicated, time demanding and expensive to establish multi-application or multi-vendor virtual private networks (VPNs)

Which Layer for Security?



© 2012, D.I. Lindner

Elements of IPsec (1)

- **IP Security Architecture**
  - Defined in RFC 4301
    - (Obsoletes RFC2401) (Updates RFC3168) (Updated by RFC6040) (Status: PROPOSED STANDARD)
  - Describes how to provide a set of security services for traffic at the IP layer
  - Describes the requirements for systems that implement IPsec, the fundamental elements of such systems, and how the elements fit together and fit into the IP environment
  - It also describes the security services offered by the IPsec protocols, and how these services can be employed in the IP environment.
  - Terms defined:
    - **Security Associations** (SA)
      - What they are and how they work, how they are managed and their associated processing
    - **Security Policy Database** (SPD)
    - **Security Association Database** (SAD)

Elements of IPsec (2)

- **Security Protocols (for traffic security)**
  - Authentication Header (AH)
    - Defined in RFC 4302
      - (Obsoletes RFC2402) (Status: PROPOSED STANDARD)
  - Encapsulating Security Payload (ESP)
    - Defined in RFC 4303
      - (Obsoletes RFC2406) (Status: PROPOSED STANDARD)
- **Algorithms for authentication and encryption**
  - Secret-key algorithms are used so far because of performance reasons
    - HMAC-SHA1, HMAC-MD5, AES-XCBC-MAC, DES-CBC, 3DES-CBC, AES-CBC, AES-CTR
  - Defined in many separate RFCs
    - see RFC 4835 Cryptographic Algorithm Implementation Requirements for ESP and AH (Obsoletes RFC4305) (Status: PROPOSED STANDARD)

Datenkommunikation 384.081 - SS 2012

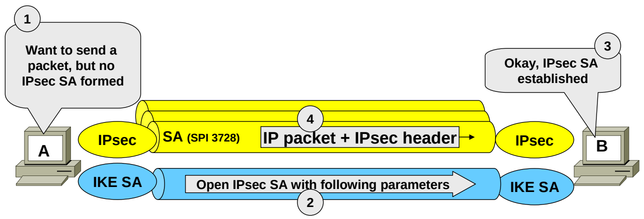
Appendix 5 - IPv6 (v4.7)

Elements of IPsec (3)

- **Management of Security Associations and Keys**
  - Manual for static and small environments
  - Automatic for scalable environments by IKE / ISAKMP
  - Internet Key Exchange (IKEv1) for ISAKMP
    - RFC 2407 The Internet IP Security Domain of Interpretation for ISAKMP (Obsoleted by RFC4306)
    - RFC 2408 Internet Security Association and Key Management Protocol (ISAKMP (Obsoleted by RFC4306)
    - RFc 2409 The Internet Key Exchange (IKE) (Obsoleted by RFC4306) (Updated by RFC4109)
  - Internet Key Exchange (IKEv2)
    - RFC 4306 Internet Key Exchange Protocol Version 2 (IKEv2) (Obsoletes RFC2407, RFC2408, RFC2409) (Obsoleted by RFC5996) (Updated by RFC5282)
    - RFC 5996 Internet Key Exchange Protocol Version 2 (IKEv2) (Obsoletes RFC4306, RFC4718) (Updated by RFC5998) (Status: PROPOSED STANDARD)

IKE General Aspects

- **Establishes an authenticated and encrypted tunnel**
  - IKE SA main mode (bidirectional), UDP port 500
- **Creates unidirectional IPsec SAs on demand**
  - Also keys are exchanged



What IPsec does? (1)

- **IPsec enables a system**
  - To select required security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services
- **IPsec can be used**
  - To protect one or more "paths" between a pair of hosts, between a pair of security gateways, or between a security gateway and a host
  - Security gateway could be for example, a router or a firewall implementing IPsec
    - VPN concentrator is another name for such a device if several SA pairs are terminated at the same point

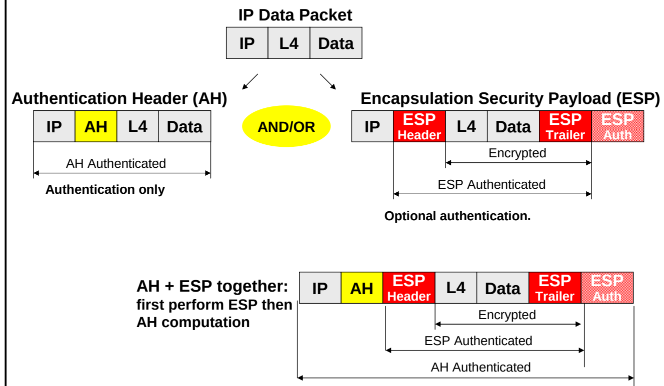
What IPsec does? (2)

- **The set of security services that IPsec can provide includes**
  - Access control
    - Prevents unauthorized use of a resource
    - The resource to which access is being controlled is
      - for a host -> computing cycles or data
      - for a security gateway -> network behind the gateway or bandwidth on that network
  - Connectionless integrity
    - detects modification of individual IP datagram's
  - Data origin authentication
  - Rejection of replayed packets (optional)
    - detects arrival of duplicate IP datagram's within a constrained window
  - Confidentiality (encryption)
  - All these services are provided at the IP layer
    - hence they can be used by any higher layer protocol e.g., TCP, UDP, ICMP, BGP, etc.

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

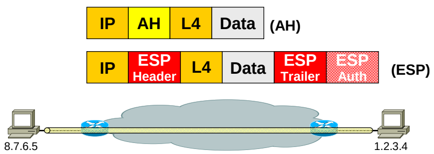
IPsec Headers



IPsec Modes (1)

Transport Mode

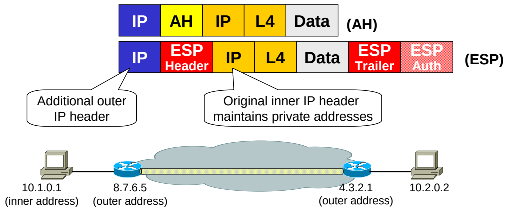
- Only one IP header.
- Used for end-to-end sessions
- Does not hide communication statistics because of network header (IP addresses of the end systems) is sent in clear text



IPsec Modes (2)

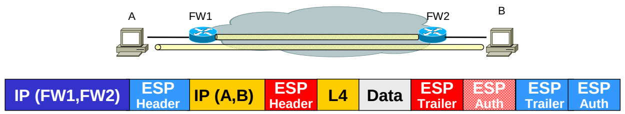
Tunnel Mode
for Site-to-Site VPN

- Whole original IP packet is IPsec-encapsulated.
- Used for VPNs.
- Does hide traffic patterns!

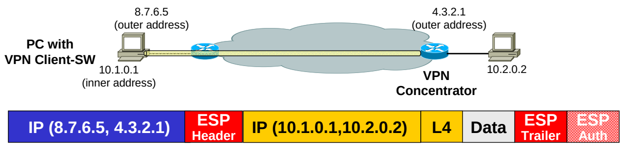


IPsec Modes (3)

Tunnel Mode and
Transport Mode (ESP only)



Tunnel Mode for
Client-to-Site VPN



© 2012, D.I. Lindner

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

IPsec in Praxis

- **"IPsec used anywhere"**
  - Firewall, Router, Hosts
  - VPN
    - Site-to-Site
    - Remote-to-Site
    - Client-to-Site
  - Scalable solutions available
  - Easy to implement
  - Defined for end-to-end security but not frequently used between end systems
- **Encryption performance**
  - Original standards: DES and Triple-DES
  - Today migration to AES already done (more efficient, longer keys)
  - HW versus SW encryption power
    - E.g. hardware crypto engines on router for higher performance

AH Security Service (RFC 4302)

- **AH provides**
  - IP datagram sender authentication by HMAC or MAC
  - IP datagram integrity assurance by HMAC or MAC
  - Replay detection and protection via sequence number (optional)
- **AH does not provide**
  - Non-repudiation because of usage of secret-keys (shared keys) for HMAC or MAC
    - note: Digital Signature needs usage of public-key technique by signing a message with the private-key
  - Confidentiality (encryption)
  - Authentication for IP fragments
    - therefore IP fragments must be assembled before authentication is checked (better avoid it by MTU path discovery)

© 2012, D.I. Lindner

IPv4 and AH

0	4	8		16		31
Vers.=4	HLEN	ToS or DSCP		Total Length		
Fragment Identifier				Flags	Fragment Offset	
TTL		protocol = 51		Header Checksum		
Source Address						
Destination Address						
IP Options						Pad
First 32 bits of AH						
....						
Last 32 bits of AH						
Payload						
....						

Authentication Header (AH)

	0	8	16	24	31
0	Next Header	Length	Reserved		
4	Security Parameters Index (SPI)				
8	Sequence Number				
10	Authentication Data (variable number of 32-bit words)				

note: AH was originally defined as extension header for IPv6 and later same structure was used for IPv4

IPv6: next header value of immediately preceding header = 51

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Authentication Header (AH)

- **Next Header (8 bits)**
  - Indicates the next header following the AH header
  - Same values allowed as protocol field in IPv4 header
    - IP in IP (4), TCP (6), UDP (17), ICMP (1), OSPF (89), etc
    - Next header value of immediately preceding header = 51 (AH)
- **Length**
  - Length of AH header
    - Number of 32-bit words
- **Security Parameter Index**
  - A 32-bit number identifying (together with IP destination address) the security association for this IP datagram
  - SPI value 0 is reserved for local implementation specific use and must not be sent on the wire

Authentication Header (AH)

- **Sequence number:**
  - Monotonically increasing counter value (mandatory and always present)
  - Defined in RFC 2085
  - Prevention against replay attacks enabled by default
  - Mandatory for transmitter but the receiver need not act upon it
  - Every new SA resets this number to zero (thus first packet = 1), no cycling: after sending the 2^{32nd} packet, a new SA must be established.

Authentication Header (AH)

- **Authentication Data:**
  - Contains Integrity Check Value (ICV)
    - All fields behind AH header plus predictable field of IP header before AH (e.g. TTL, checksum, ToS/DSCP regarded to be zero for ICV calculation)
  - The algorithm for authentication is free and must be negotiated
  - Mandatory default calculation of the authentication data must be supported
    - See RFC 4835 for details

Requirement	Algorithm
MUST	HMAC-SHA1-96 [RFC2404]
SHOULD+	AES-XCBC-MAC-96 [RFC3566]
MAY	HMAC-MD5-96 [RFC2403]

  - Non-repudiation (IP datagram signing) is not supported!

IPv4 and ESP

0	4	8	16		31
Vers.=4	HLEN	ToS	Total Length		
Fragment Identifier			Flags	Fragment Offset	
TTL		protocol = 50	Header Checksum		
Source Address					
Destination Address					
IP Options					Pad
ESP Header with ESP Parameters Encrypted Data ESP Trailer					

Encapsulating Security Payload (ESP)

	0	8	16	24	31
0	**Security Parameters Index (SPI)**				
4	**Sequence Number**				
8	**Payload Data Field (variable length) maybe starting with cryptographic synchronization data (e.g. Initialization Vector IV for DES-CBC)**				
			Padding (0-255 bytes)		
	Padding (0-255 bytes)		**Pad Length**	**Next Header**	
	ESP Authentication Data (optional)				

note: ESP was originally defined as extension header for IPv6 and later same structure was used also for IPv4
IPv6: next header value of immediately preceding header = 50

ESP Header, Payload

- **SPI and Sequence Number**
  - Used for same functions as in the AH header
    - Defining SA and prevention of replay attack
  - This are the only fields of ESP transmitted in cleartext
- **Payload Field of ESP is encrypted**
  - Actual format depends on encryption method
    - E.g location of Initialization Vector (IV) for DES-CBC
    - Note: every IP datagram must contain an IV because IP datagrams may arrive out of sequence

ESP Trailer

- **Padding Field**
  - Is used to fill the plaintext to the size required by the encryption algorithm (e.g. the block size of a block cipher)
  - Is used to align 4 byte boundaries
- **Pad Length**
  - Pointer to end of data
- **Next Header**
  - Identifies the type of data contained in the Payload Data Field, e.g., an extension header in IPv6 or an upper layer protocol identifier
    - Same values allowed as protocol field in IPv4 header

ESP Encryption Methods

- **Mandatory default transformation of the data**
  - See RFC 4835 for details

Requirement	Encryption Algorithm
MUST	NULL [RFC2410]
MUST	AES-CBC with 128-bit keys [RFC3602]
MUST-	TripleDES-CBC [RFC2451]
SHOULD	AES-CTR [RFC3686]
SHOULD NOT	DES-CBC [RFC2405]
MUST	HMAC-SHA1-96 [RFC2404]
SHOULD+	AES-XCBC-MAC-96 [RFC3566]
MAY	NULL
MAY	HMAC-MD5-96 [RFC2403]

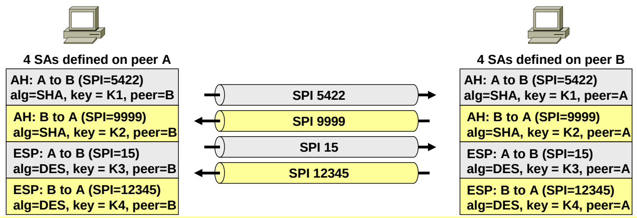
- **Null Encryption:**
  - See RFC 2410 where it is praised for ease of implementation, great speed and simplicity ;-)

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Security Associations Example

- SAs are unidirectional !
  - Thus multiple SAs are typically established between two peers
  - One SA per direction and so called transform (AH or ESP method)
  - Security policies can be totally asymmetric !
  - Identified by the Security Parameter Index (SPI) and peer's IP address
    - SPI is a 32-bit value



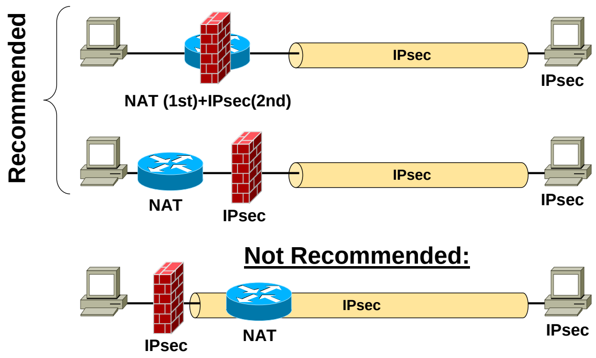
NAT and IPsec AH, ESP

- **AH hash includes the whole IP header**
  - Cannot work together with NAT
- **ESP**
  - Transport mode: authentication excludes IP but not TCP/UDP header for hash calculation!
    - but TCP checksum includes the Pseudo-IP header
    - therefore turn off TCP checksum verification in the receiver
  - Tunnel mode: Outer IP header is neither encrypted nor authenticated – no problems with NAT
  - note:
    - N(P)AT (NAT with port address translation) will modify TCP port numbers
    - if TCP + Payload is ESP encrypted that is not possible
    - propriety Cisco solution -> encapsulate ESP in UDP or TCP
- **See RFC 3715 NAT Traversal for ongoing work**

NAT and IPsec IKE

- **Internet Key Exchange (IKE)**
  - Problem if exchanged keys or certificates are bound to gateway's IP address
  - Avoid it by using other identifier of the endpoint e.g. User-ID or FQDN
- **Expiration of Security Association (SA)**
  - Re-key request is sent to the initial UDP port 500
  - Problems with multiple security gateways behind a N(P)AT device

Apply NAT before IPsec !



Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Problems with IPsec / IKEv1 (1)

- **IPsec for Site-to-Site VPN**
  - Often uses pre-shared secrets for authentication of IKE peers
  - Why?
    - certificates means maintaining a PKI (Public Key Infrastructure)
      - at least a private CA (Certification Authority) server is needed
    - VPN router/concentrator can often be physically protected
- **IPsec for Client-to-Site VPN**
  - Different situation
    - Mobile PCs calling from insecure places
    - Pres-hared secret may be compromised hence configuration and maintenance overhead if number of clients is high
  - Therefore combination of IPsec, well-known RAS Authentication Techniques (PPP with EAP, RFC 3748) and X-AUTH
    - Client dials-in, authenticates itself at a authentication server (VPN concentrator) and then the necessary IPsec configuration is pushed from the VPN concentrator to the client
      - sometimes even enhanced with activation of a host based FW function at the client side of IPsec
      - Client gets an IP address from the VPN concentrator and all client traffic may be forced to go exclusively to the VPN concentrator
    - solved with X-AUTH exchange as add-on to IKEv1
      - X-AUTH exchange is an inherent optional part of IKEv2
  - IPsec Tunnel mode is used

Problems with IPsec / IKEv1 (2)

- **IPsec for End-to-End VPN (Client-to-Client VPN in transport mode**
  - Some inherent problems when using certificates caused by the fact that certificates normally deals with names but IPsec (especially the SPD) knows only about IP addresses
  - Binding of certificates to IP addresses not possible or not wanted
- **See the following documents for more information about some basic problems of IPsec and IKEv1:**
  - **"A Cryptographic Evaluation of IPsec"**
    - **Niels Ferguson and Bruce Schneier**
    - **-> http://www.schneier.com/paper-ipsec.pdf**
  - **"Experiences with Host-to-Host IPsec"**
    - **Tuomas Aura, Michael Roe, and Anish Mohammed**
    - **http://research.microsoft.com/users/tuomaura/Publications/aura-roe-mohammed-protocols05.pdf**
  - **"Key Exchange in IPSec: Analysis of IKE"**
    - **Charlie Kaufman, Radia Perlman**
    - **IEEE Internet Computing Volume 4 Issue 6 (2000) page50-56**
    - http://ieeexplore.ieee.org/iel5/4236/19367/00895016.pdf?isnumber=19367&prod=JNL&arnumber=895016&arSt=50&ared=56&arAuthor=Perlman%2C+R.%3B+Kaufman%2C+C.
  - **"Analysis of the IPsec Key Exchange"**
    - **Charlie Kaufman, Radia Perlman**
    - http://krypt1.cs.uni-sb.de/teaching/seminars/ws2001/literature/PerKau2001.pdf
- **This leads to an adaptation of the original IPsec RFCs (IPsec Architecture, AH and ESP Headers) and to an completely rework of the IKE protocol -> IKEv2**

Agenda

- **History**
- **IPv6**
  - IPv6 Facts
  - Review IPv4 Header
  - IPv6 Main Header
  - IPv6 Extension Headers
  - Security
  - **Addressing**
- **ICMPv6 and Plug&Play**
- **Routing**
- **Transition**

IPv6 Addresses (1)

- **Same principle as for the classic IPv4 addresses**
  - 128 bit instead of 32
  - Identify individual interfaces (not nodes) and sets of interfaces
  - Structure
    - Prefix = Net-ID
    - Interface-ID = Host-ID
- **Multiple addresses may be assigned to an interfaces**
  - In order to facilitate routing or management
  - All interfaces are required to have at least one Link-Local unicast address
- **No broadcast addresses anymore!!!**
  - Such issues need to use multicast

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

IPv6 Addresses (2)

- **Three categories**
  - Unicast:
    - An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address
  - Anycast:
    - New concept appeared in IPv6 first
    - An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is **delivered to one of the interfaces** identified by that address (the "nearest" one, according to the routing protocol measure of distance)
  - Multicast:
    - An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is **delivered to all interfaces** identified by that address

IPv6 Addresses (3)

- **Notation**
  - Eight 16-bit pieces separated by colons x:x:x:x:x:x:x:x
  - Each piece x represented by one to four hexadecimal digits
    - FEDC:00b3:0000:0000:0000:34DE:7654:3210
  - Leading zeros in each hexadecimal component can be skipped
    - FEDC:b3:0:0:0:34DE:7654:3210
  - A set of consecutive null 16-bit numbers inside an address can be replaced by two colons
    - FEDC:b3::34DE:7654:3210
    - Double colon can only be used only once inside an address, because of uniqueness and should use to abbreviate longest series of 0s
  - Many ways for text representation allowed for flexibility
  - Later a canonical representation format was designed to avoid problems
    - **See RFC 5952**
    - "Recommendation for IPv6 Address Text Representation"

IPv6 Addressing Architecture (1)

- **Way to current address architecture:**
  - RFC 1884 IPv6 Addressing Architecture (Obsoleted by RFC2373) (Status: HISTORIC)
    - Defines the address architecture and the first allocation of addresses in the IPv6 space
  - RFC 2073 An IPv6 Provider-Based Unicast Address Format (Obsoleted by RFC2374)
  - RFC 2373 IPv6 Addressing Architecture (Obsoletes RFC1884) (Obsoleted by RFC3513)
  - RFC 2374 An IPv6 Aggregatable Global Unicast Address Format (Obsoletes RFC2073) (Obsoleted by RFC3587)
  - RFC 3513 IPv6 Addressing Architecture (Obsoletes RFC2373) (Obsoleted by RFC4291)
  - **RFC 3587** IPv6 Global Unicast Address Format (Obsoletes RFC2374) (Status: INFORMATIONAL)

IPv6 Addressing Architecture (2)

- **Way to current address architecture:**
  - **RFC 3879** Deprecating Site Local Addresses (Status: PROPOSED STANDARD)
  - **RFC 4193** Unique Local IPv6 Unicast Addresses (Status: PROPOSED STANDARD)
  - **RFC 4291** IPv6 Architecture (Obsoletes RFC3513) (Updated by RFC5952, RFC6052) (Status: DRAFT STANDARD)
  - **RFC 4941** Privacy Extensions for Stateless Address Autoconfiguration in IPv6 (Obsoletes RFC3041) (Status: DRAFT STANDARD)
  - **RFC 5375** IPv6 Unicast Address Assignment Considerations (Status: INFORMATIONAL)
  - **RFC 6164** Using 127-Bit IPv6 Prefixes on Inter-Router Links (Status: PROPOSED STANDARD)

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

IPv6 Initial Assignment (RFC 2373)

- **Nobody could be certain**
  - That we know the best way to assign addresses nowadays
- **Therefore IPv6 address allocation**
  - Should leave enough room to extensions or new developments
  - Address types are introduced
- **Address type**
  - Is indicated by the leading bits in the address
  - **IPv6 Format Prefix (FP)**

Initial IPv6 Prefix Allocation (RFC 2373)

Allocation	Format Prefix (binary)	Fraction of Address Space
Reserved	0000 0000	1/256
Unassigned	0000 0001	1/256
Reserved for NSAP allocation	0000 001	1/128
Reserved for IPX allocation	0000 010	1/128
Unassigned	0000 011	1/128
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Aggregatable global unicast address	001	1/8
Unassigned	010	1/8
Unassigned	011	1/8
Unassigned	100	1/8

Initial IPv6 Prefix Allocation (RFC 2373) cont.

Allocation	Format Prefix (binary)	Fraction of Address Space
Unassigned	101	1/8
Unassigned	110	1/8
Unassigned	1110	1/16
Unassigned	1111 0	1/32
Unassigned	1111 10	1/64
Unassigned	1111 110	1/128
Unassigned	1111 1110 0	1/512
Link local-use addresses	1111 1110 10	1/1024
Site local-use addresses	1111 1110 11	1/1024
multicast addresses	1111 1111	1/256

IPv6 Initial Addressing Plan Ideas (1)

- **Current growth of Internet means**
  - Explosion of the routing tables
- **Addressing should be done in a way**
  - To keep number of routing table entries of Internet core routers small
- **Route aggregation is necessary**
  - Prefix, length routing
  - Lessons learnt by CIDR
    - curbs the growth of the routing tables
- **The way to achieve this:**
  - **Aggregatable global unicast addresses**

Datenkommunikation 384.081 - SS 2012

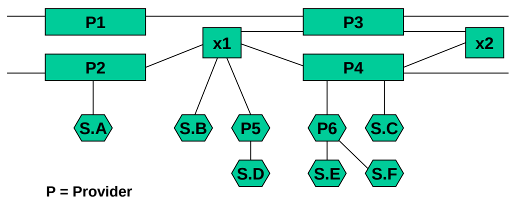
Appendix 5 - IPv6 (v4.7)

IPv6 Initial Addressing Plan Ideas (2)

- **Aggregatable global unicast addresses**
  - Based on provider based addressing (RFC 2073, RFC 2374, RFC 3587 – but provider opposed it)
    - RFC 2073, 2374 were on the standard track
    - RFC 3587 -> category: informational
  - Addresses are allocated from your provider
  - If customers want to change the provider
    - Their prefix changes
    - But renumbering of hosts, routers and sites has been included in the IPv6 protocol
  - The burden to handle this is on the customer
    - **Need for efficient auto-configuration of addresses**

IPv6 Initial Addressing Plan Ideas (3)

- Aggregatable global unicast addresses support both:
  - Provider based aggregation
  - Exchange based aggregation
- Possibility to create hierarchies
- Efficient routing aggregation



P = Provider
S = Subscriber
X = Exchange

Aggregatable Global Unicast Address (RFC 2374)

(FP) 3	13 bits	8 bits	24 bits	16 bits	64 bits
0 0 1	TLA ID	Res	NLA ID	SLA ID	Interface ID
Public Topology				Site Topology	Interface Identifier

Aggregatable Global Unicast Address
can be routed on the global Internet,
their uniqueness is guaranteed globally

RFC 2374

Format Prefix ...	3 bits
Top Level Aggregator ID	13 bits
Reserved	8 bits
Next Level Aggregator ID	24 bits
Site Level Aggregator ID	16 bits
Interface-ID ...	64 bits (EUI-64 – usually derived from MAC address)

Aggregatable Global Unicast Address (RFC 2374)

- **Top Level Aggregator (TLA)**
  - Public access points that interconnect service providers/telephone companies
  - IANA allocates these addresses
- **Next Level Aggregator (NLA)**
  - Large Internet service providers
  - NLA's assign to the next level
- **Site Level Aggregator (SLA)**
  - Called a subscriber; can be an organization, a company, a university, small ISP
  - They assign addresses to their users
  - SLA provide a block of contiguous addresses
- **Interface ID**
  - Host interface
  - IEEE has defined a 64 bit NIC address known as EUI-64
  - NIC driver for IPv6 will convert 48 bit NIC to 64 bit NIC

Global Unicast Address (RFC 3513, RFC 3587) (1)

n bits	m bits	128 – n – m bits
Global Routing Prefix	Subnet-ID	Interface ID
Public Topology	Site Topology	Interface Identifier

Global Routing Prefix is a (typically hierarchically-structured) value assigned to a site (a cluster of subnets/links)

Subnet ID is an identifier of a link within the site

Interface ID are used to identify interfaces on a link

global unicast addresses starting with binary 000 have no constraint on the size or structure of the interface ID field
(examples are the IPv6 address with embedded IPv4 addresses and the IPv6 address containing encoded NSAP addresses)

Global Unicast Address (RFC 3513, RFC 3587) (2)

n bits	m bits	64 bits
Global Routing Prefix	Subnet-ID	Interface ID
Public Topology	Site Topology	Interface Identifier

Global Routing Prefix is a (typically hierarchically-structured) value assigned to a site (a cluster of subnets/links)

Subnet ID is an identifier of a link within the site

Interface ID are used to identify interfaces on a link

global unicast addresses not starting with binary 000
have a 64-bit interface ID field,

Local Use Addresses

10-bits	54-bits	64-bits
F E C 0	Subnet ID	Interface ID

Site-local unicast

cannot be routed on the global Internet,
their uniqueness is guaranteed within a site
similar to RFC 1918 addresses like 10.0.0.0

10-bits	54-bits								64-bits
F E 8 0	0	0	0	0	0	0	0	0	Interface ID

Link-local unicast

defined only within a link and can only be used by stations
connected to the same link or the same local network

IEEE EUI-64 Identifier for Interface-ID

The only change needed to transform an IEEE EUI-64 identifier to an interface identifier is to invert the "u" (universal/local) bit. For example, a globally unique IEEE EUI-64 identifier of the form:

```
|0              1|1              3|3              4|4              6|
|0              5|6              1|2              7|8              3|
+----------------+----------------+----------------+----------------+
|cccccc0gcccccccc|ccccccccmmmmmmmm|mmmmmmmmmmmmmmmm|mmmmmmmmmmmmmmmm|
+----------------+----------------+----------------+----------------+
```

where "c" are the bits of the assigned company_id, "0" is the value of the universal/local bit to indicate global scope, "g" is individual/group bit, and "m" are the bits of the manufacturer- selected extension identifier.

The IPv6 interface identifier would be of the form:

```
|0              1|1              3|3              4|4              6|
|0              5|6              1|2              7|8              3|
+----------------+----------------+----------------+----------------+
|cccccc1gcccccccc|ccccccccmmmmmmmm|mmmmmmmmmmmmmmmm|mmmmmmmmmmmmmmmm|
+----------------+----------------+----------------+----------------+
```

Modified EUI-64 address

The only change is inverting the value of the universal/local bit.

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

IEEE 802 48-Bit MAC Address for Interface-ID

Links or Nodes with IEEE 802 48 bit MAC's defines a method to create a IEEE EUI-64 identifier from an IEEE 48bit MAC identifier. This is to insert two octets, with hexadecimal values of 0xFF and 0xFE, in the middle of the 48 bit MAC (between the company_id and vendor supplied id). For example, the 48 bit IEEE MAC with global scope:

```
|0              1|1              3|3              4|
|0              5|6              1|2              7|
+----------------+----------------+----------------+
|cccccc0gcccccccc|ccccccccmmmmmmmm|mmmmmmmmmmmmmmmm|
+----------------+----------------+----------------+
```

The IPv6 interface identifier would be of the form:

```
|0              1|1              3|3              4|4              6|
|0              5|6              1|2              7|8              3|
+----------------+----------------+----------------+----------------+
|cccccc1gcccccccc|cccccccc11111111|11111110mmmmmmmm|mmmmmmmmmmmmmmmm|
+----------------+----------------+----------------+----------------+
```

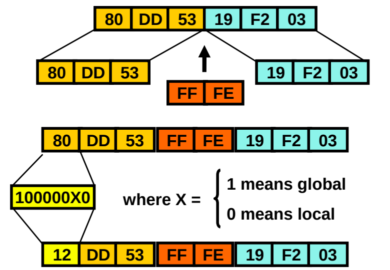
The only change is inverting the value of the universal/local bit.

Example MAC Address to Interface-ID

Ethernet MAC address (48bits, global, individual) taken as unique "Token"

Expansion to 64 bits

Modify U/L bit meaning of the first byte of a MAC address



Original IEEE meaning of first byte of a MAC address (hex 80 in our example):

100000 U/L I/G (bit 7 … 0)
- **U/L = 0 … global**
- U/L = 1 … local
- **I/G = 0 … individual**
- I/G = 1 … group

Privacy Concerns

- **By design, the interface identifier**
  - Is likely to be globally unique when generated in this fashion by SLAAC
- **If a notebook, smart-phone changes location**
  - The machine can be tracked based on the interface-ID of the IPv6 address
- **Dynamic IP addresses usage in IPv6**
  - Is not necessary because of huge IPv6 address space and therefore no NAT necessity
- **But IPv4 users get used**
  - To hide their client-machines behind a dynamic IP address
  - Kind of security by obscurity similar to NAT fairy tale
  - Home network users enjoying dynamic IP assignment by ISP´s PPP and NAT have the feeling that they could not identified so easily
    - That is an illusion

Privacy Extensions for IPv6

- **RFC 4941 specifies privacy extensions for SLAAC**
  - Global scope IPv6 address (especially the interface-ID) assigned by SLAAC can get some dynamical behavior again
  - Changing the interface identifier over time makes it more difficult for eavesdroppers and other information collectors to identify when different addresses used in different transactions actually correspond to the same node
  - For example the random interface identifier generation algorithm, as uses MD5 as the hash algorithm
  - Duplicate address detection (DAD) is mandatory

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Addresses with Embedded IPv4 Addresses

80-bits	16-bits	32-bits
00000000000000000000	0 0 0 0	IPv4 Address

IPv4-Compatible IPv6 Address (x:x:x:x:x:x:d.d.d.d)

used by hosts and routers which tunnel IPv6 packets dynamically over a IPv4 infrastructure (e.g.: ::193.170.150.1/96) (tunneling is one transition technique for IPv4 ⇨ IPv6)

80-bits	16-bits	32-bits
00000000000000000000	F F F F	IPv4 Address

IPv4-Mapped IPv6 Address (x:x:x:x:x:ffff:d.d.d.d)

represents address of IPv4-only hosts, used by hosts that do translation between IPv4 and IPv6 (e.g.: ::FFFF:193.170.150.1/80) (translation is another transition technique for IPv4 ⇨ IPv6)

see RFC 4038 for background on usage

Special Addresses / Anycast

- **Unspecified address:**
  - 0:0:0:0:0:0:0:0 or ::
  - can only be used as a source address by station that has not yet been configured with a regular address or as placeholder in some control messages
- **Loopback address:**
  - 0:0:0:0:0:0:0:1 or ::1
  - used by a node to send IPv6 packets to itself
- **Anycast:**
  - new concept in IPv6 to address a group of interfaces
  - is an address that is assigned to more than one interface (typically belonging to different nodes)

Anycast (1)

- **Anycast principle**
  - Instead of sending a packet to a specific server, one sends the packet to a generic address
  - This address will be recognized by all the servers of a given type (like multicast addressing)
  - Anycast addresses are allocated from the unicast address space (no special anycast format)
  - Thus, anycast addresses are syntactically indistinguishable from unicast addresses
  - When a unicast address is assigned to more than one interface - thus turning it into an anycast address - the nodes to which the address is assigned must be explicitly configured to know that it is an anycast address

Anycast (2)

- **Difference to multicasting**
  - For any assigned anycast address, there is a longest prefix P of that address that identifies the topological region in which all interfaces belonging to that anycast address reside
  - Within the region identified by P, the anycast address must be maintained as a separate entry in the routing system (commonly referred to as a "host route")
  - Outside the region identified by P, the anycast address may be aggregated into the routing entry for prefix P
  - The routing system is responsible to deliver the packet to the nearest of these servers
- **Have to be explicitly configured**
- **Only assigned to routers – not to end-stations**

Required Anycast Address: Subnet-Router Anycast Address

n bits	128 – n bits
Subnet Prefix	0000…………..00000

The "subnet prefix" in an anycast address is the prefix which identifies a specific link

This anycast address is syntactically the same as a unicast address for an interface on the link with the interface identifier set to zero

Packets sent to the Subnet-Router anycast address will be delivered to one router on the subnet

All routers are required to support the subnet-router anycast addresses for the subnets to which they have interfaces

The subnet-router anycast address is intended to be used for applications where a node needs to communicate with any one of the set of routers

Possible Anycast Usage

- **Anycast servers**
  - Traffics will be routed to nearest server of a given type (e.g.: time server, file server,dns server...)
- **Source selected policies**
  - A node can select which of several internet service providers it wants to carry its traffic
    - Configuring an anycast address to several routers of a given provider (several entry points)
    - Other anycast addresses configured for other providers
    - Specify anycast address in routing extension header (RH)
- **Fuzzy routing**
  - Sending a packet through one router of network X
  - First step in that direction
    - Subnet router anycast address

Multicast Address (RFC 3513)

8-bits	4-bits	4-bits	112-bits
1 1 1 1 1 1 1 1	Flags	Scope	Group ID
	0 0 0 T		

Multicast Address

T .... Transient
T = 0 ... permanently assigned (well known) multicast address, assigned by IANA
T = 1 ... non-permanently (transient) assigned multicast address

Scope:

0 ... reserved	8 ... organization-local scope
1 ... interface-local scope	9 ... unassigned
2 ... link local-scope	A ... unassigned
3 ... unassigned	B ... unassigned
4 ... admin-local scope	C ... unassigned
5 ... site-local scope	D ... unassigned
6 ... unassigned	E ... global scope
7 ... unassigned	F ... reserved

Multicast Scope Details (1)

- **Interface-local scope**
  - Spans only a single interface on a node, and is useful only for loopback transmission of multicast
- **Link-local scope**
  - spans the same topological region as the corresponding unicast scope
- **Site-local scopes**
  - is intended to span a single site
- **Global scope**
  - Spans the whole Internet
- **Admin-local scope is the smallest scope**
  - That must be administratively configured, i.e., not automatically derived from physical connectivity or other, non- multicast-related configuration
- **Organization-local scope**
  - Is intended to span multiple sites belongingto a single organization
- **Scopes labeled "unassigned"**
  - Are available for administrators to define additional multicast regions

Multicast Scope Details (2)

- **The meaning of a permanently-assigned multicast address**
  - is independent of the scope value
  - e.g. if the "NTP servers group" is assigned a permanent multicast address with a group ID of 101 (hex), then:
    - FF01:0:0:0:0:0:0:101
      means all NTP servers on the same interface as the sender
    - FF02:0:0:0:0:0:0:101 means all NTP servers on the same link as the sender
    - FF05:0:0:0:0:0:0:101
      means all NTP servers at the same site as the sender
    - FF08:0:0:0:0:0:0:101
      means all NTP servers at the same organization as the sender
    - FF0E:0:0:0:0:0:0:101
      means all NTP servers in the internet

Multicast Scope Details (3)

- **Non-permanently-assigned multicast addresses**
  - are meaningful only within a given scope
  - E.g. a group identified by the non-permanent, site-local multicast address FF15:0:0:0:0:0:0:101 at one site bears no relationship
    - To a group using the same address at a different site
    - Nor to a non-permanent group using the same group ID with different scope
    - Nor to a permanent group with the same group ID

Predefined Multicast Addresses

- **The following well-known multicast addresses are pre-defined:**
  - Reserved Multicast Addresses for future purposes:
    - FF00:0:0:0:0:0:0:0 up to FF0F:0:0:0:0:0:0:0
  - **All-Nodes Addresses** within scope 1 and 2
    - FF01:0:0:0:0:0:0:1
    - FF02:0:0:0:0:0:0:1
  - **All-Routers Addresses** within scope 1, 2 and 5
    - FF01:0:0:0:0:0:0:2
    - FF02:0:0:0:0:0:0:2
    - FF05:0:0:0:0:0:0:2

Predefined Multicast Addresses (cont.)

– FF01:0:0:0:0:0:0:5	OSPFIGP
– FF02:0:0:0:0:0:0:6	OSPFIGP Designated Routers
– FF02:0:0:0:0:0:0:9	RIP Routers
– FF02:0:0:0:0:0:0:A	EIGRP
– FF02:0:0:0:0:0:0:D	All PIM Routers
– FF02:0:0:0:0:0:0:12	VRRP
– FF02:0:0:0:0:0:0:12	All MLDv2 capable routers
– FF02:0:0:0:0:0:1:2	All DHCP Agents
– FF05:0:0:0:0:0:1:3	All DHCP Servers

- **Actual list of all assignments controlled by IANA**
  - www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml
  - www.iana.org/protocols/
  - see RFC 3307 for "Allocation Guidelines for IPv6 Multicast Addresses"

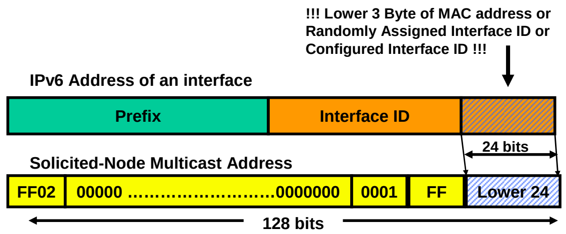
Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Predefined Multicast Addresses (cont.)

- **Solicited-Node Address**
  - FF02:0:0:0:0:1:FFXX:XXXX
  - Used within neighbor solicitation messages and for neighbor discovery
  - This multicast address is computed as a function of a node's unicast and anycast addresses
  - The solicited-node multicast address is formed by taking the low-order 24 bits of the address (unicast or anycast) and appending those bits to the 104-bit prefix FF02:0:0:0:0:1:FF
  - Example:
    - The solicited node multicast address corresponding to the IPv6 address 3202:1206:1977:ABCD:7A53:78:40**0E:7C8C**
    - is FF02::1:FF**0E:7C8C**.
  - A node is required to compute and support a Solicited-Node multicast addresses for every unicast and anycast address it is assigned

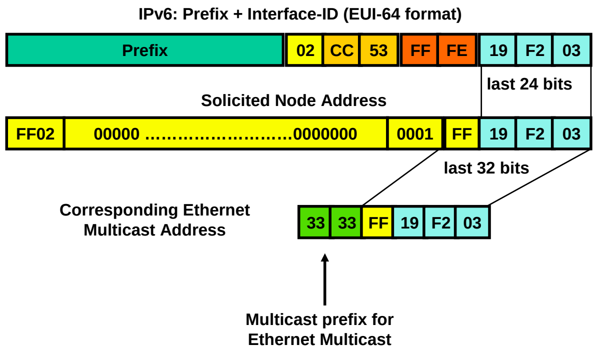
Solicited-Node Address



This is a special multicast address which is derived from the MAC address -> there will only an overlap if two machines have the same lower 3 bytes of the MAC address

Remember: Broadcasts are not possible, in order to make such things like ARP you specify this address instead of a broadcast

Example: Mapping IPv6 Solicited Node Address (MAC based) to an Ethernet Multicast Address



Host Required Addresses

- **A host is required to recognize the following addresses as identifying itself:**
  - Its link-local address for each interface
  - Any additional assigned unicast addresses
  - Loopback address
  - All-nodes multicast address
  - Solicited-node multicast address for each of its assigned unicast and anycast addresses
  - Multicast addresses of all other groups which the host belongs

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Router Required Addresses

- **A router is required to recognize the following additional addresses as identifying itself:**
  - The subnet-router anycast addresses for the links it has interfaces
  - All other anycast addresses with which the router has been configured
  - All-router multicast address

IP Addressing Architecture (RFC 3513)

- **Addressing structure of RFC 2373 changed**
  - To simplify and clarify how different address types are identified
  - This was done to insure that implementations do not build in any knowledge about global unicast format prefixes
  - Changes include:
    - Removed Format Prefix (FP) terminology
    - Revised list of address types to only include exceptions to global unicast and a single entry that identifies everything else as global unicast
  - Only address types are described

IPv6 Address Types (RFC 3513)

Address Type	Binary Prefix	IPv6 Notation
Unspecified	0……………..0	::/128
Loopback	0……………..1	::1/128
Link local-unicast	1111 1110 10	FE80::/10
Site local-unicast	1111 1110 11	FEC0::/10
Multicast	1111 1111	FF00::/8
Global unicast	everything else	

Assignment of Addresses (RFC 3513)

Allocation	Binary Prefix	Fraction of Address Space
Unassigned (note 1)	0000 0000	1/256
Unassigned	0000 0001	1/256
Reserved for NSAP allocation	0000 001	1/128
Unassigned	0000 01	1/64
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Global unicast address **(IANA)**	001	1/8
Unassigned	010	1/8
Unassigned	011	1/8
Unassigned	100	1/8

Note 1: The "unspecified address", the "loopback address", and the IPv6 Addresses with Embedded IPv4 Addresses are assigned out of the 0000 0000 binary prefix space.

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Assignment of Addresses (RFC 3513)

Allocation	Binary Prefix	Fraction of Address Space
Unassigned	101	1/8
Unassigned	110	1/8
Unassigned	1110	1/16
Unassigned	1111 0	1/32
Unassigned	1111 10	1/64
Unassigned	1111 110	1/128
Unassigned	1111 1110 0	1/512
Link local-use addresses	1111 1110 10	1/1024
Site local-use addresses	1111 1110 11	1/1024
Multicast addresses	1111	1/256

Note 2: For now, IANA should limit its allocation of IPv6 unicast address space to the range of addresses that start with binary value 001. The rest of the global unicast address space (approximately 85% of the IPv6 address space) is reserved for future definition and use and is not to be assigned by IANA at this time.

Final IP Addressing Architecture (RFC 4291)

- **Changes to RFC 3513**
  - Deprecated the Site-Local unicast prefix
    - Removed Site-Local from special list of prefixes in Section
    - Split section titled "Local-use IPv6 Unicast Addresses" into two sections, "Link-Local IPv6 Unicast Addresses" and "Site-Local IPv6 Unicast Addresses"
    - Added text to new section describing Site-Local deprecation
    - see RFC 3879 for reasons of deprecation
  - Deprecated the "IPv6 Compatible Address"
    - Because it is not being used in the IPv6 transition mechanisms
  - The restrictions on using IPv6 anycast addresses were removed
    - Because there is now sufficient experience with the use of anycast addresses, the issues are not specific to IPv6, the GROW working group is working in this area
  - Clarified that the "x" in the textual representation
    - Can be one to four digits.
  - Added the "R" and "P" flags
    - On multicast addresses and points to the documents that define them

Deprecation of Site Local Unicast (RFC 4291) Introduction of Unique Local Unicast (RFC 4193)

10-bits	54-bits	64-bits
F E C 0	Subnet ID	Interface ID

Site-local unicast

see RFC 3879 for reasons of deprecation

7-bits	1-bit	40-bits	16-bits	64-bits
Prefix	L	Global ID	Subnet ID	Interface ID

Unique-local unicast

see RFC 4193 Unique Local IPv6 Unicast Addresses
Prefix = FC00::/7
L = 1 address locally assigned (prefix FD00::/8)
L= 0 reserved for future usage (prefix FC00::/8)

Unique Local IPv6 Unicast Address (RFC 4193)

- **New concept for private addresses**
  - see RFC 4193 for details
- **Instead of using**
  - the same range private IP address numbering as in RFC 1918
- **The Global ID**
  - is calculated by a pseudo random algorithm for every organization with private addressing needs or wishes
- **Hence connecting**
  - two different privately addressed organizations in an private agreement is no longer more a problem
- **Of course**
  - Unique local addresses are not routed in the Internet
- **Nowadays turned on by default**
  - Windows and Linux systems
  - Be careful if you troubleshoot in such situation !!!

Deprecation IPv4-compatible IPv6 Addresses (RFC 4291)

80-bits	16-bits	32-bits
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	IPv4 Address

IPv4-compatible IPv6 address (x:x:x:x:x:x:d.d.d.d)

The "IPv4-Compatible IPv6 address" is now deprecated because the current IPv6 transition mechanisms no longer use these addresses
New or updated implementations are not required to support this address type

80-bits	16-bits	32-bits
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	F F F F	IPv4 Address

IPv4-mapped IPv6 address (x:x:x:x:x:ffff:d.d.d.d)

represents address of IPv4-only hosts,
used by hosts that do translation between IPv4 and IPv6 (e.g.: ::FFFF:193.170.150.1/80)
(translation is another transition technique for IPv4 ⇨ IPv6)
see **RFC 4038** for background on usage

Multicast Address (RFC 4291)

8-bits	4-bits	4-bits	112-bits
1 1 1 1 1 1 1 1	Flags	Scope	Group ID
	0 R P T		

Multicast Address

The P flag's definition and usage:
RFC 3306 Unicast-Prefix-based IPv6 Multicast Addresses (Updated by RFC3956, RFC4489) (Status: PROPOSED STANDARD)

The R flag's definition and usage:
RFC 3956 Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address (Updates RFC3306) (Status: PROPOSED STANDARD)

Multicast Group Management

- **Multicast membership handling for transient groups in IPv4**
  - Done by IGMP
    - Version 1 RFC 1112
    - Version 2 RFC 2236
    - Version 3 RFC 3376
- **Multicast membership handling for transient groups in IPv6**
  - Done by ICMPv6 extensions which includes tasks of IGMP
    - RFC 2710 Multicast Listener Discovery (MLD)
    - RFC 3810 Multicast Listener Discovery Version 2 (MLDv2)
  - Procedures for a station to join a multicast group is very similar to IPv4

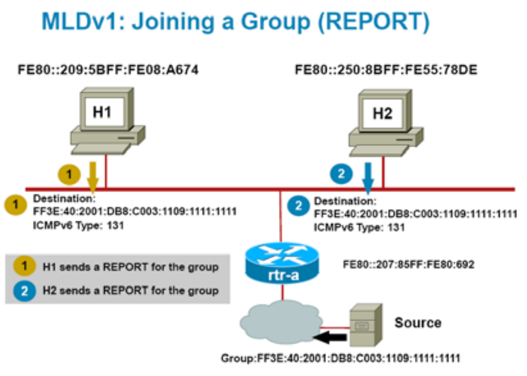
Multicast Listener Discovery (MLD)

- **RFC 2710 specifies MLDv1**
  - ICMPv6 type field defined
  - Multicast Listener Query (Type = decimal 130)
  - Multicast Listener Report (Type = decimal 131)
  - Multicast Listener Done (Type = decimal 132)
- **RFC 3810 specifies MLDv2 (== IGMPv3)**
  - Adds the ability for a node to report interest in listening to packets with a particular multicast address only **from specific source addresses** or from all sources except for specific source addresses
  - New Multicast Listener Report v2 (Type = decimal 143)
- **BTW**
  - IGMP snooping on Ethernet switches had to be enhanced by MLD snooping for IPv6 support

Datenkommunikation 384.081 - SS 2012

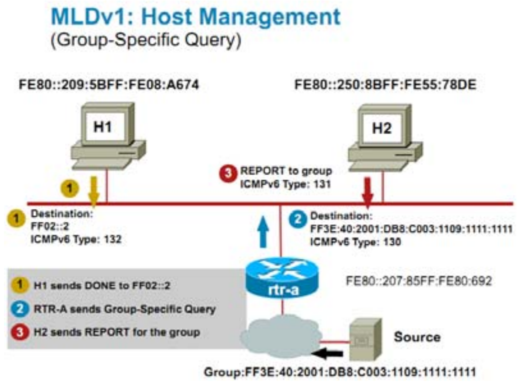
Appendix 5 - IPv6 (v4.7)

MLDv1 Example 1



Source: CiscoLive2011/2012 Presentation

MLDv1 Example 2



Source: CiscoLive2011/2012 Presentation

MLDv1 Example 3

Other MLD Operations

- Leave/DONE
  - Last host leaves—sends DONE (Type 132)
  - Router will respond with group-specific query (Type 130)
  - Router will use the last member query response interval (Default=1 sec) for each query
  - Query is sent twice and if no reports occur then entry is removed (2 seconds)
- General Query (Type 130)
  - Sent to learn of listeners on the attached link
  - Sets the multicast address field to zero
  - Sent every 125 seconds (configurable)

Source: CiscoLive2011/2012 Presentation

Agenda

- **History**
- **IPv6**
- **ICMPv6 and Plug&Play**
  - **Introduction**
  - ICMPv6
  - Neighbor Discovery
  - SLAAC
  - DHCPv6
  - Path MTU Discovery
- **Routing**
- **Transition**

Base Elements for Plug and Play

- **In order to understand the different forms of auto-configuration**
  - A detailed knowledge about, "Neighbor Discovery" and "IPv6 stateless Address Auto-configuration SLACC" and ICMPv6 is necessary
- **Neighbor Discovery (ND)**
  - Defined in RFC 4861 (Obsoletes RFC2461) (Updated by RFC5942) (Status: DRAFT STANDARD)
  - Router and prefix discovery, address resolution and redirect
  - The following ICMP message types are used
    - Router Solicitation (ICMPv6 type 133)
    - Router Advertisement (ICMPv6 type 134)
    - Neighbor Solicitation (ICMPv6 type 135)
    - Neighbor Advertisements (ICMPv6 type 136)
    - Redirect (ICMPv6 type 137)
- **IPv6 Stateless Address Auto-configuration SLAAC**
  - Defined in RFC 4862 (Obsoletes RFC2462) (Status: DRAFT STANDARD)
  - Stateless (e.g. no DHCP server necessary)
  - Node can obtain an IP address even in a server-less or router-less environment

Agenda

- **History**
- **IPv6**
- **ICMPv6 and Plug&Play**
  - Introduction
  - **ICMPv6**
  - Neighbor Discovery
  - SLAAC
  - DHCPv6
  - Path MTU Discovery
- **Routing**
- **Transition**

ICMPv6 1

- **Internet Control Message Protocol for IPv6 (ICMPv6)**
  - Defined in RFC 4443 Obsoletes RFC2463) (Updates RFC2780) (Updated by RFC4884) (Status: DRAFT STANDARD)
  - Is used by IPv6 nodes
    - To report errors encountered in processing packets,
    - To perform other internet-layer functions, such as diagnostics ("Ping")
  - Is an integral part of IPv6
    - The base protocol MUST be fully implemented by every IPv6 node
  - Have an look to the rate limiting (RFC 4443 chapter 2.4) and security considerations (RFC 4443 chapter 5)

ICMPv6 2

- **Base protocol elements:**
  - Next Header value of **58** in the immediately preceding header
  - Error messages
    - Destination unreachable (ICMPv6 type 1)
    - Packet too big (ICMPv6 type 2)
      - Pointing to the actual MTU to be used towards next hop
    - Time exceeded (ICMPv6 type 3)
      - Hop count or fragment reassembly time exceeded
    - Parameter problem (ICMPv6 type 4)
      - Erroneous header field encountered
      - Unrecognized Next Header type encountered
      - Unrecognized IPv6 option encountered
      - Pointer to the problem encountered
    - Private experimentation (ICMPv6 type 100, 101)
    - Reserved (ICMPv6 type 127)
      - For expansion of ICMPv6 error messages

ICMPv6 3

- **Base protocol elements (cont.):**
  - Informational messages (e.g. PING)
    - Echo request (ICMPv6 type 128)
    - Echo reply (ICMPv6 type 129)
      - Every node MUST implement an ICMPv6 Echo responder function that receives Echo Requests and originates corresponding Echo Replies
      - A node SHOULD also implement an application-layer interface for originating Echo Requests and receiving Echo Replies, for diagnostic purposes
    - Private experimentation (ICMPv6 type 200, 201)
    - Reserved (ICMPv6 type 255)
      - For expansion of ICMPv6 informational messages
- **Implements a similar behavior to the IPv6 world**
  - As ICMPv4 does it for the IPv4 world
  - Difference to ICMPv4 (RFC 792, 950)
    - No "Source quench" (flow control) anymore
    - No "Timestamp Request/Reply"
    - No "Information Request/Reply"
    - No "Address Mask Request/Reply"

Agenda

- **History**
- **IPv6**
- **ICMPv6 and Plug&Play**
  - Introduction
  - ICMPv6
  - **Neighbor Discovery**
  - SLAAC
  - DHCPv6
  - Path MTU Discovery
- **Routing**
- **Transition**

RFC 4861 Procedures

- **Includes the following procedures**
  - Router Discovery:
    - How hosts locate routers that reside on an attached link
    - No default-gateway entry in the host
  - Prefix Discovery:
    - How hosts discover the set of address prefixes that define which destinations are on-link for an attached link (nodes use prefixes to distinguish destinations that reside on-link from those only reachable through a router)
  - Parameter Discovery:
    - How a node learns link parameters such as the link MTU or Internet parameters such as the hop limit value to place in outgoing packets

RFC 4861 Procedures (cont.)

- Address Resolution = Neighbor Solicitations:
  - How nodes determine the link-layer address of an on-link destination (e.g., a neighbor) given only the destination's IP address.
- Redirect:
  - How a router informs a host of a better first-hop node to reach a particular destination
- Neighbor Unreachability Detection:
  - How nodes determine that a neighbor is no longer reachable
    - For neighbors used as routers, alternate default routers can be tried
    - For both routers and hosts, address resolution can be performed again
- Next-hop Determination:
  - The algorithm for mapping an IP destination address into the IP address of the neighbor to which traffic for the destination should be sent
  - The next-hop can be a router or the destination itself

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

RFC 4861 Procedures (cont.)

- Duplicate Address Detection:
  - How a node determines that an address it wishes to use is not already in use by another node
  - Defined in RFC 4862
- Address Auto-configuration (defined in RFC 4862): :
  - How nodes automatically configure an address for an interface
  - Mentioned in 4861 but defined in RFC 4862:

- **All protocol procedures uses**
  - the following well-known multicast types or address types
    - all-nodes multicast address
    - all-routers multicast address
    - solicited-node multicast address
    - link-local address
    - unspecified address

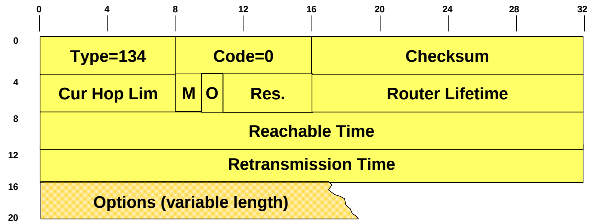
RFC 4861 ICMPv6 Messages

- Router Solicitation (RS), (ICMPv6 type 133):
  - When an interface becomes enabled, hosts may send out **Router Solicitations** that request routers to generate **Router Advertisements** immediately rather than at their next scheduled time
- Router Advertisement (RA), (ICMPv6 type 134):
  - Routers advertise their presence together with various link and Internet parameters either periodically, or in response to a Router Solicitation message
  - Contains prefixes that are used for on-link determination and/or address configuration, a suggested hop limit value, etc

RFC 4861 ICMPv6 Messages (cont.)

- Neighbor Solicitation (NS), (ICMPv6 type 135):
  - Sent by a node to determine the link-layer address of a neighbor, or to verify that a neighbor is still reachable via a cached link-layer address
  - Neighbor Solicitations are also used for Duplicate Address Detection
- Neighbor Advertisement (NA), (ICMPv6 type 136):
  - A response to a Neighbor Solicitation message
  - A node may also send unsolicited Neighbor Advertisements to announce a link-layer address change
- Redirect (ICMPv6 type 137):
  - Used by routers to inform hosts of a better first hop for a destination

Router Advertisement (RA) Header (RFC 4861)



Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Router Advertisement Fields (RFC 4861) 1

- **Current Hop Limit (8 bit)**
  - The default value that should be placed in the Hop Count field of the IP header for outgoing IP packets. A value of zero means unspecified (by this router)
- **Router Lifetime (16 bit)**
  - The lifetime associated with the default router in units of seconds. This field can contain values up to 65535 and receivers should handle any value, while the sending rules in limit the lifetime to 9000 seconds. A Lifetime of 0 indicates that the router is not a default router and SHOULD NOT appear on the default router list
  - Applies only to the router's usefulness as a default router; it does not apply to information contained in other message fields or options. Options that need time limits for their information include their own lifetime fields

Router Advertisement Fields (RFC 4861) 2

- **M - bit**
  - "Managed address configuration" flag. When set, it indicates that addresses are available via DHCPv6. If the M flag is set, the O flag is redundant and can be ignored because DHCPv6 will return all available configuration information
- **O - bit**
  - "Other configuration" flag. When set, it indicates that other configuration information is available via DHCPv6. Examples of such information are DNS-related information or information on other servers within the network
  - Note: If neither M nor O flags are set, this indicates that no information is available via DHCPv6

Router Advertisement Fields (RFC 4861) 3

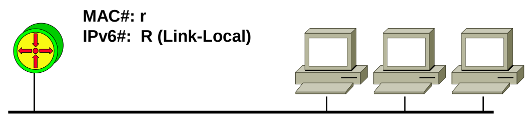
- **Reachable Time (32 bit)**
  - The time, in milliseconds, that a node assumes a neighbor is reachable after having received a reachability confirmation. Used by the Neighbor Unreachability Detection algorithm. A value of zero means unspecified (by this router)
- **Retransmission Timer (32 bit)**
  - The time, in milliseconds, between retransmitted Neighbor Solicitation messages. Used by address resolution and the Neighbor Unreachability Detection algorithm. A value of zero means unspecified (by this router)

Router Advertisement Fields (RFC 4861) 4

- **Possible Options:**
  - Source link-layer address
    - The link-layer address of the interface from which the Router Advertisement is sent. Only used on link layers that have addresses. A router MAY omit this option in order to enable inbound load sharing across multiple link-layer addresses
  - MTU
    - SHOULD be sent on links that have a variable MTU. MAY be sent on other links.
  - Prefix Information
    - These options specify the prefixes that are on-link and/or are used for stateless address autoconfiguration. A router SHOULD include all its on-link prefixes (except the link-local prefix) so that multihomed hosts have complete prefix information about on-link destinations for the links to which they attach..

© 2012, D.I. Lindner

Neighbor Discovery Router Discovery



Router periodically announces IPv6 prefixes (site-local, global) by sending out the following <u>Router Advertisement RA</u> (ICMP type 134) message in <u>multicast style</u>:

L2 Src = r
<u>L2 Dest = emc-All-Nodes</u>
ICMP type = 134
IP Src = R
<u>IP Dst = All-Nodes-MC (FF02::1)</u>
ICMP Data = prefixes, lifetime, other configuration parameters (MTU, Hop Limit, control bits for auto-configuration, ….)

Hosts use this message to fill the <u>Default Router List</u> and the <u>Prefix List</u>

Neighbor Discovery Router Solicitation 1

Station B requests a RA by sending out the following <u>Router Solicitation RS</u> (ICMP type 133) message:

L2 Src = b
<u>L2 Dest = emc-All-Routers</u>
ICMP type = 133
IP Src = B
<u>IP Dst = All-Routers-MC(FF02::2)</u>
ICMP Data = link-layer address of B = b

© 2012, D.I. Lindner

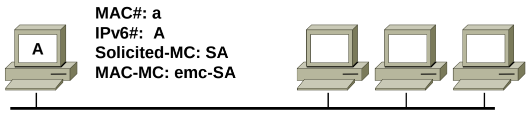
Neighbor Discovery Router Solicitation 2

Router R answers the request announces by sending out the following <u>Router Advertisement RA</u> (ICMP type 134) message in <u>unicast style</u>:

L2 Src = r
<u>L2 Dest = b</u>
ICMP type = 134
IP Src = R
<u>IP Dst = B</u>
ICMP Data = prefixes, lifetime, other configuration parameters (MTU, Hop Limit, Methods for auto-configuration, ….)

Hosts use this message to fill the <u>Default Router List</u> and the <u>Prefix List</u>

Neighbor Discovery Duplicate Address Detection (DAD)



Station A checks if a its IP address A is used by another system on the link by sending out the following <u>Neighbor Solicitation NS</u> (ICMP type 135) message in <u>multicast style</u>:

L2 Src = a
<u>L2 Dest = emc-SA</u>
ICMP type = 135
IP Src = 0 (::)
<u>IP Dst = SA</u>
ICMP Data = Target A, source link-layer address of A = a (option1)
(Query = What is your link-layer address?)

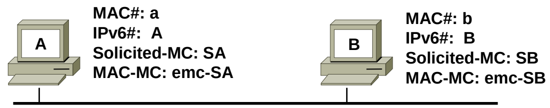
If there is no answer then IP address A can be used

© 2012, D.I. Lindner

Duplicate Address Detection (DAD)

- **Uses Neighbor Solicitation (NS) to check if another node on the link has the same IPv6 address**
- **DAD is used during the auto-configuration process**
  - It sends an NS packet to the solicited-node multicast address of ist own IPv6 address.
  - The source address of this packet is the „unspecified address"::
  - If a node responds to that request, it means that the IPv6 address is used and the requesting node should not use that address

Neighbor Discovery Address Resolution 1

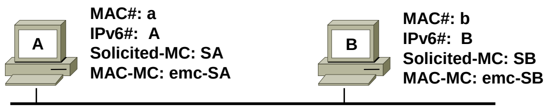


Station A tries to resolve IP address B (= get the MAC address of B) by sending out the following <u>Neighbor Solicitation NS</u> (ICMP type 135) message in <u>multicast style</u>:

L2 Src = a
<u>L2 Dest = emc-SB</u>
ICMP type = 135
IP Src = A
<u>IP Dst = SB</u>
ICMP Data = Target B, source link-layer address of A = a (option1)
(Query = What is your link-layer address?)

© 2012, D.I. Lindner

Neighbor Discovery Address Resolution 2

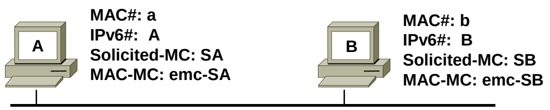


Station B resolves his IP address B by sending out the following Neighbor Advertisement NA (ICMP type 136) message in unicast style:

L2 Src = b
L2 Dest = a
ICMP type = 136
IP Src = B
IP Dst = A
ICMP Data = Target B, target link-layer address of B = b (option2)

A remembers answer in Neighbor Cache (mapping of IP -> MAC address)

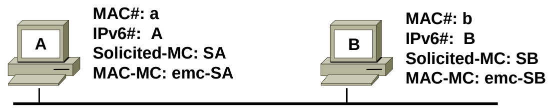
Neighbor Discovery Neighbor Reachability Test 1



Station A tries to check if IP address B is still reachable by sending out the following Neighbor Solicitation NS (ICMP type 135) message in unicast style:

L2 Src = a
L2 Dest = b
ICMP type = 135
IP Src = A
IP Dst = B
ICMP Data = Target B, source link-layer address of A = a (option1)
(Query = What is your link-layer address?)

Neighbor Discovery Neighbor Reachability Test 2



B resolves his IP address B by sending out the following Neighbor Advertisement NA (ICMP type 136) message in unicast style:

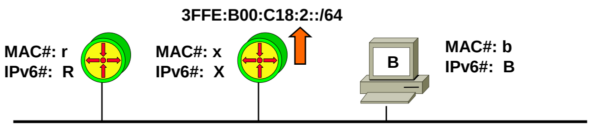
L2 Src = b
L2 Dest = a
ICMP type = 136
IP Src = B
IP Dst = A
ICMP Data = Target B, target link-layer address of B = b (option2)

A use answer to refresh his Neighbor Cache (mapping of IP -> MAC address)

Neighbor Unreachability Detection

- **Two possible scenarios for unreachable neighbors:**
  - If the end nodes are concerned
    - No recovery is possible
  - If the path between 2 nodes is concerned and an alternative path exists
    - Communication could be continued without upper layers detecting any change but what if the “Neighbor Cache” points into a “black hole” for the lifetime of an entry?
- **Therefore**
  - If an entry of the “Neighbor Cache” is not refreshed within 30 sec by normal activity it changes to a „Probe state“
  - 3 probe packets are sent and if there is no reply the entry gets deleted

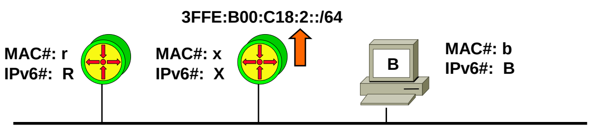
Neighbor Discovery Router Redirect 1



Router X has shortest way to destination 3FFE:B00:C18:2::1 but router R is default router of B; B sends a message to 3FFE:B00:C18:2::1; router R forwards the message to router X

L2 Src = b
L2 Dest = r
IP Src = B
IP Dst = 3FFE:B00:C18:2::1
Data = data for destination

Neighbor Discovery Router Redirect 2



Router R informs host B to take router X as next hop in order to reach 3FFE:B00:C18:2::1 by sending out the following <u>Router Redirect</u> (ICMP type 137) message in <u>unicast style</u>:

L2 Src = r
<u>L2 Dest = b</u>
ICMP type = 137
IP Src = R
<u>IP Dst = B</u>
ICMP Data = Target X, destination address 3FFE:B00:C18:2::1
(use next hop X to reach 3FFE:B00:C18:2::1)
target link-layer address of X = x (option2)

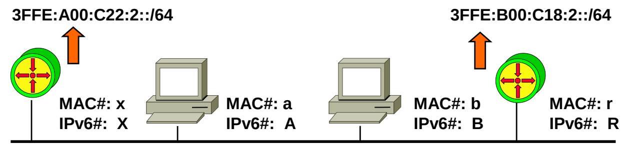
Host use this message to actualize the <u>Destination Cache</u>

© 2012, D.I. Lindner

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Example for Caches in System A



Prefix (Net-ID): FEC0::25:/64 (site-local) and 3FFE:B00:D24:2::25:/64 (global)

<u>Destination Cache</u>
3FFE:B00:C18:2::1
3FFE:A00:C22:2::1
B

<u>Neighbor Cache</u>
IPv6 R via MAC r
IPv6 X via MAC x
IPv6 B via MAC b

<u>Default Router List</u>
IPv6 R
IPv6 X

<u>Prefix List</u>
FEC0::25:/64
3FFE:B00:D24:2::25:/64

Conceptual Data Structures (RFC 4861) (1)

- **Neighbor Cache**
  - List of individual neighbors to which traffic has been sent recently
  - Entry is keyed on the neighbor's on-link unicast IP address
    - Contains such information as its link-layer address, a flag indicating whether the neighbor is a router or a host, a pointer to any queued packets waiting for address resolution to complete, etc.
    - Contains information used by the Neighbor Unreachability Detection algorithm, including the reachability state, the number of unanswered probes, and the time the next Neighbor Unreachability Detection event is scheduled to take place
  - Reachability State
    - Complete
    - Reachable
    - Stale
    - Delay
    - Probe

© 2012, D.I. Lindner

Conceptual Data Structures (RFC 4861) (2)

- **Neighbor Cache (cont.)**
  - Reachability state in detail:
    - INCOMPLETE
      - Address resolution is in progress and the link-layer address of the neighbor has not yet been determined
    - REACHABLE
      - The neighbor is known to have been reachable recently (within tens of seconds ago)
    - STALE
      - The neighbor is no longer known to be reachable but until traffic is sent to the neighbor, no attempt should be made to verify its reachability.
    - DELAY
      - The neighbor is no longer known to be reachable, and traffic has recently been sent to the neighbor
      - Rather than probe the neighbor immediately, however, delay sending probes for a short while in order to give upper-layer protocols a chance to provide reachability confirmation
    - PROBE
      - The neighbor is no longer known to be reachable, and unicast Neighbor Solicitation probes are being sent to verify reachability

Conceptual Data Structures (RFC 4861) (3)

- **Destination Cache**
  - List destinations to which traffic has been sent recently
  - Includes both on-link and off-link destinations and provides a level of indirection into the Neighbor Cache
  - Maps a destination IP address to the IP address of the next-hop neighbor
  - This cache is updated with information learned from Redirect messages
  - Implementations may find it convenient to store additional information not directly related to Neighbor Discovery in Destination Cache entries, such as the Path MTU (PMTU) and round-trip timers maintained by transport protocols

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Conceptual Data Structures (RFC 4861) (4)

- **Prefix List**
  - A list of the prefixes that define a set of addresses that are on-link
  - Created from information received in Router Advertisements.
  - Each entry has an associated invalidation timer value (extracted from the advetisement) used to expire prefixes when they become invalid. A special "infinity" timer value specifies that a prefix remains valid forever, unless a new (finite) value is received in a subsequent advertisement
  - The link-local prefix is considered to be on the prefix list with an infinite invalidation timer regardless of whether routers are advertising a prefix for it

Conceptual Data Structures (RFC 4861) (5)

- **Default Router List:**
  - A list of routers to which packets may be sent
  - Points to entries in the Neighbor Cache
  - The algorithm for selecting a default router favors routers known to be reachable over those whose reachability is suspect
  - Each entry also has an associated invalidation timer value (extracted from Router Advertisements) used to delete entries that are no longer advertised.

Conceptual Sending Algorithm (RFC 4861) (1)

- **Sending a packet to a destination**
  - Node uses a combination of the Destination Cache, the Prefix List, and the Default Router List to determine the IP address of the appropriate next hop ("next-hop determination")
  - Once the IP address of the next hop is known, the Neighbor Cache is consulted for link-layer information about that neighbor
- **Next-hop determination**
  - is not performed on every packet that is sent. Instead, the results of next-hop determination computations are saved in the Destination Cache (which also contains updates learned from Redirect messages). When the sending node has a packet to send, it first examines the Destination Cache. If no entry exists for the destination, next-hop determination is invoked to create a Destination Cache entry.

Conceptual Sending Algorithm (RFC 4861) (2)

- **Once the IP address of the next-hop node is known**
  - The sender examines the Neighbor Cache for link-layer information about that neighbor. If no entry exists, the sender creates one, sets its state to INCOMPLETE, initiates Address Resolution, and then queues the data packet pending completion of address resolution.
  - When Neighbor Advertisement response is received, the link-layer addresses is entered in the Neighbor Cache entry and the queued packet is transmitted
- **Each time a Neighbor Cache entry is accessed while transmitting a unicast packet**
  - the sender checks Neighbor Unreachability Detection related information according to the Neighbor Unreachability Detection algorithm. This unreachability check might result in the sender transmitting a unicast Neighbor Solicitation to verify that the neighbor is still reachable

Conceptual Sending Algorithm (RFC 4861) (3)

- **Next-hop determination is done**
  - The first time traffic is sent to a destination. As long as subsequent communication to that destination proceeds successfully, the Destination Cache entry continues to be used
  - If at some point communication ceases to proceed, a determined by the Neighbor Unreachability Detection algorithm, next-hop determination may need to be performed again.

Garbage Collection and Timeout Requirements (RFC 4861)

- **From the perspective of correctness there is no need to periodically purge Destination and Neighbor Cache entries.**
- **Although stale information can potentially remain in the cache indefinitely, the Neighbor Unreachability Detection algorithm ensures that stale information is purged quickly if it is actually being used.**
- **To limit the storage needed for the Destination and Neighbor Caches, a node may need to garbage-collect old entries. However, care must be taken to ensure that sufficient space is always present to hold the working set of active entries. A small cache may result in an excessive number of Neighbor Discovery messages if entries are discarded and rebuilt in quick succession. Any Least Recently Used (LRU)-based policy that only reclaims entries that have not been used in some time (e.g., ten minutes or more) should be adequate for garbage-collecting unused entries.**
- **A node should retain entries in the Default Router List and the Prefix List until their lifetimes expire. However, a node may arbage-collect entries prematurely if it is low on memory. If not all routers are kept on the Default Router list, a node should retain at least two entries in the Default Router List (and preferably more) in order to maintain robust connectivity for off-link destinations.**
- **When removing an entry from the Prefix List, there is no need to purge any entries from the Destination or Neighbor Caches. Neighbor Unreachability Detection will efficiently purge any entries in these caches that have become invalid. When removing an entry from the Default Router List, however, any entries in the Destination Cache that go through that router must perform next-hop determination againto select a new default router.**

ICMPv6 Enhancements (1)

- **Multihoming:**
  - There are a number of complicating issues that arise when Neighbor Discovery is used by hosts that have multiple interfaces
  - See RFC 4861 Appendix A
    - which identifies issues that require further study
  - See RFC 4191 "Default Router Preferences and More-Specific Routes"
    - New "Preference Value" field is introduced in Router Advertisement (RA)
- **Mobility support**
  - See RFC 6275 for new ICMPv6 types
    - Home Agent Address Discovery Request/Reply Msg. (ICMPv6 type 144/145)
    - Mobile Prefix Solicitation/Advertisment Message (ICMPv6 type 146/147)
    - Modified Router Advertisement (RA) Message (chapter 7.1)
      - The Home Agent (H) bit is set in a Router Advertisement to indicate that the router is also functioning as a Mobile IPv6 home agent on this link

ICMPv6 Enhancements (2)

- **Security:**
  - See security considerations in RFC 4861
    - Principle threats and securing ND by statically configured security associations (IPsec) are mentioned
  - See RFC 3756 "IPv6 Neighbor Discovery (ND) Trust Models and Threats"
    - The existing IETF standards specify that IPv6 Neighbor Discovery (ND) and Address Autoconfiguration mechanisms may be protected with IPsec Authentication Header (AH). However, the current specifications limit the security solutions to manual keying due to practical problems faced with automatic key management. This document specifies three different trust models and discusses the threats pertinent to IPv6 Neighbor Discovery
  - See RFC 3971 "SEcure Neighbor Discovery (SEND)"
    - IPv6 nodes use the Neighbor Discovery to discover other nodes on the link, to determine their link-layer addresses to find routers, and to maintain reachability information about the paths to active neighbors. If not secured, ND is vulnerable to various attacks. This document specifies security mechanisms for ND. Unlike those in the original ND specifications, these mechanisms do not use IPsec.
      - Cryptographically Generated Address (CGA)
      - RSA Signatures, Nonces, Timestamps
      - Trust Anchor, Certificates, Public-Keys

© 2012, D.I. Lindner

Agenda

- **History**
- **IPv6**
- **ICMPv6 and Plug&Play**
  - Introduction
  - ICMPv6
  - Neighbor Discovery
  - **SLAAC**
  - DHCPv6
  - Path MTU Discovery
- **Routing**
- **Transition**

Address Auto-Configuration

- **A feature**
  - That enables host to configure one or more addresses per interface automatically
- **Allows plug and play operation of a host**
  - One weaknesses of IPv4
- **Allows re-addressing of a host in case of**
  - Change of location or change of service provider
- **In IPv4**
  - BOOTP and DHCP servers could be used for address configuration
    - BOOTP depends on statically and manually database entries
    - DHCP can support dynamic reconfiguration (⇨ lifetime)

© 2012, D.I. Lindner

Address Auto-Configuration

- **Address configuration done by BOOTP and DHCP**
  - Depends on presence of a server
  - Stateful address auto-configuration
- **One challenge for IPv6 was**
  - To provide stateless address auto-configuration in addition do stateful configuration performed by DHCPv6
- **Stateless**
  - Host can obtain an address without any database pre-configuration of a server (no manual configuration)
  - Host can obtain an address even in a server-less and router-less environment

Stateless Auto Configuration

- **Every interface is pre-configured with a token**
  - Token must be unique per link
    - e.g. 48 bit MAC address
  - Token must be suitable for use as the Interface ID portion of an IPv6 address (EUI-64 addressing mechanism to get a 64 bit Interface ID)
- **In router-less/server-less topology**
  - The link-local address can be formed autonomously by the node
    - Using the token as Interface ID
    - Using the well known prefix of such an IPv6 address type
      - FE80:0:0:0:EUI-64 address or FE80::EUI-64 address
  - After a duplicate test (unspecified address as source address and link local address as destination address) the node can communicate with other nodes on the same link using this address

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Stateless Auto Configuration (cont.)

- **In topologies with routers**
  - A host can determine the current prefix associated with the link
    - Current prefix information is announced periodically by routers or may be solicited by the host from the routers on request
  - A host can use these prefixes together with the token to form a site-global / unique-local or Internet-global IPv6 address
  - Re-addressing can be done dynamically
    - By the help of two lifetimes associated with an announced prefix
    - Valid Lifetime
      - indicates how long the address formed from that prefix is valid
    - Preferred Lifetime
      - indicates when the address formed from the prefix will be deprecated

Stateless Auto Configuration (cont.)

- **Whenever the prefix is re-advertised**
  - The valid lifetime of the address is reset to the new value
- **When the prefix is not longer advertised**
  - The address will expire after the last advertised lifetime runs out
    - Can not be used any longer ⇨ harsh consequences on ongoing communications (e.g. TCP session will break)
- **The preferred lifetime can be used**
  - To indicate that an address (prefix) - although still valid - is about to become invalid
  - Hence a node should not use this address (prefix) as source address when initiating new communications
  - The node will choose another non-deprecated address for new communications instead

Stateless Autoconfiguration - RA

- **Router Advertisements (RA)**
  - Are sent periodically, and on request, by routers on all their configured interfaces
  - Is sent to the all-nodes multicast address
  - Message info:
    - One or more prefixes that can be used on the link
    - Lifetime of the prefixes
    - Default router information: existence and lifetime
    - Flags indicating the kind of autoconfiguration that can be done by hosts

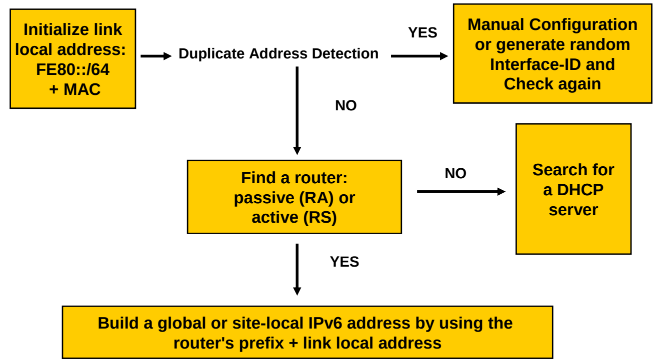
Stateless Autoconfiguration - RS

- **Router Solicitations (RS)**
  - Are sent by hosts at boot time
  - To ask routers to send an immediate RA on the local link
  - So hosts can receive the autoconfiguration information without waiting for the next scheduled RA
  - To avoid over flooding, RS should only be sent at boot time and only 3 times

Renumbering

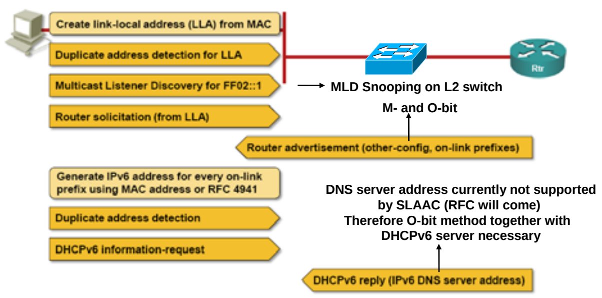
- **Renumbering**
  - Is achieved by sending RAs
  - They contain both the old and the new prefix
    - Old prefix with short lifetime
    - New prefix with nomal lifetime
  - The old prefix gets deprecated
    - That means nodes should use the new prefix for new connections while still keeping their current connections opened with the old prefix
  - During a certain time, node has two unicast addresses

IPv6 Address Configuration Overview



IPv6 Addressing Aspects and Procedures

IPv6 Host Configuration (Media-Independent)



Source: Webinar "Enterprise_IPv6_-_The_First_Steps" Ivan Pepelnjak 2011 257

IP Addressing Rules (Original)

- **ISP will typically get an /32 prefix from RIR**
  - Just by asking for IPv6
  - That means 4 Billions of prefixes (=subnets) are available for numbering leaving the 64-bits of the interface-ID untouched
- **ISP will use /64**
  - For every ISP internal network (VLAN, P2P link)
  - For every single small users needing just one subnet
- **ISP will give a /60 ... /56**
  - To residential customers allowing them to address 16 ... 255 subnets in their domain
- **ISP will give a /48**
  - To business customers allowing them to address 65536 subnets in their domain
- **SLAAC works only**
  - On subnets using /64 prefixes; other subnets should use the same length for consistency; some equipment might not work with longer prefixes than /64
- **Result**
  - Use /64 everywhere (including P2P and loopback links)

© 2012, D.I. Lindner IPv6, v4.7 258

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

IP Addressing Rules Issues

- **Interface-ID = 64 bits**
  - There are 2^{64} possible interface-IDs on a subnet
- **Problems with DOS attacks:**
  - An IPv6 ND exhaustion attack pointed to ASICs-based L3 switches using a random target interface-ID will block resources on that L3 switch
  - More effective than ARP exhaustion in IPv4 because of the larger address range for host
- **Problems with loops**
  - Ping-pong of packets on multi-access network used in a P2P style (e.g. Ethernet) with a destination address containing a random target interface-ID
    - Ping-pong will cease until hop-count of IPv6 is decremented to 0

© 2012, D.I. Lindner IPv6, v4.7 259

IP Addressing Rules (Modified)

- **Therefore**
  - On subnets where autoconfiguration (SLAAC) is needed for clients -> /64
  - On server subnets with static or DHCPv6-assigned addresses reduce it to /120
    - prefix taken out of /64 to have the choice to change it back if there are troubles)
  - On core P2P links -> /127
    - **RFC 6164** Using 127-Bit IPv6 Prefixes on Inter-Router Links (Status: PROPOSED STANDARD)
  - On customer P2P links where you need SLAAC still -> /64
  - On loopback -> /128
  - Should work but must be tested

© 2012, D.I. Lindner IPv6, v4.7 260

IP Addressing Rules Enterprise

- **Get a public /48 prefix from your ISP**
  - For your whole organization or each major site connected to the Internet
- **Change of provider**
  - Means renumbering
  - Easy for small sites
  - Nightmare for large sites
  - DNS issues even with autonumbering
- **Therefore**
  - Try to get a Provider-Independent (PI) address space from RIR
    - Arguments for RIR are multihoming intent, long term provider independence, largeness / importance of enterprise
  - Of course PI does not help against core routing table explosion
    - PI can not be aggregated by the ISPs toward the Internet core
    - But that is the problem of the ISPs and the Internet

Agenda

- **History**
- **IPv6**
- **ICMPv6 and Plug&Play**
  - Introduction
  - ICMPv6
  - Neighbor Discovery
  - SLAAC
  - **DHCPv6**
  - Path MTU Discovery
- **Routing**
- **Transition**

© 2012, D.I. Lindner

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Stateful Autoconfiguration

- **DHCPv6**
  - **RFC 3315** Dynamic Host Configuration Protocol for IPv6 (DHCPv6) (Updated by RFC4361, RFC5494, RFC6221) (Status: PROPOSED STANDARD)
  - Provides a device with addresses assigned by a DHCP server and other configuration information, which are carried in Stateful counterpart to SLAAC
  - The operational models and relevant configuration information for DHCPv4 and DHCPv6 are sufficiently different that integration between the two services is not included in the RFC 3315
    - Not any more on top of BootP
  - Can be used for automatic domain registration of hosts using dynamic DNS

DHCPv6 Details (RFC 3315) 1

- **Clients first detect the presence of routers on the link**
  - If found, then examines RAs to determine if DHCP can be used
- **If no router is found or if DHCP should be used there are two options:**
  - 4 messages exchange, if address configuration has to be delivered
  - 2 messages exchange, if only other configuration than addresses has to be delivered
- **Clients and servers exchange DHCP messages using UDP**
  - Clients listen on **UDP port 546**
  - Servers and relay agents on **UDP port 547**
- **To allow a DHCP client to send a message to a DHCP server that is not attached to the same link**
  - A DHCP relay agent on the client's link will relay messages between the client and server; the operation of the relay agent is transparent to the client
- **Addressing:**
  - The client uses a **link-local address** or addresses determined through other mechanisms for transmitting and receiving DHCP messages
  - DHCP servers receive messages from clients using a reserved, **link-scoped multicast address**
  - A DHCP client transmits most messages to this reserved multicast address, so that the client need not be configured with the address or addresses of DHCP servers
- **Once the client has determined the address of a server**
  - It may under some circumstances send messages directly to the server using unicast

© 2012, D.I. Lindner

DHCPv6 Details (RFC 3315) 2

- **4 Messages exchange:**
  - **DHCPv6 Solicit** message is sent to “All-DHCP-Agents and Relay Agents” multicast address **(FF02::1:2)** with link-local scope
    - Using the link-local address as the source address
  - If no local DHCP server is present on the link but a DHCP relay agent is implemented on a machine
    - DHCP relay agents forwards the request to the “All-DHCP-Server” multicast address **(FF05::1:3)** which is site scope
  - The DHCP server responds with a **DHCPv6 Advertise** message
    - This message contains one or more IPv6 unicast addresses of DHCP servers
  - By using **DHCPv6 Request** and **DHCPv6 Reply** messages the address can be delivered to the host
  - By using **DHCPv6 Release** or **DHCPv6 Reconfigure** messages the address can be returned or refreshed

DHCPv6 Details (RFC 3315) 3

- **2 Messages exchange:**
  - **DHCPv6 Information Request** message is sent to “All-DHCP-Agents and Relay Agents” multicast address **(FF02::1:2)** with link-local scope
    - Using the link-local address as the source address
  - If no local DHCP server is present on the link but a DHCP relay agent is implemented on a machine
    - DHCP relay agents forwards the request to the “All-DHCP-Server” multicast address **(FF05::1:3)** which is site scope
  - The DHCP server responds with a **DHCPv6 Reply** message

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

DHCPv6 Details (RFC 3315) 4

- **Message Types:**
  - SOLICIT (1)
    - A client sends a Solicit message to locate servers
  - ADVERTISE (2)
    - A server sends an Advertise message to indicate that it is available for DHCP service, in response to a Solicit message received from a client
  - REQUEST (3)
    - A client sends a Request message to request configuration parameters, including IP addresses, from a specific server
  - CONFIRM (4)
    - A client sends a Confirm message to any available server to determine whether the addresses it was assigned are still appropriate to the link to which the client is connected

DHCPv6 Details (RFC 3315) 5

- **Message Types (cont.):**
  - RENEW (5)
    - A client sends a Renew message to the server that originally provided the client's addresses and configuration parameters to extend the lifetimes on the addresses assigned to the client and to update other configuration parameters
  - REBIND (6)
    - A client sends a Rebind message to any available server to extend the lifetimes on the addresses assigned to the client and to update other configuration parameters; this message is sent after a client receives no response to a Renew message
  - REPLY (7)
    - A server sends a Reply message containing assigned addresses and configuration parameters in response to a Solicit, Request, Renew, Rebind message received from a client. A server sends a Reply message containing configuration parameters in response to an Information-request message. A server sends a Reply message in response to a Confirm message confirming or denying that the addresses assigned to the client are appropriate to the link to which the client is connected. A server sends a Reply message to acknowledge receipt of a Release or Decline message

DHCPv6 Details (RFC 3315) 6

- **Message Types (cont.):**
  - RELEASE (8)
    - A client sends a Release message to the server that assigned addresses to the client to indicate that the client will no longer use one or more of the assigned addresses
  - DECLINE (9)
    - A client sends a Decline message to a server to indicate that the client has determined that one or more addresses assigned by the server are already in use on the link to which the client is connected
  - RECONFIGURE (10)
    - A server sends a Reconfigure message to a client to inform the client that the server has new or updated configuration parameters, and that the client is to initiate a Renew/Reply or Information-request/Reply transaction with the server in order to receive the update information

DHCPv6 Details (RFC 3315) 6

- **Message Types (cont.):**
  - INFORMATION-REQUEST (11)
    - A client sends an Information-request message to a server to request configuration parameters without the assignment of any IP addresses to the client
  - RELAY-FORW (12)
    - A relay agent sends a Relay-forward message to relay messages to servers, either directly or through another relay agent. The received message, either a client message or a Relay-forward message from another relay agent, is encapsulated in an option in the Relay-forward message
  - RELAY-REPL (13)
    - A server sends a Relay-reply message to a relay agent containing a message that the relay agent delivers to a client. The Relay-reply message may be relayed by other relay agents for delivery to the destination relay agent. The server encapsulates the client message as an option in the Relay-reply message, which the relay agent extracts and relays to the client

Agenda

- **History**
- **IPv6**
- **ICMPv6 and Plug&Play**
  - Introduction
  - Neighbor Discovery
  - SLAAC
  - DHCPv6
  - Path MTU Discovery
- **Routing**
- **Transition**

Path MTU Discovery (RFC 1981) (1)

- **Basic ideas:**
  - Source node initially assumes that the PMTU of a path is the (known) MTU of the first hop in the path
  - If any of the packets sent on that path are too large to be forwarded by some node along the path, that node will discard them and return ICMPv6 Packet Too Big messages
  - Source node reduces its assumed PMTU for the path based on the MTU of the constricting hop as reported in the Packet Too Big message.
  - Several iterations of the packet-sent/Packet-Too-Big-message-received cycle possible

Path MTU Discovery (RFC 1981) (2)

- **Handling aspects:**
  - The PMTU of a path may change over time, due to changes in the routing topology
  - Reductions of the PMTU are detected by Packet Too Big messages.
  - To detect increases in a path's PMTU, a node periodically increases its assumed PMTU.
  - Attempts to detect increases in a path's PMTU should be done infrequently.
  - Path MTU Discovery supports multicast as well as unicast destinations
  - Path MTU Discovery must be performed even in cases where a node "thinks" a destination is attached to the same link as itself like a neighboring router acts as proxy for some destination

Path MTU Discovery (RFC 1981) (3)

- **Implementation aspects:**
  - IPv6 nodes SHOULD implement Path MTU Discovery in order to discover and take advantage of paths with PMTU greater than the IPv6 minimum link MTU [IPv6-SPEC]
  - A minimal IPv6 implementation may choose to omit implementation of Path MTU Discovery
  - Nodes not implementing Path MTU Discovery use the IPv6 minimum link MTU as the maximum packet size.

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Further Information IPv6 General

- **Internet Protocol Journal** **(www.cisco.com/ipj)**
  - Issue Volume 6, Number 2 (June 2003)
    - „The Myth of IPv6"
  - Issue Volume 6, Number 3 (September 2003)
    - „IPv6 Behind the Wall"
  - Issue Volume 7, Number 2 (June 2004)
    - „IPv6 Autoconfiguration"
  - Issue Volume 9, Number 3 (September 2006)
    - „IPv6 Internals"
  - Issue Volume 10, Number 2 (June 2007)
    - „IPv6 Network Mobility" Mobile IP
  - Issue Volume 13, Number 3 (September 2010)
    - „" Proxy Mobile IPv6 (PMIPv6)"

Agenda

- **History**
- **IPv6**
- **ICMPv6 and Plug&Play**
- **<u>Routing</u>**
- **Transition**
- **Miscellaneous**

IPv6 and Unicast Routing

- **Unicast routing in IPv6**
  - Is almost identical to IPv4 routing under CIDR except for the effect of 128-bit address size
    - Prefix routing
    - Longest match routing rule
      - If several matches in he routing table then the best match
  - Straightforward extensions to use all of IPv4´s routing algorithms
    - OSPF over IPv6 (RFC 5340, OSPFv3)
    - RIPng for IPv6 (RFC 2080)
      - Same bad functionality as RIPv2
      - (Count-to-Infinity, Split Horizon, Hold-Down, max hop 15)
    - BGP
      - RFC 4760 Multiprotocol Extensions for BGP-4 (Obsoletes RFC2858) (Status: DRAFT STANDARD)
      - RFC 2545 Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing (Status: PROPOSED STANDARD)

OSPFv3 Overview

- **OSPF for IPv6**
- **Based on OSPFv2, with enhancements**
- **Distributes IPv6 prefixes**
- **Runs directly over IPv6**
- **Ships-in-the-night with OSPFv2**

OSPFv3 / OSPFv2 Similarities

- **Basic packet types**
  - Hello, DBD, LSR, LSU, LSA
- **Mechanisms for neighbor discovery and adjacency formation**
- **Interface types**
  - P2P, P2MP, Broadcast, NBMA, Virtual
- **LSA flooding and aging**
- **Nearly identical LSA types**

OSPFv3 / OSPFv2 Differences (1)

- **OSPFv3 now runs on a per-link basis rather than on a per-IP-subnet basis**
  - IPv6 uses the term "link" to indicate "a communication facility or medium over which nodes can communicate at the link layer"
  - "Interfaces" connect to links. Multiple IPv6 subnets can be assigned to a single link, and two nodes can talk directly over a single link, even if they do not share a common IPv6 subnet (IPv6 prefix)
  - For this reason, OSPF for IPv6 runs per-link instead of the IPv4 behaviour of per-IP-subnet. Likewise, an OSPF interface now connects to a link instead of an IP subnet
  - This change affects the receiving of OSPF protocol packets, the contents of Hello packets, and the contents of network-LSAs

OSPFv3 / OSPFv2 Differences (2)

- **Addressing semantics have been removed from the OSPF protocol packets and the main LSA types, leaving a network-protocol-independent core**
  - IPv6 addresses are not present in OSPF packets, except in LSA payloads carried by the Link State Update packets
  - Router-LSAs and network-LSAs no longer contain network addresses
    - They simply express topology information based on Router-IDs and Link-IDs
  - OSPF Router IDs, Area IDs, and LSA Link State IDs remain at the IPv4 size of 32 bits. They can no longer be assigned as (IPv6) addresses
  - Neighbouring routers are now always identified by Router ID. Previously, they had been identified by an IPv4 address on broadcast, NBMA (Non-Broadcast Multi-Access), and point-to-multipoint links

OSPFv3 / OSPFv2 Differences (3)

- **Addition of Flooding Scope**
  - Flooding scope for LSAs has been generalized and is now explicitly coded in the LSA's LS type field
  - There are now three separate flooding scopes for LSAs
  - Link-local scope:
    - LSA is only flooded on the local link and no further. Used for the new **link-LSA**
  - Area scope:
    - LSA is only flooded throughout a single OSPF area. Used for router-LSAs, network-LSAs, inter-area-prefix-LSAs, inter-area-router-LSAs, and intra-area-prefix-LSAs
  - AS scope:
    - LSA is flooded throughout the routing domain. Used for AS-external-LSAs. A router that originates AS scoped LSAs is considered an AS Boundary Router (ASBR) and will set its E-bit in router-LSAs for regular areas

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

OSPFv3 / OSPFv2 Differences (4)

- **Use of Link-Local Addresses**
  - IPv6 link-local addresses are for use on a single link, for purposes of neighbour discovery, auto-configuration, etc
  - IIPv6 routers do not forward IPv6 datagrams having link-local source addresses
  - Link-local unicast addresses are assigned from the IPv6 address range FE80/10
  - OSPF for IPv6 assumes that each router has been assigned link-local unicast addresses on each of the router's attached physical links
  - On all OSPF interfaces OSPF packets are sent using the interface's associated link-local unicast address as the source address
  - A router learns the link-local addresses of all other routers attached to its links and uses these addresses as next-hop information during packet forwarding

OSPFv3 / OSPFv2 Differences (5)

- **Authentication has been removed from the OSPF protocol**
  - **The "AuType" and "Authentication" fields have been removed from the OSPF packet header, and all authentication-related fields have been removed from the OSPF area and interface data structures**
  - **When running over IPv6, OSPF relies on the IP Authentication Header and the IP Encapsulating Security Payload (see as described in RFC 4552) to ensure integrity and authentication/confidentiality of routing exchanges**
- **Explicit Support for Multiple Instances per Link**
  - **OSPF now supports the ability to run multiple OSPF protocol instances on a single link**
  - **Support for multiple protocol instances on a link is accomplished via an "Instance ID" contained in the OSPF packet header and OSPF interface data structures**

OSPFv3 / OSPFv2 Differences (6)

- **New LSAs**
  - Have been created to carry IPv6 addresses and prefixes
  - Link-LSAs:
    - Have link-local flooding scope; they are never flooded beyond the link with which they are associated
    - Hence will not be found in the database of non-adjacent routers
    - Provide the router's link-local address to all other routers attached to the link and inform other routers attached to the link of a list of IPv6 prefixes to associate with the link
  - Intra-area-prefix-LSA
    - This LSA carries all IPv6 prefix information that in OSPV2 is included in router-LSAs and network-LSAs
- **Renamed LSAs**
  - Inter-area-prefix-LSAs
    - Renaming of OSPFv2 Type-3 summary-LSAs
  - Inter-area-router-LSAs
    - Renaming of OSPFv2 Type-4 summary LSAs

LSA Type Review

	LSA Function Code	LSA type
Router-LSA	1	0x2001
Network-LSA	2	0x2002
Inter-Area-Prefix-LSA	3	0x2003
Inter-Area-Router-LSA	4	0x2004
AS-External-LSA	5	0x4005
Group-membership-LSA	6	0x2006
Type-7-LSA	7	0x2007
Link-LSA	8	0x0008
Intra-Area-Prefix-LSA	9	0x2009

Source: CiscoLive2011/2012 Presentation

OSPv3 Summary (1)

OSPF
High-level overview

High-level perspective	·OSPF is for the most part more "optimized" (and therefore significantly more complex) ·Only LSAs are extensible (not hellos, etc.). ·Unrecognized LSA types are not flooded (though opaque LSAs can suffice, if implemented universally). ·Uses complex, multistate process to synchronize databases between neighbors. Intended to minimize transient routing problems by ensuring that a newborn router has nearly complete routing information before it begins carrying traffic.
Encapsulation	OSPF runs on top of IP ·Traditional IP routing protocol approach ·Allows virtual links (if you like them) ·Relies on IP fragmentation for large LSAs ·Subject to spoofing and DoS attacks (use of authentication is strongly advised).

Source: CiscoLive2011/2012 Presentation

OSPv3 Summary (2)

OSPFv3
Comparative overview

Implementation	Similar Concepts as OSPFv2: - Runs directly over IPv6 (port 89) - Uses the same basic packet types - Neighbor discovery and adjacency formation mechanisms are identical (All OSPF Routers FF02::5, All OSPF DRs FF02::6) - LSA flooding and aging mechanisms are identical - Same interface types (P2P, P2MP, Broadcast, NBMA, Virtual) Independent process from OSPFv2
Important Differences	**OSPFv3 Is Running per Link Instead of per Node (and IP Subnet)** **Support of Multiple Instances per Link:** - New field (instance) in OSPF packet header allows running multiple instances per link - Instance ID should match before packet is being accepted - Useful for traffic separation, multiple areas per link **Generalization of Flooding Scope:** - Three flooding scopes for LSAs (link-local scope, area scope, AS scope) and they are coded in the LS type explicitly

Source: CiscoLive2011/2012 Presentation

OSPv3 Summary (3)

OSPFv3

Comparative overview

Important Differences (cont.)	**OSPF Packet Format has Been Changed:** - The mask field has been removed from Hello packet - IPv6 prefixes are only present in payload of Link State update packet **Two New LSAs Have Been Introduced:** - Link-LSA has a link local flooding scope and has three purposes Carry IPv6 link local address used for NH calculation Advertise IPv6 global address to other routers on the link (used for multi-access link) Convey router options to DR on the link - Intra-area-prefix-LSA to advertise router's IPv6 address within the area
Notes	**Standardization** Main standard: RFC 2740, RFC 5340 Evolution: draft-ietf-ospf-mt-ospfv3 draft-ietf-ospfv3-af-alt

Source: CiscoLive2011/2012 Presentation

IPv6 and Routing Extensions

- **IPv6 includes simple routing extensions**
  - Which support powerful new routing functionality
    - Provider selection
    - Host mobility
    - Auto re-addressing
  - Basis is a sequenced list of routers to be visited
    - Routing extension header
  - RHT 0 deprecated
  - RHT 2 for "Mobility Support in IPv6"

IPv6 and Multicast Routing

IPv4 and IPv6 Multicast Comparison

Service	IPv4 Solution	IPv6 Solution
Addressing Range	32-bit, Class D	128-bit (112-bit Group)
Routing	Protocol Independent, All IGPs and MBGP	Protocol Independent, All IGPs and MBGP with v6 mcast SAFI
Forwarding	~~PIM-DM,~~ PIM-SM, PIM-SSM, PIM-bidir, PIM-BSR	PIM-SM, PIM-SSM, PIM-bidir, PIM-BSR
Group Management	IGMPv1, v2, v3	MLDv1, v2
Domain Control	Boundary, Border	Scope Identifier
Interdomain Solutions	MSDP Across Independent PIM Domains	Single RP Within Globally Shared Domains

Source: CiscoLive2011/2012 Presentation

Agenda

- **History**
- **IPv6**
- **ICMPv6 and Plug&Play**
- **Routing**
- **Transition**
- **Miscellaneous**

Original Goal of Transition IPv4 to IPv6

- **Transition mechanism must ensure an easy evolution from IPv4 to IPv6**
  - Otherwise IPv6 will not be accepted
  - IP society has learned the lessons experienced by similar approaches in other protocol worlds
    - Decnet IV to Decnet OSI
    - AppleTalk Phase 1 to Phase 2
  - Statement from RFC 2893
    - “The key to a successful IPv6 transition is compatibility with the large installed base of IPv4 hosts and routers. Maintaining compatibility with IPv4 while deploying IPv6 will streamline the task of transitioning the Internet to IPv6”

Major Elements for Transition Mechanism (Original Idea) (1)

- **Dual-IP layers (dual stack) in hosts and routers**
  - Name servers and routers will provide support for both IPv4 and IPv6 during the transition period but hosts are gradually upgraded over a certain period of time
  - Upgraded hosts can communicate with both IPv4 and IPv6 nodes using their native protocol
  - Temporary IPv4 addresses are assigned when communicating with an IPv4-only host
  - Necessary steps:
    - Small DNS upgrade to support IPv6 addresses
    - Relatively small host upgrade to provide
      - IPv6, ICMPv6, Neighbor Discovery, handling of IPv6 within TCP and UDP, sockets or winsock libraries, interface with the name service
    - Slightly more complex router upgrade to provide
      - IPv6 forwarding, IPv6 routing, IPv6 management
      - should not be a problem for ships in the night multiprotocol router

© 2012, D.I. Lindner

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Major Elements for Transition Mechanisms (Original Idea) (2)

- **Tunneling IPv6 over IPv4**
  - Routers can tunnel traffic through IPv4 routing topologies by encapsulation IPv6 in IPv4
    - Lots of experience about tunneling gained in MBONE experiment
    - Configured tunneling
  - Hosts can tunnel traffic through IPv4 routing topologies by encapsulation IPv6 in IPv4
    - IPv4 network is used as virtual interface that enables an IPv6 host to reach other IPv6 hosts or routers through tunnels
      - Reaching the IPv6 Internet
      - Reaching isolated IPv6 hosts
    - IPv4-compatible IPv6 address format is used to configure a link-local address for that virtual interface
    - Automatic tunneling
  - Existing installed IPv4 routing system can be used but IPv6 operation is allowed to get started early

Transition Evolution (1)

- **RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers (Obsoleted by RFC 2893) (Status: PROPOSED STANDARD)**
- **RFC 2185 Routing Aspects of IPv6 Transition (Status: INFORMATIONAL)**
  - Dual-IP-layer route computation, manual configuration of point-to-point tunnel
- **RFC 2893 Transition Mechanisms for IPv6 Hosts and Routers) (Obsoletes RFC 1933) (Obsoleted by RFC 4213)**
  - **Dual IP layer** (also known as Dual Stack):
    - A technique for providing complete support for both Internet protocols -- IPv4 and IPv6 -- in hosts and routers
  - **IPv4-compatible IPv6 addresses**:
    - An IPv6 address format that employs embedded IPv4 addresses.
  - **IPv6 over IPv4 tunneling:**
    - The technique of encapsulating IPv6 packets within IPv4 so that they can be carried across IPv4 routing infrastructures
  - **Configured tunneling**:
    - Point-to-point tunnels made by encapsulating IPv6 packets within IPv4 headers to carry them over IPv4 routing infrastructures. The IPv4 tunnel endpoint address is determined by configuration information on the encapsulating node,
  - **Automatic tunneling**:
    - A mechanism for using IPv4-compatible addresses to automatically tunnel IPv6 packets over IPv4 networks. The IPv4 tunnel endpoint address is determined from the IPv4 address embedded in the IPv4-compatible destination address of the IPv6 packet

© 2012, D.I. Lindner

Transition Evolution (2)

- **RFC 4213 Basic Transition Mechanisms for IPv6 Hosts and Routers (Obsoletes RFC2893) (Status: PROPOSED STANDARD)**
  - Same as RFC 2893 but removes automatic tunneling method and use of IPv4-compatible addresses
  - Automatic tunneling described in RFC 3056
- **RFC 3056 Connection of IPv6 Domains via IPv4 Clouds (Status: PROPOSED STANDARD)**
  - Optional interim mechanism for IPv6 sites first to communicate with each other over the IPv4 network without explicit tunnel setup, and second for communication to the IPv6 Internet via relay routers
  - Effectively treats the wide area IPv4 network as a unicast point-to-point link layer
  - The document defines a method for assigning an interim unique IPv6 address prefix to any site that currently has at least one globally unique IPv4 address, and specifies an encapsulation mechanism for transmitting IPv6 packets using such a prefix over the global IPv4 network
  - The mechanism is intended as a start-up transition tool used during the period of co-existence of IPv4 and IPv6
  - It is not intended as a permanent solution
  - **6to4** and **6to4 relay**

Transition Evolution (3)

- **RFC 2529 Transmission of IPv6 over IPv4 Domains without Explicit Tunnels (Status: PROPOSED STANDARD)**
  - **6over4**
- **RFC 2765 Stateless IP/ICMP Translation Algorithm (Obsoleted by RFC6145)**
  - **SIIT**
- **RFC 2767 Dual Stack Hosts using the "Bump-In-the-Stack" Technique (Status: INFORMATIONAL)**
  - **BIS**
- **RFC 2766 Network Address Translation - Protocol Translation (Obsoleted by RFC4966) (Status HISTORIC)**
  - **NAT-PT**
- **RFC 4038 "Application Aspects of IPv6 Transition" (Status: Informational)**

© 2012, D.I. Lindner

Transition Approaches

- **Dual-Stack Mechanisms**
  - Dual-Stack
  - Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  - IPv4-Compatible Tunnel
  - 6to4
  - Tunnel Broker
  - 6over4
  - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  - Teredo
- **Translation**
  - Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  - Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  - NAT444, NAT64/DNS64, 6RD, DS-Lite
  - 6PE, 6VPE
  - SHIM6 and LISP

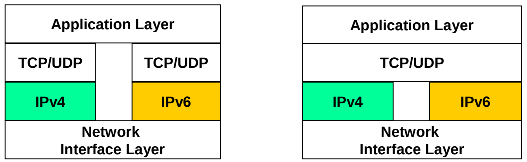
Dual-Stack / Dual-IP Layer

- First dual-stacks are implemented in network devices
  - Allow handling of both IPv4 and IPv6 packet types
- **Then end systems are gradually upgraded over a certain period of time from IPv4 to IPv6**
  - Upgraded hosts can communicate with both IPv4 and IPv6 nodes using the corresponding native protocol. Applications have to be modified to IPv6.
  - Old applications will still take IPv4 and new or modified applications will take IPv6
- Finally dual-stacks are implemented in all end systems
  - Both IPv4- and IPv6-capable applications can operate on the same node
- **Dual stacks needs new TCP and UDP**
  - That never happens
- **Therefore from that point on Dual IP**
  - Dual stack terminology still used but Dual IP meant instead of it
- **Name servers and routers**
  - will provide support for both IPv4 and IPv6 during the transition period

© 2012, D.I. Lindner

Dual-Stack versus Dual-IP Layer

Dual-Stack **Dual-IP Layer**



Separate driver implementation of TCP/UDP for IPv4 and IPv6

Dual-Stack / Dual-IP Layer

- **But dual-stack mechanisms do not solve IPv4 and IPv6 interoperation problems**
  - That will be the case if not all network components or end systems can be migrated to IPv6
  - Therefore translation between IPv4 and IPv6 is also required for this
- **Address issue solved?**
  - NO
  - For classical dual-stack method every IPv6 node needs also an unique IPv4 address
    - This is not applicable for practical implementations

Transition Approaches

- **Dual-Stack Mechanisms**
  - Dual-Stack
  - Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  - IPv4-Compatible Tunnel
  - 6to4
  - Tunnel Broker
  - 6over4
  - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  - Teredo
- **Translation**
  - Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  - Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  - NAT444, NAT64/DNS64, 6RD, DS-Lite
  - 6PE, 6VPE
  - SHIM6 and LISP

Dual-Stack Dominant Transition Mechanism (DSTM)

- **If an organization starts with IPv6 from the scratch and uses IPv6 in a dominant way**
  - E.g. if there are only IPv6 routers in the network and most end systems are provided with dual-stack but using IPv6 mainly
  - Then we still need a mechanism to be backward compatible to IPv4 in order to communicate with IPv4 only hosts
  - But giving an IPv4 address to every IPv6 host in order to be able to communicate with an IPv4 only node
    - Does not solve the address space problem
  - That is where DSTM comes in
    - Described in „draft-bound-dstm-exp-04.txt
    - http://bgp.potaroo.net/ietf/all-ids/draft-bound-dstm-exp-04.txt

DSTM (cont.)

- Therefore temporary IPv4 addresses are assigned to an IPv6 host when communicating with an IPv4 only host (or vice versa)
- **DSTM client** in an dual-stack of the IPv6 domain node get this address from a **DSTM server**
- **DSTM client** uses a "**Dynamic Tunnel Interface"** (DTI) to
  - encapsulates the IPv4 packets over IPv6 infrastructure
  - IPv4-over-IPv6 tunnel
- Tunnel endpoint is a **DSTM TEP router**
  - Which connects the IPv6 domain to a conventional IPv4 domain
- An IPv4 address will only assigned when needed
  - Communication of IPv6 hosts to IPv4 only hosts
  - Native IPv4 applications on IPv6 hosts

Transition Approaches

- **Dual-Stack Mechanisms**
  - Dual-Stack
  - Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  - IPv4-Compatible Tunnel
  - 6to4
  - Tunnel Broker
  - 6over4
  - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  - Teredo
- **Translation**
  - Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  - Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  - NAT444, NAT64/DNS64, 6RD, DS-Lite
  - 6PE, 6VPE
  - SHIM6 and LISP

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

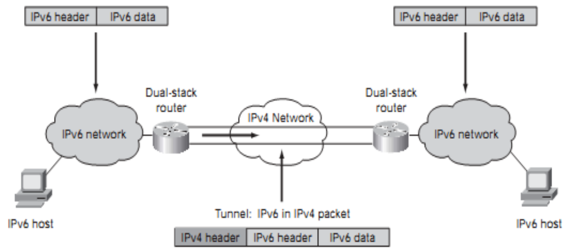
General

Tunneling

- Tunneling will be used in most cases during the migration process
- IPv4 routing infrastructure exists and IPv6 will use this infrastructure
- Dual stack hosts and routers can transmit IPv6 packets over an existing IPv4 topology
- IPv6 packet in an IPv4 tunnel

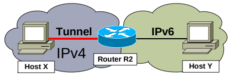
IPv4 Header	IPv6 Header	Payload

IPv6 over IPv4 Tunnels

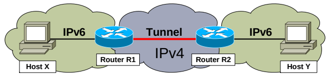


General (cont.)

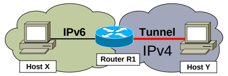
– IPv6 Host to Router via IPv4
(H to R)



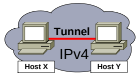
– Router to Router via IPv4
(R to R)



– Router to IPv6 Host via IPv4
(R to H)



– IPv6 Host to IPv6 Host via IPv4
(H to H)



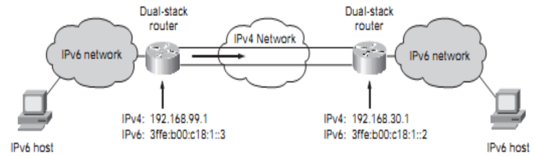
General (cont.)

- **Configured Tunnels**
  – R to R and H to R:
  – Tunnel endpoint is a router that must decode the packet and forward it to the final destination
  – No relationship between the router address and tunnel endpoint - the router address must be manually configured
  – Tunnel endpoint address is determined from configuration information in the encapsulating node
- **Automatic Tunnels**
  – H to H and R to H:
  – Tunnel endpoint address and the destination host address are the same
  – With IPv4 compatible IPv6 addresses the tunnel endpoint IPv4 address can automatically derived from the IPv6 address

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Manually Configured Tunnel



- **IPv4 network is seen as single IPv6 link**
- **Nat not allowed along path of tunnel**

How are Tunnels configured? 1

- **Manually**
  – Usually done at routers or hosts that tunnel traffic through IPv4 only topologies by encapsulation IPv6 in IPv4
    - There are lots of experience about such tunneling gained in MBONE experiment
  – Generic term: "**Configured Tunneling**"
  – In R to R or H to R scenarios:
    - Tunnel endpoint address
      – That is the IPv4 address to which an in IPv4 encapsulated IPv6 packet for a given IPv6 destination should be sent
    - is determined from configuration information in the encapsulating node (router or host)

How are Tunnels configured? 2

- **Automatic Tunneling**
  - In automatic tunneling, the tunnel endpoint address is determined from the packet being tunneled.
  - Usually done at IPv6 hosts to tunnel traffic through IPv4 only topologies in order to reach another IPv6 host or done at an IPv6 router to reach an isolated IPv6 host via IPv4 only topology
  - In H to H or R to H scenarios
    - If IPv4-compatible IPv6 addresses are assigned at the destination host then tunnel endpoint IPv4 address can automatically derived from the IPv6 address
    - Tunnel endpoint address and the destination host address are the same or could be derived from the destination host address
  - Examples
    - "IPv4-Compatible Tunneling" (RFC 2893) -> prefix 0:0:0:0:0:0/96
    - "6to4 Tunneling" (RFC 4213) -> prefix 2002::/16

How are Tunnels configured? 3

- **Semi-Automatic**
  - Dual stack clients connected to an IPv4 only topology want to get the right IPv4 address of a tunnel end-point on demand without manually configuring such addresses
  - A server function (called "Tunnel Broker") receives requests from dual-stack clients and tell them which IPv4 address should be used to reach the right tunnel endpoint for a certain IPv6 destination
  - In H to R scenarios:
    - Tunnel endpoint address
      - That is the IPv4 address to which an in IPv4 encapsulated IPv6 packet for a given IPv6 destination should be sent
    - is requested by the encapsulating node from the broker
  - Generic term: "**Tunnel Broker**"

© 2012, D.I. Lindner

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Transition Approaches

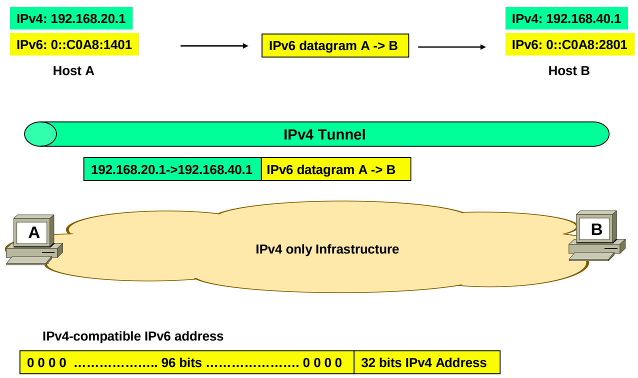
- **Dual-Stack Mechanisms**
  - Dual-Stack
  - Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  - IPv4-Compatible Tunnel
  - 6to4
  - Tunnel Broker
  - 6over4
  - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  - Teredo
- **Translation**
  - Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  - Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  - NAT444, NAT64/DNS64, 6RD, DS-Lite
  - 6PE, 6VPE
  - SHIM6 and LISP

Automatic IPv4-Compatible Tunnel

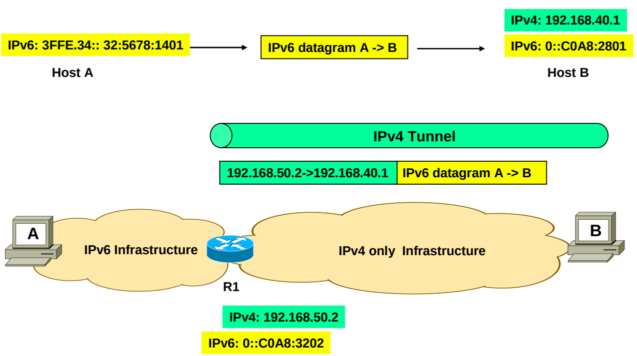
- **Dual Stack at end-system**
- **If destination address is an**
  - IPv4-compatible IPv6 address (e.g.: 0::0:192.168.1.4)
- **Then**
  - An automatic tunnels (IPv6 traffic in IPv4 encapsulated) can be setup
  - The destination IPv4 address can be derived from the IPv4-compatible IPv6 address
- **But this approach does not scale**
  - Because every IPv6 node must be configured with an IPv4 address
  - Address space limitations still the problem

© 2012, D.I. Lindner

Automatic IPv4-Compatible Tunnel (H to H)

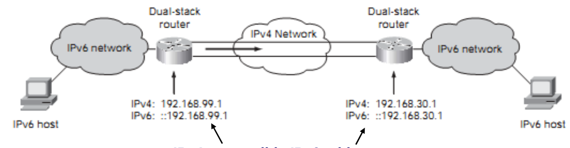


Automatic IPv4-Compatible Tunnel (R to H)



© 2012, D.I. Lindner

Automatic IPv4-Compatible Tunnel



IPv4-compatible IPv6 addresses

- **RFC 2893 (obsoleted by RFC 4213) describes IPv4-compatible tunnels;**
  - The IPv4-compatible tunnel is largely replaced by the 6to4 (RFC 3056, Connection of IPv6 Domains via IPv4 Clouds) automatic tunnel mechanism. Hence, the use of IPv4-compatible tunnel as a transition mechanism is nearly deprecated.
  - Note (RFC 4213 removes automatic tunnel mechanism)

Automatic IPv4-Compatible Tunnel

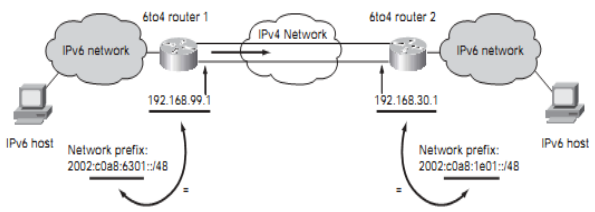
- Although an automatic tunnel can be configured between end systems, edge routers, or an edge router and an end system, the automatic IPv4-compatible tunnel has mainly been used to establish connection between routers.
- Unlike a manually configured tunnel, the automatic IPv4-compatible tunnel technique constructs tunnels with remote nodes on the fly.
- Manual configuration of the endpoints of the tunnel is not required because the tunnel source and the tunnel destination are automatically determined by the IPv4 address. The automatic tunnels are set up and taken down as required, and last only as long as the communication.
- Although an easy way to create tunnels, the IPv4-compatible tunnel mechanism does not scale well for IPv6 net-works deployment, because each host requires an IPv4 address removing the benefit of the large IPv6 addressing space.

© 2012, D.I. Lindner

Transition Approaches

- **Dual-Stack Mechanisms**
  - Dual-Stack
  - Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  - IPv4-Compatible Tunnel
  - 6to4
  - Tunnel Broker
  - 6over4
  - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  - Teredo
- **Translation**
  - Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  - Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  - NAT444, NAT64/DNS64, 6RD, DS-Lite
  - 6PE, 6VPE
  - SHIM6 and LISP

Automatic 6to4 Tunnel



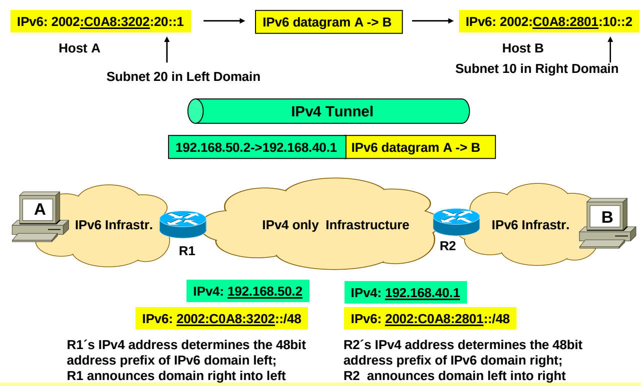
- **RFC 3056 Connection of IPv6 Domains via IPv4 Clouds**
  - Describes 6to4 tunnels

© 2012, D.I. Lindner

6to4 Tunnel 1

- **Automatic tunnel method to connect isolated IPv6 islands over IPv4 infrastructure**
- **Every IPv6 island**
  - Receives an 48 bit IPv6 prefix which is the concatenation of 2002::/16 with the 32 bit of routers IP address on the IPv4 side of the router
    - 2002:<192.168.1.4>::/48
  - Remaining 16 bits can be used for subnetting of the IPv6 island (note: last 64 bits are the Interface-ID)
  - 2002::/16 prefix is exclusively reserved for 6to4
- **This prefix will be announced by the router**
  - Towards the other side of the tunnel and hence as IPv6 Net-ID in the other IPv6 island

6to4 Tunnel 2



© 2012, D.I. Lindner

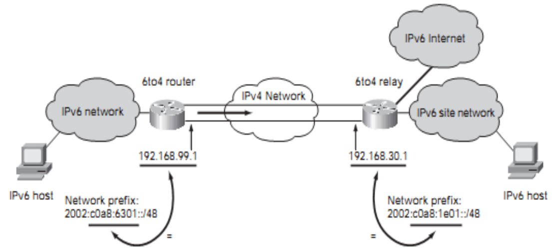
6to4 Tunnel 3

- **Whenever a IPv6 end-systems**
  - Has to transmit an packet to a destination address which starts with such a prefix then the packet is sent to the router which announced this prefix with normal IPv6 technology
- **The receiving router**
  - Encapsulates the packet in IPv4 and forwards it to the other side of the tunnel
- **Minimal manual configuration**
  - Neither an IPv4-compatible IPv6 address nor an configured tunnel is necessary for an IPv6 host
  - But an according IPv6 address plan must be implemented
    - Alternative: 6to4 router announces a default route ::/0
  - 6to4 mechanism is implemented only in the border-routers (so called 6to4 routers)

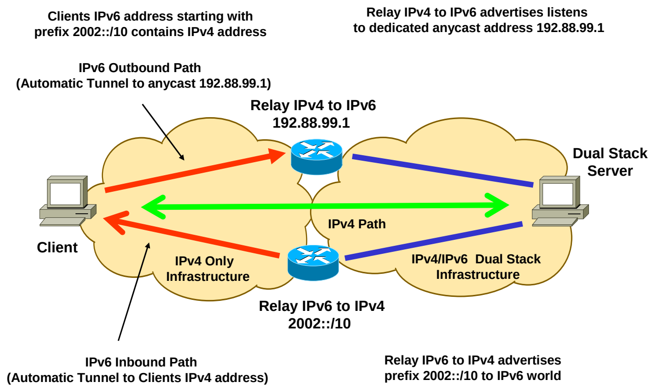
6to4 Relay

- **But how to reach the real IPv6 Internet in such a scenario?**
  - Other addresses than with prefix 2002::/48
- **Use a 6to4 relay router**
  - Acting as Default Router for all the 6to4 routers

6to4 Relay



6to4 Tunneling with Relays



Transition Approaches

- **Dual-Stack Mechanisms**
  – Dual-Stack
  – Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  – IPv4-Compatible Tunnel
  – 6to4
  – Tunnel Broker
  – 6over4
  – Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  – Teredo
- **Translation**
  – Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  – Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  – NAT444, NAT64/DNS64, 6RD, DS-Lite
  – 6PE, 6VPE
  – SHIM6 and LISP

Tunnel Broker

– RFC 3053 IPv6 Tunnel Broker (Status: INFORMATIONAL)
– In general there is some manual configuration needed to establish tunneling
– With a “Tunnel Broker” you can implement an IPv6 to IPv4 tunnel automatically
  - Clients send IPV4 HTTP request to get the information which tunnel router for a given IPv6 destination should be used
  - Tunnel broker configure tunnel information at the tunnel router
– Tunnel Broker manages activation, maintenance and termination of the tunnel
– Allows web based setup of a tunnel

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Transition Approaches

- **Dual-Stack Mechanisms**
  – Dual-Stack
  – Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  – IPv4-Compatible Tunnel
  – 6to4
  – Tunnel Broker
  – 6over4
  – Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  – Teredo
- **Translation**
  – Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  – Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  – NAT444, NAT64/DNS64, 6RD, DS-Lite
  – 6PE, 6VPE
  – SHIM6 and LISP

6over4

– RFC 2529 Transmission of IPv6 over IPv4 Domains without Explicit Tunnels (PROPOSED STANDARD)
– Allows isolated IPv6 hosts to communicate over an IPv4 infrastructure without explicit tunnels
  - Using an IPv4 multicast domain as their virtual local-link
  - Using ordinary IPv4 multicast to transport the IPv6 packet
  - All IPv6 hosts become IPv4 multicast members forming one big virtual local-link by doing this
  - IPv6 packets are transported in IPv4 multicasts packets with IPv4 protocol = 41
– Neither an IPv4-compatible IPv6 address nor an configured tunnel is necessary for an IPv6 host
  - Instead the IPv6 address is configured automatically from the IPv4 address
  - Format: Link-local prefix (FE80::/16) concatenated with the 32bit IPv4 address

Transition Approaches

- **Dual-Stack Mechanisms**
  - Dual-Stack
  - Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  - IPv4-Compatible Tunnel
  - 6to4
  - Tunnel Broker
  - 6over4
  - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  - Teredo
- **Translation**
  - Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  - Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  - NAT444, NAT64/DNS64, 6RD, DS-Lite
  - 6PE, 6VPE
  - SHIM6 and LISP

ISATAP 1

- **ISATAP connects IPv6 hosts over IPv4 networks**
  - Intra-Site Automatic Tunnel Addressing Protocol
    - RFC 4214 (Experimental)
- **Every IPv6 host**
  - Builds a 64 bit Interface ID which is the concatenation of 24 bit IANA - Code 0x00005E + 0xFE + <32bit IP address w.x.y.z>
    - ::0:5EFE:w.x.y.z
  - With this interface ID a link-local IPv6 ISATAP address can be built:
    - FE80::0:5EFE:w.x.y.z
- **Using such addresses**
  - Every IPv6 host can communicate with every other ISATAP host by encapsulating IPv6 into IPv4
    - The IPv4 address is derived from the ISATAP IPv6 address
  - NBMA style for a flat IPv6 network

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

ISATAP 2

- **ISATAP can also connects IPv6 hosts over IPv4 networks to a ISATAP router**
  - Which has a connection to the real IPv6 domain and the view of the real IPv6 addresses
  - Which advertises address prefixes to identify the logical subnet on which ISATAP hosts are located.
  - ISATAP hosts use the advertised address prefixes to configure global ISATAP addresses
- **ISATAP hosts**
  - Are configured with a default route towards a ISATAP router and will forward all IPv6 packets which can not directly be reached to this router
  - Packets which can be directly reached starts with an IPv6 prefix FE80::0:5EFE::/96

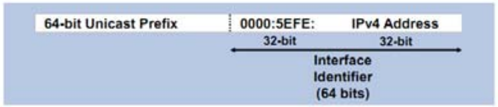
ISATAP 3

- **How to find ISATAP router?**
  - Host asks DNS for ISATAP and get an IPv4 address of this router
  - Get the IPv6 ISATAP prefix from this router by router solicitation done via IPv4 encapsulated traffic and form a global IPv6 ISATAP address

ISATAP Example 1

Intrasite Automatic Tunnel Address Protocol

Use IANA's OUI 00-00-5E and
Encode IPv4 Address as Part of EUI-64



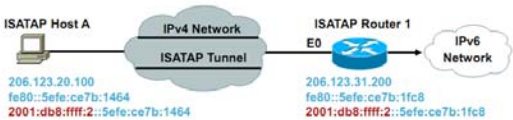
- ISATAP is used to tunnel IPv4 within as administrative domain (a site) to create a virtual IPv6 network over a IPv4 network
- Supported in Windows XP Pro SP1 and others

Source: CiscoLive2011/2012 Presentation

337

ISATAP Example 2

Automatic Address Assignment of Host and Router



- ISATAP host A receives the ISATAP prefix 2001:db8:ffff:2::/64 from ISATAP Router 1
- When ISATAP host A wants to send IPv6 packets to 2001:db8:ffff:2::5efe:ce7b:1fc8, ISATAP host A encapsulates IPv6 packets in IPv4. The IPv4 packets of the IPv6 encapsulated packets use IPv4 source and destination address.

Source: CiscoLive2011/2012 Presentation

338

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Transition Approaches

- **Dual-Stack Mechanisms**
  - Dual-Stack
  - Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  - IPv4-Compatible Tunnel
  - 6to4
  - Tunnel Broker
  - 6over4
  - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  - Teredo
- **Translation**
  - Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  - Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  - NAT444, NAT64/DNS64, 6RD, DS-Lite
  - 6PE, 6PEV
  - SHIM6, LISP

© 2012, D.I. Lindner IPv6, v4.7 339

Teredo 1

- **Aka as IPv4 network address translator (NAT) traversal (NAT-T) for IPv6**
  - Provides address assignment and host-to-host automatic tunneling for unicast IPv6 connectivity across the IPv4 Internet when IPv6/IPv4 hosts are located behind one or multiple IPv4 NATs
- **Microsoft's solution for SOHO**
  - NAT aware transition mechanism for providing dual stack hosts behind a NAT device with global IPv6 address
  - These hosts are also reachable from the outside
- **To traverse IPv4 NATs**
  - IPv6 packets are sent as IPv4-based User Datagram Protocol (UDP) messages. UDP messages can be translated by most NATs and can traverse multiple layers of NATs.

© 2012, D.I. Lindner IPv6, v4.7 340

Teredo 2

- **6to4 <->Teredo**
  - 6to4 router support is required in the edge device that is connected to the Internet. But this is not widely supported by IPv4 NATs. Even if the NAT were 6to4-enabled, 6to4 would still not work for configurations in which there are multiple NATs between a site and the Internet.
- **Teredo resolves the issues of the lack of 6to4 functionality in modern-day NATs or multi-layered NAT configurations**
  - By tunneling IPv6 packets between the hosts within the sites
- **Teredo is designed as a last resort transition technology for IPv6 connectivity**
  - If native IPv6, 6to4, or ISATAP connectivity is present between communicating nodes, Teredo is not used. As more IPv4 NATs are upgraded to support 6to4 and IPv6 connectivity become ubiquitous, Teredo will be used less and less, until eventually it is not used at all.

Teredo 3

- **Teredo client**
  - An IPv6/IPv4 node that supports a Teredo tunneling interface through which packets are tunneled to either other Teredo clients or nodes on the IPv6 Internet (through a Teredo relay)
- **Teredo server**
  - An IPv6/IPv4 node that is connected to both the IPv4 Internet and the IPv6 Internet. The role of the Teredo server is to assist in the initial configuration of Teredo clients and to facilitate the initial communication between either different Teredo clients or between Teredo clients and IPv6-only hosts.
- **Teredo relay**
  - An IPv6/IPv4 router that can forward packets between Teredo clients on the IPv4 Internet and IPv6-only hosts on the IPv6 Internet.

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Teredo Example 1

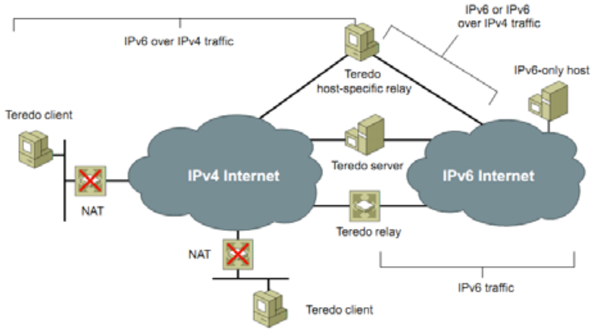
What Is Teredo?

- RFC4380
- Tunnel IPv6 through NATs (NAT types defined in RFC3489)
  - Full Cone NATs (aka one-to-one)—Supported by Teredo
  - Restricted NATs—Supported by Teredo
  - Symmetric NATs—Supported by Teredo with Vista/Server 2008 if only one Teredo client is behind a Symmetric NATs
- Uses UDP port 3544
- Is complex—many sequences for communication and has several attack vectors
- Available on:
  - Microsoft Windows XP SP1 w/Advanced Networking Pack
  - Microsoft Windows Server 2003 SP1
  - Microsoft Windows Vista (enabled by default—inactive until application requires it)
  - Microsoft Server 2008 http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/teredo.mspx
  - Linux, BSD and Mac OS X—"Miredo" http://www.simphalempin.com/dev/miredo/

Source: CiscoLive2011/2012 Presentation

Teredo Example 2

Teredo Overview



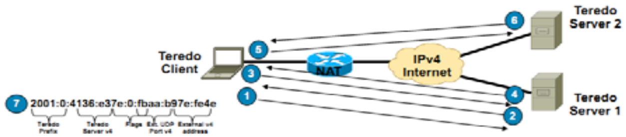
*From Microsoft "Teredo Overview" paper

Source: CiscoLive2011/2012 Presentation

Teredo Example 3

Initial Configuration for Client

1. RS message sent from Teredo client to server—RS from LL address with Cone flag set
2. Server responds with RA—RS has Cone flag set—server sends RA from alternate v4 address—if client receives the RA, client is behind cone NAT
3. If RA is not received by client, client sends another RA with Cone flag not set
4. Server responds with RA from v4 address = destination v4 address from RS—if client receives the RA, client is behind restricted NAT
5. To ensure client is not behind symmetric NAT, client sends another RS to secondary server
6. 2nd server sends an RA to client—client compares mapped address and UDP ports in the Origin indicators of the RA received by both servers. If different, then the NAT is mapping same internal address/port to different external address/port and NAT is a symmetric NAT
7. Client constructs Teredo address from RA
   First 64 bits are the value from prefix received in RA (32 bits for IPv6 Teredo prefix + 32 bits of hex representation of IPv4 Teredo server address)
   Next 16 bits are the Flags field (0x0000 = Restricted NAT, 0x8000 = Cone NAT)
   Next 16 bits are external obscured UDP port from Origin indicator in RA
   Last 32 bits are obscured external IP address from Origin indicator in RA



Source: CiscoLive2011/2012 Presentation

Teredo Example 4

What Happens on the Wire—2

```
No.  Time       Source                      Destination            Protocol Info
  28 33.595460  fe80::8000:ffff:ffff:fffd   ff02::2                ICMPv6   Router solicitation
Internet Protocol, Src: 172.16.1.103 (172.16.1.103), Dst: 65.54.227.126 (65.54.227.126)
User Datagram Protocol, Src Port: 1109 (1109), Dst Port: 3544 (3544)

No.  Time       Source                      Destination            Protocol Info
  29 37.593598  fe80::8000:ffff:ffff:fffd   ff02::2                ICMPv6   Router solicitation
Internet Protocol, Src: 172.16.1.103 (172.16.1.103), Dst: 65.54.227.126 (65.54.227.126)

No.  Time       Source                      Destination            Protocol Info
  31 45.546052  fe80::ffff:ffff:fffd        ff02::2                ICMPv6   Router solicitation
Internet Protocol, Src: 172.16.1.103 (172.16.1.103), Dst: 65.54.227.127 (65.54.227.127)
User Datagram Protocol, Src Port: 1109 (1109), Dst Port: 3544 (3544)

No.  Time       Source                      Destination            Protocol Info
  32 46.039706  fe80::8000:f227:bec9:1c81   fe80::ffff:ffff:fffd   ICMPv6   Router advertisement
Internet Protocol, Src: 65.54.227.127 (65.54.227.127), Dst: 172.16.1.103 (172.16.1.103)
User Datagram Protocol, Src Port: 3544 (3544), Dst Port: 1109 (1109)
 Teredo Origin Indication header
     Origin UDP port: 1109
     Origin IPv4 address: 70.120.2.1 (70.120.2.1)
Prefix: 2001:0:4136:e37e::

No.  Time       Source                      Destination            Protocol Info
  33 46.093832  fe80::ffff:ffff:fffd        ff02::2                ICMPv6   Router solicitation
Internet Protocol, Src: 172.16.1.103 (172.16.1.103), Dst: 65.54.227.126 (65.54.227.126)
User Datagram Protocol, Src Port: 1109 (1109), Dst Port: 3544 (3544)

No.  Time       Source                      Destination            Protocol Info
  34 46.398745  fe80::8000:f227:bec9:1c81   fe80::ffff:ffff:fffd   ICMPv6   Router advertisement
Internet Protocol, Src: 65.54.227.126 (65.54.227.126), Dst: 172.16.1.103 (172.16.1.103)
   Teredo Origin Indication header
     Origin UDP port: 1109
     Origin IPv4 address: 70.120.2.1 (70.120.2.1)
Prefix: 2001:0:4136:e37e::
```

Send RS Cone Flag=1 (Cone NAT), every 4 seconds

If no reply, send Flag=0 (restricted NAT)

Receive RA with Origin header and prefix

Send RS to 2nd server to check for symmetric NAT

Compare 2nd RA—Origin port/address from 2nd server

Source: CiscoLive2011/2012 Presentation

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Teredo Example 4

What Happens on the Wire—3 (Cont.)

```
Interface 7: Teredo Tunneling Pseudo-Interface

Addr Type  DAD State  Valid Life   Pref. Life   Address
---------  ---------- ------------ ------------ -----------------------------
Public     Preferred      infinite     infinite 2001:0:4136:e37e:0:fbaa:b97e:fe4e
Link       Preferred      infinite     infinite fe80::ffff:ffff:fffd
```

```
C:\>ping www.kame.net

Pinging www.kame.net [2001:200:0:8002:203:47ff:fea5:3085] with 32 bytes of data


Reply from 2001:200:0:8002:203:47ff:fea5:3085:  time=829ms
Reply from 2001:200:0:8002:203:47ff:fea5:3085:  time=453ms
Reply from 2001:200:0:8002:203:47ff:fea5:3085:  time=288ms
Reply from 2001:200:0:8002:203:47ff:fea5:3085:  time=438ms
```

Source: CiscoLive2011/2012 Presentation

Transition Approaches

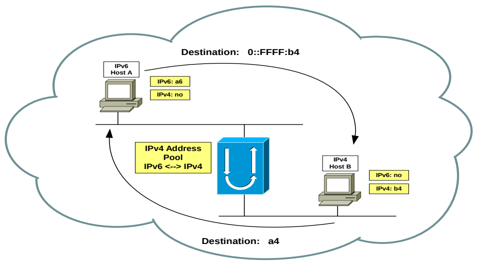
- **Dual-Stack Mechanisms**
  - Dual-Stack
  - Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  - IPv4-Compatible Tunnel
  - 6to4
  - Tunnel Broker
  - 6over4
  - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  - Teredo
- **Translation**
  - Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  - Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  - NAT444, NAT64/DNS64, 6RD, DS-Lite
  - 6PE, 6PEV
  - SHIM6, LISP

Stateless IP/ICMP Translator (SIIT)

- RFC 2765 (Proposed Standard)
- Communication between IPv4 only and IPv6 only hosts - no need for two different protocol stacks (IPv4 or IPv6)
- Algorithm in explicit “Translation boxes” which translates between IPv6 and IPv4 packet header (IP and ICMP)
- Only packet header information is translated
- Translates in stateless mode
- SIIT neither translates options of IPv4 packets to IPv6 nor extension headers of IPv6 to IPv4

Stateless IP/ICMP Translator (SIIT) (cont.)

- Translation box assigns IPv6 site an IPv4 address
- IPv6 uses an IPv4-mapped IPv6 address (0::FFFF:a.b.c.d) to send packets to IPv4 site

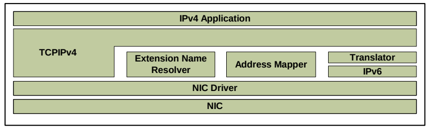


Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Bump in the Stack (BIS)

- RFC 2767 (Informational)
- Allows IPv4-only applications on a dual stack host to communicate with IPv6-only hosts
- Additional module between IP Layer and NIC driver is necessary
- Same functionality as SIIT
- For OS where source code is not available

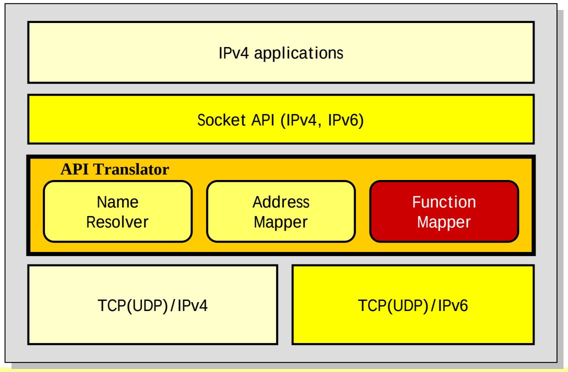


Bump in the API (BIA)

- RFC 3338 (Experimental)
- Allows IPv4-only applications on a dual stack host to communicate with IPv6-only hosts
- But the bump layer is inserted higher up, as part of the socket layer, enabling the interception of Socket API calls.
- The location of the BIA module avoids the translation of IP packets and modifications in the operating system kernel
- BIA implementations consist of three bump components
  - Name resolver
  - Address mapper
  - Function mapper
    - Intercepts IPv4 socket function calls and translates them to the equivalent IPv6 socket calls

Bump in the API (cont.)

– Architecture



Network Address Translation Protocol Translation (NAT-PT)

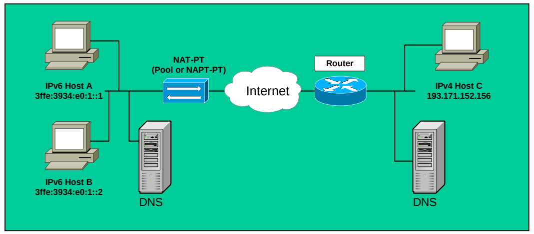
– RFC 2766 (Proposed Standard)
– Provides IPv6 only node communication with IPv4 only node
– Address translation is identical to SIIT, but IPv4 addresses are assigned per TCP/UDP session (not per host)
– IPv4 addresses will be dynamically assigned to IPv6 nodes (NAT-PT possible)
– all traffic has to use the same NAT-PT router/translator
– bidirectional NAT-PT possible
– DNS queries and responses are translated by an application level gateway (DNS-ALG) in the device

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Network Address Translation Protocol Translation (NAT-PT) (cont.)

– bidirectional NA(P)T-PT



Transport Relay Translator (TRT)

– RFC 3142 (Informational)
– Transport layer relays can also be extended into IPv6/IPv4 translators.
– TRT (Transport Relay Translator)
  - TRT translates between TCP/UDPv6 and TCP/UDPv4 sessions
  - Communication is initiated from the IPv6 side
  - The routing information is configured to route this prefix toward the dual-stack TRT router, which terminates the IPv6 session and initiates IPv4 communication to the final destination

SOCKS64

- RFC 3089 (Informational)
- SOCKS64 uses a dual-stacked SOCKS64 router and socksified applications to enable communication between IPv4 and IPv6 nodes
- Applications are socksified by using a special SOCKS64 library that replaces Socket and DNS APIs
- The SOCKS64 library intercepts session-initiating DNS name lookups from the end system application and responds with “fake” IP addresses mapped for the given session
- The SOCKS64 library also issues session control calls to the local SOCKS64 router

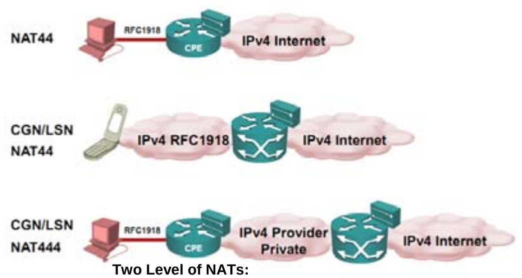
Transition Approaches

- **Dual-Stack Mechanisms**
  - Dual-Stack
  - Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  - IPv4-Compatible Tunnel
  - 6to4
  - Tunnel Broker
  - 6over4
  - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  - Teredo
- **Translation**
  - Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  - Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  - NAT444, NAT64/DNS64, 6RD, DS-Lite
  - 6PE, 6PEV
  - SHIM6, LISP

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Prolonging Traditional NAT (NAT44) Large Scale NAT – Carrier Grade NAT - NAT444

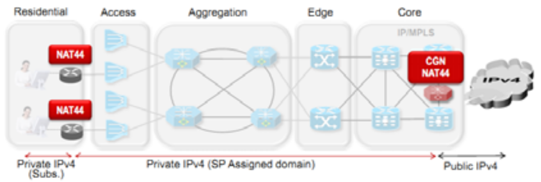


Two Level of NATs:
Email and Browsing will work but what about the rest
Absolutely will break many peer-to-peer applications !!!

Source: Webinar “Service_Provider_IPv6_Intro” Ivan Pepelnjak 2011

NAT444

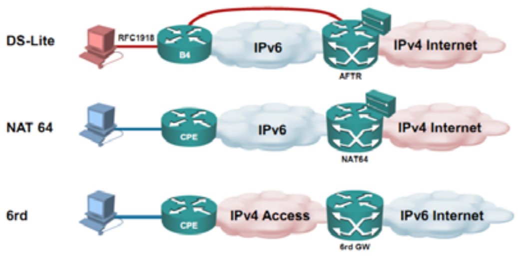
Public IPv4 Exhaustion with NAT444 Solution



- Short-term solution to public IPv4 exhaustion issues without any changes on RG and SP Access/Aggregation/Edge infrastructure
- Subscriber uses NAT44 (i.e. IPv4 NAT) in addition to the SP using CGN with NAT44 within its network
- CGN NAT44 multiplexes several customers onto the same public IPv4 address
- CGN performance and capabilities should be analyzed in planning phase
- Long-term solution is to have IPv6 deployed

Source: CiscoLive2011/2012 Presentation

Transition Scenarios Provider 1

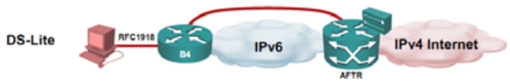


Source: Webinar "Service_Provider_IPv6_Intro" Ivan Pepelnjak 2011

361

Transition Scenarios Provider 2

DS-Lite: IPv4 ony host, Dual Stack Lite in the CPE equipment which is already managed by the service provider using IPv6
Actually bridging customer IPv4 by tunneling to AFTR over IPv6 domain
AFTR does Large Scale NAT but only one level of NAT



Source: Webinar "Service_Provider_IPv6_Intro" Ivan Pepelnjak 2011

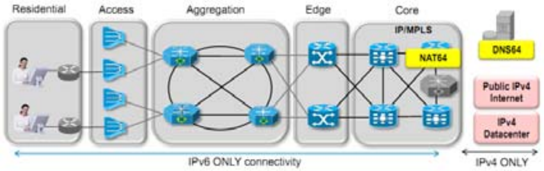
362

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

DS-Lite / AFT64

Connecting IPv6-only with IPv4-only: AFT64



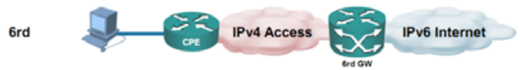
- AFT64 technology is only applicable in case where there are IPv6 only end-points that need to talk to IPv4 only end-points (AFT64 for going from IPv6 to IPv4)
- AFT64:= "stateful v6 to v4 translation" or "stateless translation", ALG still required
- Key components includes NAT64 and DNS64
- Assumption: Network infrastructure and services have fully transitioned to IPv6 and IPv4 has been phased out

Source: CiscoLive2011/2012 Presentation

363

Transition Scenarios Provider 2

6RD: Rapid Deployment with IPv6 only host
But only an IPv4 access network is operated by the service provider
Done by automatic tunneling method 6rd: maps IPv4 address of CPE to an internal used IPv6 address

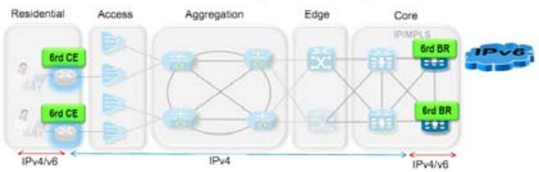


Source: Webinar "Service_Provider_IPv6_Intro" Ivan Pepelnjak 2011

364

6RD

IPv6 over IPv4 via 6rd (RFC 5569)



- Introduction of two Components: 6rd CE (Customer Edge) and 6rd BR (Border Relay)
- Automatic Prefix Delegation on 6rd CE
- Simple, stateless, automatic IPv6-in-IPv4 encap and decap functions on 6rd (CE & BR)
- IPv6 traffic automatically follows IPv4 Routing
- 6rd BRs addressed with IPv4 anycast for load-balancing and resiliency
- Limited investment & impact on existing infrastructure

Source: CiscoLive2011/2012 Presentation

365

Transition Scenarios Provider 2

NAT64: **Stateful, defined in RFC 6146**
IPv6 only host connected to the IPv6 world
Single instance of NAT used if old IPv4 content should be reached
It is not NAT-PT (RFC 2766 -> Deprecated by RFC 4966)

NAT 64

Source: Webinar "Service_Provider_IPv6_Intro" Ivan Pepelnjak 2011

366

© 2012, D.I. Lindner

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

NAT64 Topology

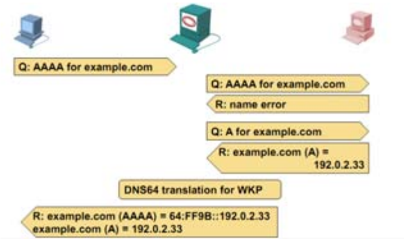
- Whole IPv4 Internet is mapped into a single IPv6 prefix
- NAT64 device advertises this single IPv6 address into the IPv6 world, so traffic from an IPv6 only hosts will go via NAT64 device
- NAT64 device can be located anywhere – not necessarily in the forwarding path like normal NAT
- DNS64 device resolves names of IPv4 only servers to be reachable via NAT64 device

Source: Webinar "Service_Provider_IPv6_Intro" Ivan Pepelnjak 2011

367

DNS64 in Action 1

- IPv6 client sends out only a DNS IPv6 AAAA request for an IPv4 server
- DNS64 device will asks DNS servers for this AAAA request; will get an error; will try it with an DNS IPv4 A request and will get an answer
- DNS64 algorithmically translates this in a Well Known IPv6 Prefix (WPK) with the IPv4 address embedded; no state and no translation table is necessary -> quite simple
- IPv6 host – if dual stacked - can also find out IPv4 address from the answer



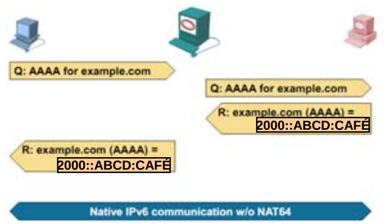
Source: Webinar "Service_Provider_IPv6_Intro" Ivan Pepelnjak 2011

368

© 2012, D.I. Lindner

DNS64 in Action 2

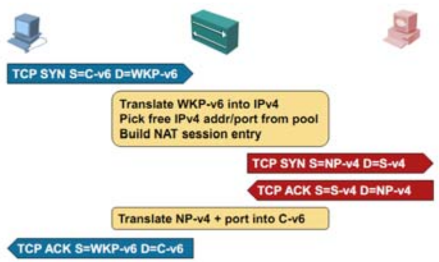
- If IPv6 client sends a DNS IPv6 AAAA request for an IPv6 server
- DNS64 device will asks DNS servers for this AAAA request; will get an answer
- Native IPv6 communication is possible without NAT64



Source: Webinar "Service_Provider_IPv6_Intro" Ivan Pepelnjak 2011 369

NAT64 in Action

- Works like normal NAT (static, dynamic with PAT) already used in IPv4 (connecting IPv4 clients with private addresses to the globally addressed IPv4 Internet)
- NAT64 could be stateless or stateful



Source: Webinar "Service_Provider_IPv6_Intro" Ivan Pepelnjak 2011 370

NAT Types

- **NAT64**
  - Maps IPv4 addresses into a subset of IPv6 addresses (Well Known NAT64 prefix plus IPv4 address)
- **Stateless NAT64**
  - 1 – 1 mapping between IPv4 and IPv6 addresses
  - Requires special format of IPv6 addresses (no SLAAC) ???
  - Does not solve IPv4 address exhaustion
  - But useful for IPv4 clients accessing IPv6 servers
- **Stateful NAT64**
  - Many IPv6 addresses into one IPv4 address
  - Put some relief to IPv4 address shortage
  - IPv6 clients can access IPv4 servers
  - IPv4 clients can not access IPv6 servers
  - What's about peer-to-peer?

© 2012, D.I. Lindner IPv6, v4.7 371

Transition Approaches

- **Dual-Stack Mechanisms**
  - Dual-Stack
  - Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  - IPv4-Compatible Tunnel
  - 6to4
  - Tunnel Broker
  - 6over4
  - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  - Teredo
- **Translation**
  - Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  - Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  - NAT444, NAT64/DNS64, 6RD, DS-Lite
  - 6PE, 6PEV
  - SHIM6, LISP

© 2012, D.I. Lindner IPv6, v4.7 372

6PE 1

IPv6 Provider Edge Router (6PE) over MPLS

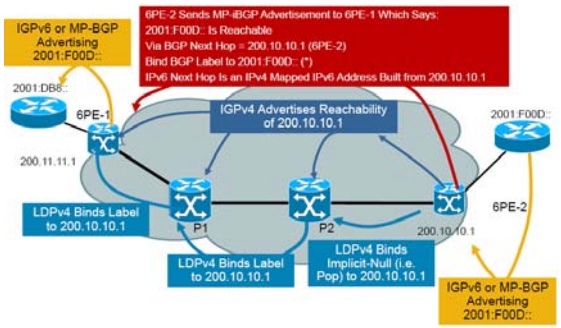
- IPv6 global connectivity over and IPv4-MPLS core
- Transitioning mechanism for providing unicast IP
- PEs are updated to support dual stack/6PE
- IPv6 reachability exchanged among 6PEs via iBGP (MBGP)
- IPv6 packets transported from 6PE to 6PE inside MPLS

Source: CiscoLive2011/2012 Presentation

373

6PE 2

6PE Routing/Label Distribution



Source: CiscoLive2011/2012 Presentation

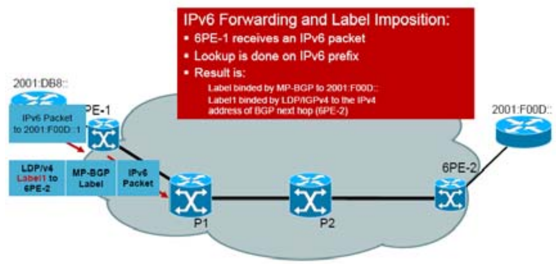
374

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

6PE 3

6PE Forwarding (6PE-1)

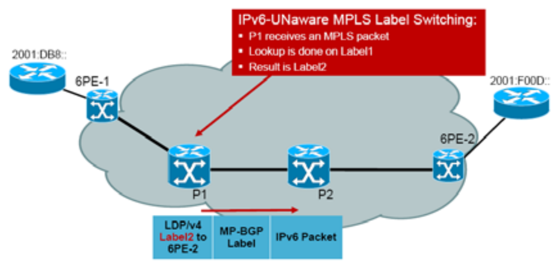


Source: CiscoLive2011/2012 Presentation

375

6PE 4

6PE Forwarding (P1)

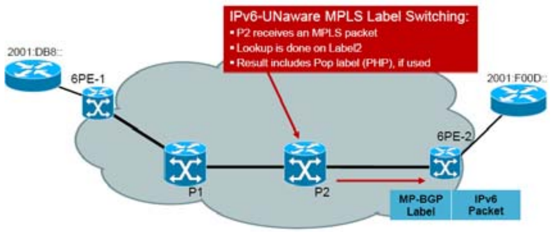


Source: CiscoLive2011/2012 Presentation

376

6PE 5

6PE Forwarding (P2)

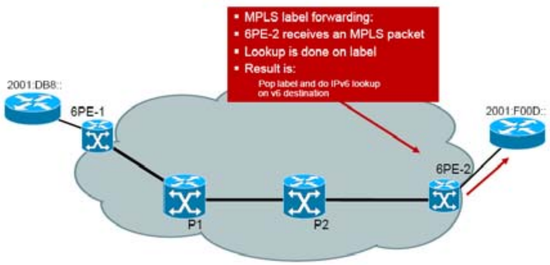


Source: CiscoLive2011/2012 Presentation

377

6PE 6

6PE Forwarding (6PE-2)



Source: CiscoLive2011/2012 Presentation

378

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

6PE 7

6PE Benefits/Drawbacks

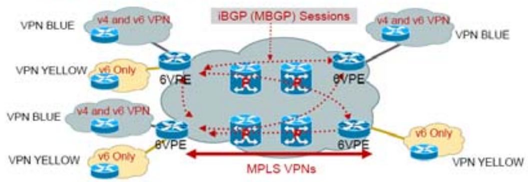
- Core network (Ps) untouched
- IPv6 traffic inherits MPLS benefits (fast re-route, TE, etc.)
- Incremental deployment possible (i.e., only upgrade the PE routers which have to provide IPv6 connectivity)
- Each site can be v4-only, v4VPN-only, v4+v6, v4VPN+v6
- P routers won't be able to send ICMPv6 messages (TTL expired, trace route)
- Scalability issues arise as a separate RIB and FIB is required for each connected customer
- Good solution only for SPs with limited devices in PE role
- Cisco 6PE Documentation/Presentations:
  http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_data_sheet09186a008052edd3.html

Source: CiscoLive2011/2012 Presentation

379

6VPE 1

6VPE over MPLS



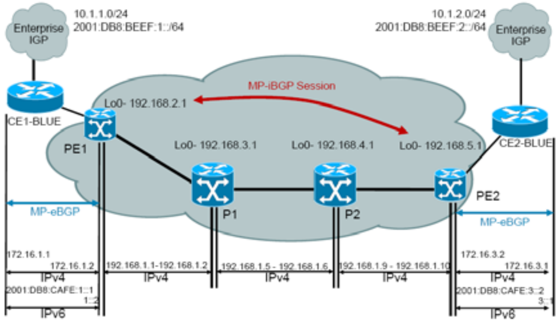
- 6VPE ~ IPv6 + BGP-MPLS IPv4VPN + 6PE
- Cisco 6VPE is an implementation of RFC4659
- VPNv6 address:
  Address including the 64 bits route distinguisher and the 128 bits IPv6 address
- MP-BGP VPNv6 address-family:
  AFI "IPv6" (2), SAFI "VPN" (128)
- VPN IPv6 MP_REACH_NLRI
  With VPNv6 next-hop (192bits) and NLRI in the form of <length, IPv6-prefix, label>
- Encoding of the BGP next-hop

Source: CiscoLive2011/2012 Presentation

380

6VPE 2

6VPE Example Design
Addressing/Routing



Source: CiscoLive2011/2012 Presentation

381

6VPE 3

6VPE Summary

- RFC4659: BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN
- 6VPE simply adds IPv6 support to current IPv4 MPLS VPN offering
- For end-users: v6-VPN is same as v4-VPN services (QoS, hub and spoke, internet access, etc.)
- For operators:
  - Same configuration operation for v4 and v6 VPN
  - No upgrade of IPv4/MPLS core (IPv6 unaware)
- Cisco 6VPE Documentation:
  - http://www.cisco.com/en/US/docs/net_mgmt/ip_solution_center/5.2/mpls_vpn/user/guide/ipv6.html

Source: CiscoLive2011/2012 Presentation

382

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

Transition Approaches

- **Dual-Stack Mechanisms**
  - Dual-Stack
  - Dual-Stack Dominant Transition Mechanism (DSTM)
- **Tunneling Mechanisms**
  - IPv4-Compatible Tunnel
  - 6to4
  - Tunnel Broker
  - 6over4
  - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
  - Teredo
- **Translation**
  - Stateless IP/ICMP Translator (SIIT) / Bump in the Stack (BITSv6)
  - Bump in the API (BIA) / Network Address Translation -Protocol Translation (NAT-PT) / Transport Relay Translator (TRT) / SOCKS64
  - NAT444, NAT64/DNS64, 6RD, DS-Lite
  - 6PE, 6PEV
  - SHIM6, LISP

© 2012, D.I. Lindner IPv6, v4.7 383

Multihoming 1

Multi-homing Goals

- Redundancy: Insulate from failures in upstream providers
- Load Sharing: Concurrent use of multiple transit provider and distribute traffic load
- Simple and scalable solution
- Minimal impact on existing network devices

Source: CiscoLive2011/2012 Presentation

384

Multihoming 2

IPv6 Multi-homing Solutions

- Traditional Multi-homing (PI)
  - Advertise address space to multiple transit providers
  - Longer prefixes to prefer a ISP and/or Load balancing
  - Large BGP routing table
- Dual Address blocks from upstream ISPs (PA)
  - Receive and utilize address blocks from both ISPs
  - Overhead and renumbering problems
- NAT66/Proxy
  - Address independence without PI
  - NAT again?
- Multi-homing without NAT or PI
  - SHIM6 – Host based
  - LISP – Network based

Source: CiscoLive2011/2012 Presentation

385

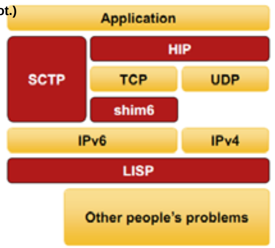
Multihoming 3

SCTP (Stream Control Transportation Prot.)
- New transport protocol
- Supports multihoming & streams

LISP (Locator/ID Seperation Protocol)
- Global directory-driven mGRE/NHRP-like solution

shim6
- Add-on for TCP over IPv6 and UDP streams over IPv6

HIP (Host Identity Protocol)
- Replaces IP address with signed host identifiers



Source: Webinar "Upcoming_Internet_challenges" Ivan Pepelnjak 2011

386

SHIM6 1

SHIM6: Host based Multi-homing Solution (RFC 5533)

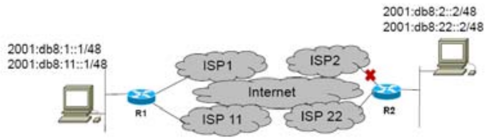
- Shim6 protocol, a layer 3 shim for providing locator agility with failover capabilities for IPv6 nodes.
- Hosts that employ Shim6 use multiple IPv6 address prefixes and setup state with peer hosts.
- This state can later be used to failover to a different set of locators, should the original locators stop working.
- Modification of the IP stack within the protocol stack of the endpoint.
- SHIM6 Advantages:
  - Individual host implementation rather than through site-wide mechanisms
  - Multi-homing without requiring a PI IPv6 address prefix
  - Transparent to Core infrastructure devices between the end hosts

Source: CiscoLive2011/2012 Presentation

387

SHIM6 2

SHIM6: Host based Multi-homing Solution



SHIM6 protocol exchange (ULIDs) between end hosts

Host1 sends packet to Host 2 →

IPv6 header Source Locator 2001:Db8:1::1 Dest. Locator 2001:db8:22::2	Shim Source ULID 2001:db8:1::1 Dest. ULID 2001:db8:2::2	Transport Header	Payload

Source: CiscoLive2011/2012 Presentation

388

© 2012, D.I. Lindner

SHIMv6 3

- – Part of the Shim6 solution involves detecting when a currently used pair of addresses (or interfaces) between two communication nodes has failed and picking another pair when this occurs
- – the former is called "failure detection", and the latter, "locator pair exploration"

- **RFC 5334**
  - – Specifies how the level 3 multihoming Shim6 protocol (Shim6) detects failures between two communicating nodes
  - – Also specifies an exploration protocol for switching to another pair of interfaces and/or addresses between the same nodes if a failure occurs and an operational pair can be found
  - – Specifies the mechanisms and protocol messages to achieve both failure detection and locator pair exploration
  - – This part of the Shim6 protocol is called the REAchability Protocol (REAP)

LISP 1

LISP: Network based Multi-homing Solution

(Locator/ID Seperation Protocol)

- LISP decouples Internet addresses into EIDs/RLOCs.
- Network-based map-n-encap protocol implemented mostly on network edge routers
- Reduces the size and dynamic properties of the core routing tables
- LISP can also be used as IPv6 transition mechanism
- LISP Advantages:
  - Network-based solution
  - No host changes
  - No new addressing to site devices; minimal configuration changes
  - Incrementally deployable; interoperable with existing Internet

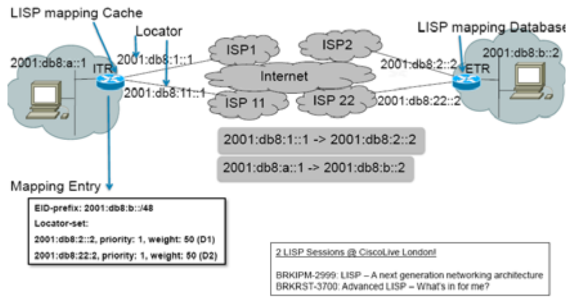
Source: CiscoLive2011/2012 Presentation

Datenkommunikation 384.081 - SS 2012

Appendix 5 - IPv6 (v4.7)

LISP 2

LISP: Network based Multi-homing Solution



Source: CiscoLive2011/2012 Presentation

Some Hints for IPv6 Rollout (Service Provider)

- – IPv6 readiness audit – from network devices to <u>applications</u>
  - Applications will be the problem because IP addresses are deeply integrated in provisioning and billing applications
  - Keep an eye on network devices if IPv6 switching / packet filtering is done in hardware or in software (later maybe a performance problem)
- – Plan address space needs and get addresses from RIR
- – Deploy IPv6 in the network -> three choices
  - Dual stack
  - Do 6PE if you have MPLS ->IPv4 core need not to be touched
    - – IPv6 is acting over backbone like IPv4 VPN over MPLS VPN IPv4 core network (second label, BGP between PE routers to transport IPv6 labels)
  - Do 6VPE if you have MPLS-VPN
- – Adapt provisioning and billing applications
- – Deploy IPv6 with enterprise customers
- – Starts consumer trials
  - Major pain because lack of CPE devices and IPv4 only access network

Residential Customers (Issues and Problems)

- Windows XP
  - Can not be IPv6 only
  - Can not resolve host names with IPv6
- Many CPE devices and set-top boxes do not support IPv6
- IPv6 multicast support for digital TV is not existing
- IPv6 to IPv4 translation to allow new IPv6 users to access old IPv4 content
- Lack of IPv6 support on DSL CPE devices and mobile networks 3G
- Lack of IPv6 DHCP snooping, ND- and RA-guard techniques on carrier Ethernet switches (securing addressing of IPv6)

Further Information IPv6 Transition

- **Internet Protocol Journal (www.cisco.com/ipj)**
  - Issue Volume 3, Number 1 (March 2000)
    - „Routing IPv6 over IPv4" – 6to4 Tunneling and Relay
  - Issue Volume 8, Number 2 (June 2005)
    - „IPv6 and MPLS" 6VPE
  - Issue Volume 11 Number 1 (March 2008)
    - „LISP" (Locator Identifier Separation Protocol)
  - Issue Volume 14, Number 1 (March 2011)
    - „Address Exhaustion"
    - "Transitional Myths"
    - "Transitioning Protocols"
  - Issue Volume 14, Number 2 (June 2011)
    - „IPv6 Site Multihoming"

Agenda

- **History**
- **IPv6**
- **ICMPv6 and Plug&Play**
- **Routing**
- **Transition**
- **Miscellaneous**

Some Problems

- Windows Teredo security problem:
  - On older versions Teredo tunnel is established automatically and workstation announces itself as IPv6 router
  - Therefore a single workstation can expose the whole network through a Teredo tunnel
  - Block Teredo udp port on your firewall
    - Terode servers are listening on udp port 3544
- Privacy Extensions for SLAAC
  - Automatic on Windows
  - Configured on Linux
- Host with random interface ID needs reverse DNS registration for easy troubleshooting

IPv6 Support on OS

Operating System Support

- Every major OS supports IPv6 today
- Windows—top-to-bottom TCP/IP stack re-design
- IPv6 is on by default and preferred over IPv4 (considering network/DNS/application support)
- Tunnels will be used before IPv4 if required by IPv6-enabled application

  ISATAP, Teredo, 6to4, configured
- All applications and services that ship with Vista/Server 2008 support IPv4 and IPv6 (IPv6-only is supported)

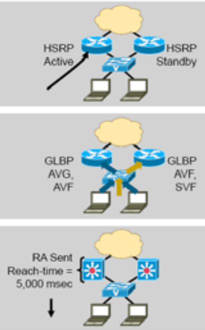
  Active Directory, IIS, File/Print/Fax, WINS/DNS/DHCP/LDAP, Windows Media Services, Terminal Services, Network Access Services—Remote Access (VPN/Dial-up), Network Access Protection (NAP), Windows Deployment Service, Certificate Services, SharePoint services, Network Load-Balancing, Internet Authentication Server, Server Clustering, etc.
- http://www.microsoft.com/technet/network/ipv6/default.mspx

Source: CiscoLive2011/2012 Presentation

397

FHRP (1)

First Hop Router Redundancy



HSRP for v6

- Modification to Neighbor Advertisement, router Advertisement, and ICMPv6 redirects
- Virtual MAC derived from HSRP group number and virtual IPv6 link-local address

GLBP for v6

- Modification to Neighbor Advertisement, Router Advertisement—GW is announced via RAs
- Virtual MAC derived from GLBP group number and virtual IPv6 link-local address

Neighbor Unreachability Detection

- For rudimentary HA at the first HOP
- Hosts use NUD "reachable time" to cycle to next known default gateway (30s by default)

No longer needed

Source: CiscoLive2011/2012 Presentation

398

FHRP (2)

First-Hop Redundancy

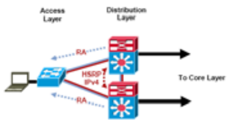
- When HSRP,GLBP and VRRP for IPv6 are not available
- NUD can be used for rudimentary HA at the first-hop (today this only applies to the Campus/DC—HSRP is available on routers)

  `(config-if)#ipv6 nd reachable-time 5000`
- Hosts use NUD "reachable time" to cycle to next known default gateway (30 seconds by default)
- Can be combined with default router preference to determine primary gw:

  `(config-if)#ipv6 nd router-preference {high | medium | low}`

```
Default Gateway . . . . . . . . . : 10.121.10.1
                                    fe80::211:bcff:fec0:d000%4
                                    fe80::211:bcff:fec0:c800%4
```

```
Reachable Time        : 6s
Base Reachable Time   : 5s
```



HSRP for IPv4
RA's with adjusted reachable-time for IPv6

Source: CiscoLive2011/2012 Presentation

399

FHRP (3)

HSRP for IPv6

- Many similarities with HSRP for IPv4
- Changes occur in Neighbor Advertisement, Router Advertisement, and ICMPv6 redirects
- No need to configure GW on hosts (RAs are sent from HSRP active router)
- Virtual MAC derived from HSRP group number and virtual IPv6 link-local address
- IPv6 Virtual MAC range:

  0005.73A0.0000 - 0005.73A0.0FFF (4096 addresses)
- HSRP IPv6 UDP Port Number 2029 (IANA Assigned)
- No HSRP IPv6 secondary address
- No HSRP IPv6 specific debug


```
interface FastEthernet0/1
 ipv6 address 2001:DB8:66:67::2/64
 ipv6 cef
 standby version 2
 standby 1 ipv6 autoconfig
 standby 1 timers msec 250 msec 800
 standby 1 preempt
 standby 1 preempt delay minimum 180
 standby 1 authentication md5 key-string cisco
 standby 1 track FastEthernet0/0
```

Host with GW of Virtual IP

```
#route -A inet6 | grep ::/0 | grep eth2
::/0       fe80::5:73ff:fea0:1               UGDA  1024   0        0 eth2
```

Source: CiscoLive2011/2012 Presentation

400

© 2012, D.I. Lindner

© 2012, D.I. Lindner

FHRP (4)

GLBP for IPv6

- Many similarities with GLBP for IPv4 (CLI, load-balancing)
- Modification to Neighbor Advertisement, Router Advertisement
- GW is announced via RAs
- Virtual MAC derived from GLBP group number and virtual IPv6 link-local address


```
interface FastEthernet0/0
 ipv6 address 2001:DB8:1::1/64
 ipv6 cef
 glbp 1 ipv6 autoconfig
 glbp 1 timers msec 250 msec 750
 glbp 1 preempt delay minimum 180
 glbp 1 authentication md5 key-string cisco
```

AVG=Active Virtual Gateway
AVF=Active Virtual Forwarder
SVF=Standby Virtual Forwarder

Source: CiscoLive2011/2012 Presentation

401

© 2012, D.I. Lindner