IP Multicast

Compendium

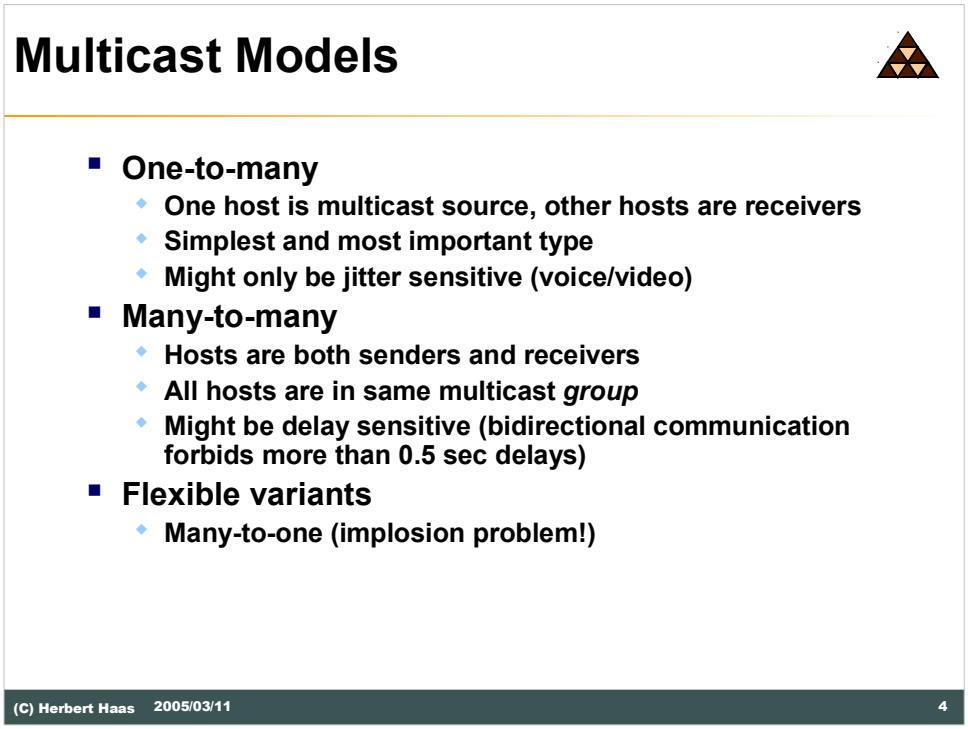
Table of Contents:

Introduction

New IP Applications

- Corporate Broadcasts
- Distance Learning/Training
- Video Conferencing
- Whiteboard/Collaboration
- Multicast File Transfer
- Multicast Data and File Replication
- Real-Time Data Delivery for Financial Applications
- Video-On-Demand
- Live TV and Radio Broadcast to the Desktop
- Multicast Games

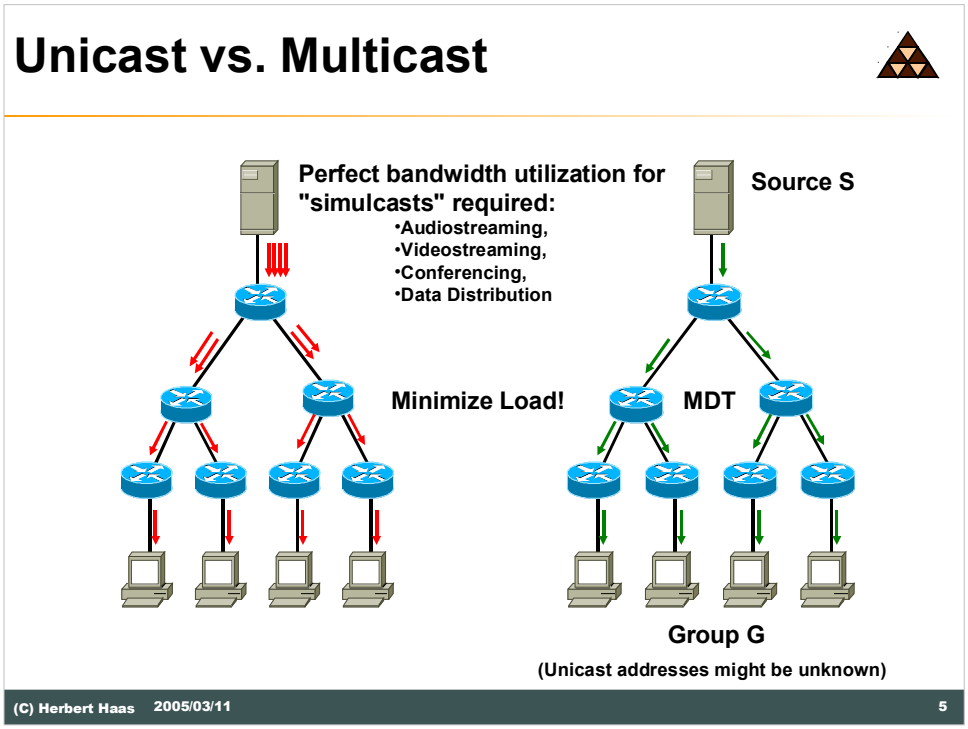
Real-time applications include games, live broadcasts, financial data delivery, whiteboard collaboration, and video conferencing. Non-real-time applications include file transfer, data and file replication, and video on demand (VoD).



The **many-to-many** multicast concept supports several new applications such as collaboration systems, concurrent processing, and distributed interactive simulations.

Other models involve the **many-to-one** model, where many receivers may send data back to one sender (similar to few-to-many). These models are typically used in financial applications/networks. Consider auctions for example. Here any number of receivers might send data back to a source (via unicast or multicast). Note the "implosion problem" as a response storm might occur when responses arrive simultaneously.

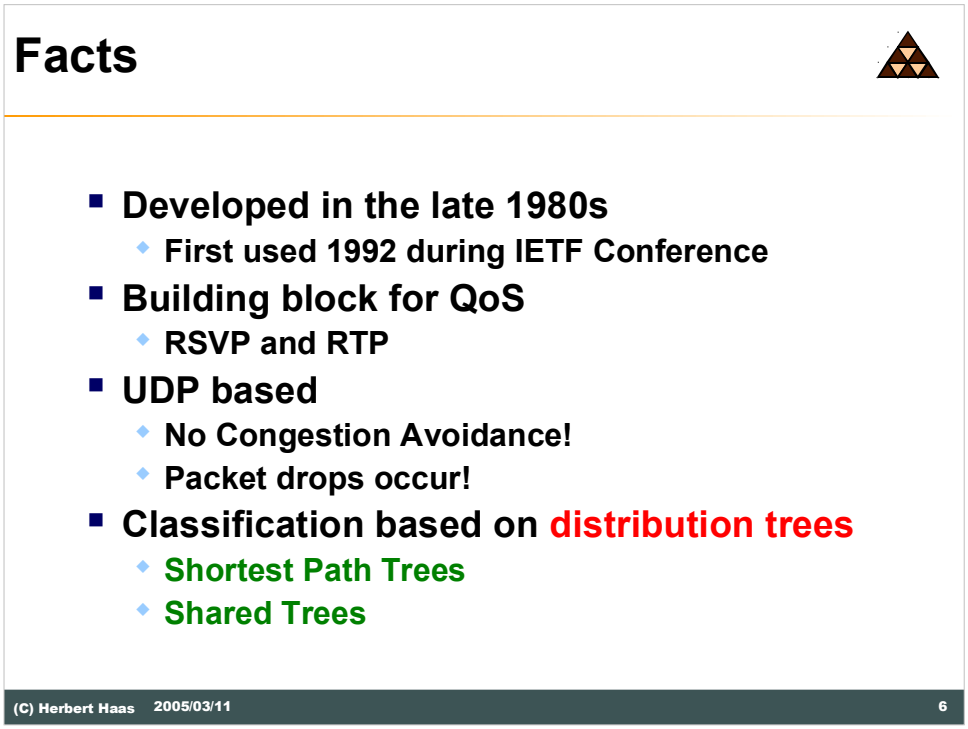
Modern solutions to these problems involve **bidirectional trees** and other mechanisms. For example, responses could be sent "out-of-band". However, most implementations require modifications of the applications.



If several (if not thousands) of users should receive a certain service then there are two choices of implementation: Either sending multiple unicast packets or a single multicast packet. The latter solution requires a special configured network which supports forwarding of multicast packets. We call this a **Multicast Distribution Tree (MDT).** Only a MDT allows the simultaneous delivery of data to multiple receivers (**simulcast**).

Note that sending multiple unicast packets might significantly impact some local links, especially those close to the source. Using IP multicast just requires the source to send one packet at a time. We denote the sender as "**Source S**" and the receivers as "**Group G**". Both S and G are identified by IP numbers. Multicast packets use a class D destination address, which corresponds to the group G.

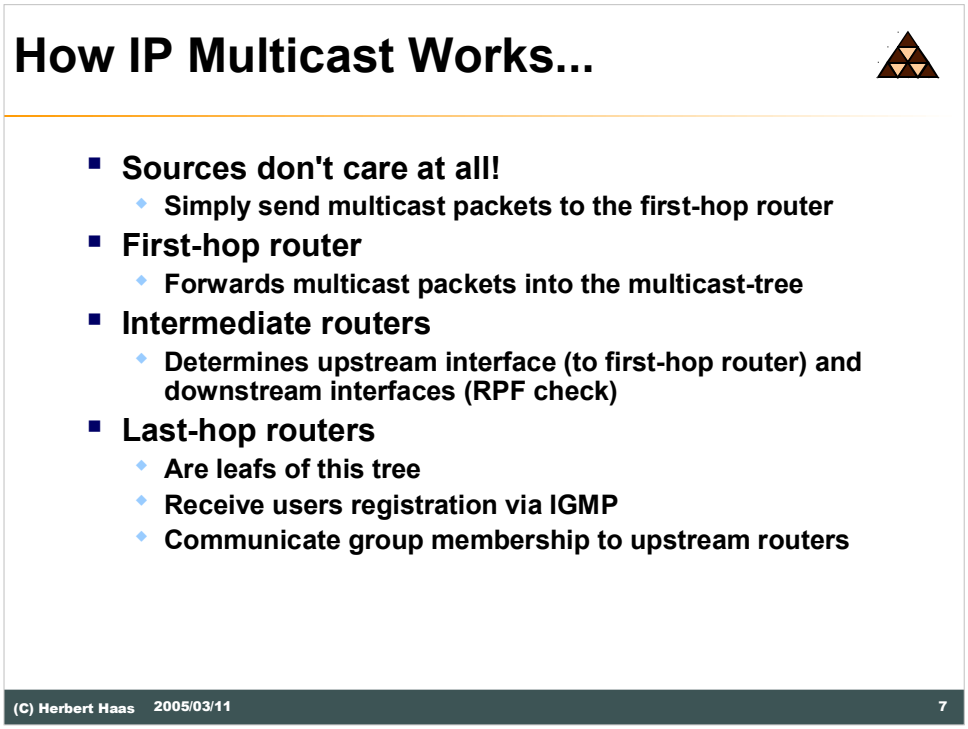
Typical applications for IP multicast: Audio and video streaming, conferencing, and other traffic distribution applications ("warehousing").



IP Multicast routing has been developed in the **late 1980s** and had a great impact on **QoS research** in the Internet. RSVP and RTP serve as helper protocol for IP Multicast, which is fully **UDP based** and therefore lacks congestion avoidance and error recovery.

All multicast methods can be classified according to their type of distribution tree. Either "**Shortest Path Trees**" (SPT) or "**Shared Trees**" are used. These are explained next.

It might be interesting to know that the first notable use of IP multicast was during the **IETF conference in 1992** where the whole conference (video and audio) had been multicasted.



RFC 1112 defines “Host Extensions for Multicast Support”. The very basic idea is that members join and leave multicast groups and the routers must manage this!

The Mbone

- World-wide multicast backbone
  - Based on tunnels
  - Playground for experiments
- Rich Mbone toolset
  - Session Directory (SDR)
  - Visual Audio Tool (VAT)
  - Robust Audio Tool (RAT)
  - Video Conferencing Tool (VIC)
  - Whiteboarding Tool (WB)

(C) Herbert Haas 2005/03/11 8

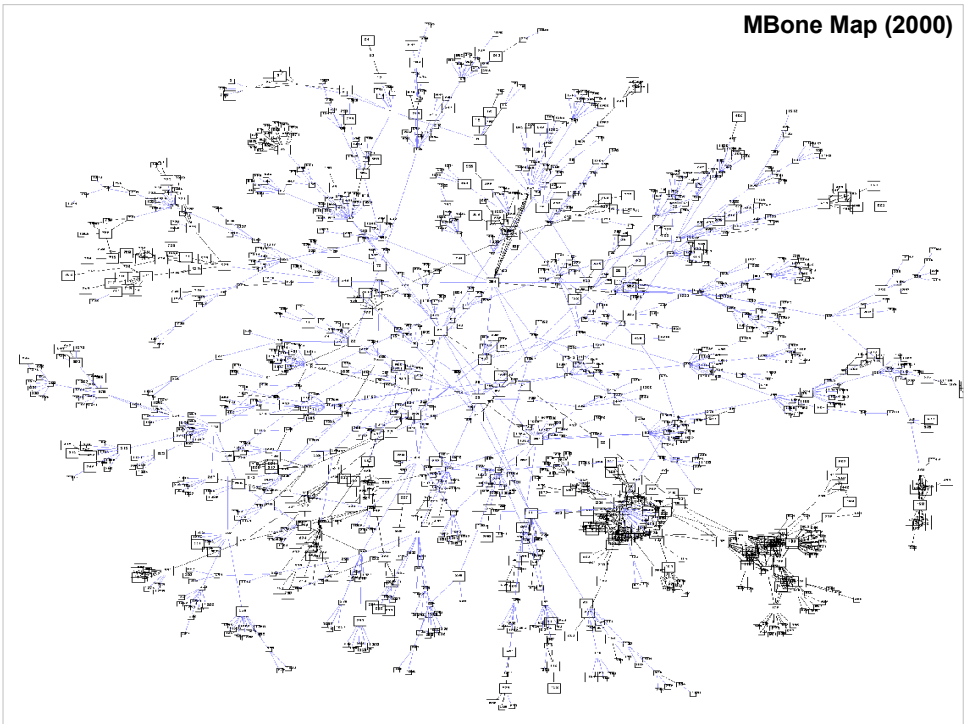
The Mbone had been developed since 1992 and had become a world-wide overlay network with dedicated multicast routers at important nodes.

The **Session Directory (SDR)** tool allows multicast group members to view advertised multicast sessions and launch appropriate multicast applications to join an existing session. SDR is based on SD (Session Directory), but they are not compatible, because SDR implements a later version of the Session Description Protocol (SDP).

The **Visual Audio Tool (VAT)** supports audio conferencing and allows multiple participants to share audio interactively. VAT is based on the RTP. VAT (and RAT) supports various codecs such as PCM, GSM, LPC4, etc.

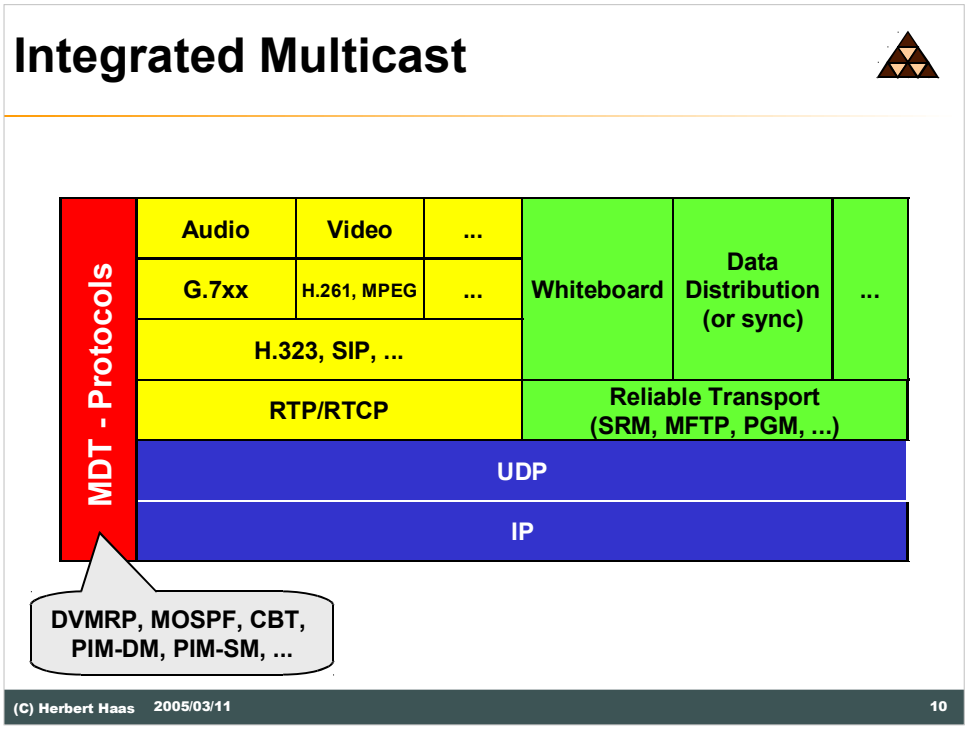
The **Video Conferencing** tool **(VIC)** allows video conferencing among multiple participants. VIC utilizes the H.261 video compression codec.

The **Whiteboarding** tool **(WB)** allows multiple participants to collaborate interactively in a text and graphical environment. Documents may be either in plain ASCII text or PostScript. WB relies on a reliable multicast protocol such as the Scalable Reliable Multicast (SRM).



This picture above shows a map of the Mbone by the year 2000. Just look at the basic structure of this network. Nodes and areas of higher and lower density can be seen. DVMRP is used between them, which is explained later on.

Mrouted is an implementation of the Distance-Vector Multicast Routing Protocol (DVMRP), an earlier version of which is specified in RFC-1075. In order to support multicasting among subnets that are separated by (unicast) routers that do not support IP multicasting, mrouted includes support for "tunnels", which are virtual point-to-point links between pairs of mrouteds located anywhere in the Internet.



The diagram above shows the basic layer structure of a fully-featured multicast infrastructure.

All data is sent over **UDP over IP** which reflects the inherent **connectionless** nature of multicast communication.

The yellow area (left half) shows **real-time applications** which need a **presentation** layer (codecs) and a **session** layer (H.323, SIP, ...) and some **real-time transport** protocols (RTP, RTCP).

The green area (right half) shows applications that demand for **reliable** data transmission and are typically non-realtime. Special protocols (SRM, MFTP, PGM, ...) are needed that provide **feedback** whether sent data has been delivered or not.

But **any** multicast environment relies on protocols that establish and maintain a **Multicast Distribution Tree (MDT).** Such protocols—often called multicast routing protocols—are for example PIM, DVMRP, MOSPF, CBT, ..., which are all explained soon. This important functionality is depicted by the red area on the left. SRM stands for Scalable Reliable Multicast but there are lots of other protocols such as MFTP and PGM... all these are explained later.

Realtime Protocols

Audio and Video

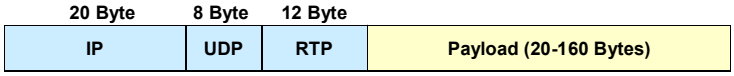
- Are typically transported by RTP/RTCP
- Feedback mechanism very important
  - For maintaining multicast distribution tree (MDT)
  - For applications to switch codecs when bandwidth becomes scarce

Practically all modern realtime protocols are sent via RTP/RTCP. RTCP provides the feedback mechanism which allows to react upon congestion problems.

If congestion occurs, most sources can change the codec. This is learned via RTCP.

Realtime Transmission

- Real Time Transport Protocol (RTP)
  - Connectionless environment
  - Payload type identification and sequence numbering
  - Time-stamping and delivery monitoring
- RTP Control Protocol (RTCP)
  - Provides feedback on current network conditions
  - Helps with lip synchronization and QoS management, etc



(C) Herbert Haas 2005/03/11 13

The **Real Time Protocol (RTP)** provides fast UDP delivery plus payload type identification and sequence numbers. Additionally a time stamp is used to verify delivery delays.

> The 16 bit **sequence number** increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. The initial value of the sequence number should be random (unpredictable) to make known-plaintext attacks on encryption more difficult, even if the source itself does not encrypt because the packets may flow through a translator that does.
>
> The 32 bit **timestamp** reflects the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived from a clock that increments monotonically and linearly in time to allow synchronization and jitter calculations. The resolution of the clock must be sufficient for the desired synchronization accuracy and for measuring packet arrival jitter (one tick per video frame is typically not sufficient).

The **RTP control protocol (RTCP)** is based on the periodic transmission of control packets to all participants in the session, using the same distribution mechanism as the data packets. The underlying protocol must provide multiplexing of the data and control packets, for example using separate port numbers with UDP.

> The primary function is to provide **feedback** on the quality of the data distribution. This is an integral part of the RTP's role as a transport protocol and is related to the flow and congestion control functions of other transport protocols. The feedback may be directly useful for control of adaptive encodings. Furthermore, RTCP carries a persistent transport-level **identifier** for an RTP source called the canonical name or CNAME. Furthermore, if all participants send RTCP packets, the **rate** must be controlled in order for RTP to scale up to a large number of participants. By having each participant send its control packets to all the others, each can independently observe the number of participants. This number is used to calculate the rate at which the packets are sent. A fourth, optional function is to convey minimal **session control information**, for example participant identification to be displayed in the user interface.

RTP Facts

- RTP does NOT provide:
  - Reliable packet delivery
  - QoS
  - Prevent out-of-order delivery
- RTP uses *mixers*
  - Special relays to combine separate video streams into one video stream
  - Also care for synchronization
  - Optionally re-encode an original stream to meet link-specific bandwidth requirements

Note that applications itself must re-sequence any packets that were sent out of order. This can be done using the timestamp

RTCP Facts

- Sent by RTP receivers
  - RTCP provides feedback for RTP senders *and other receivers!*
  - Sent to same multicast group!
- RTP sender (=multicast source) uses RTCP information to
  - Log group activity
  - Measure QoS conditions
- Other RTP receivers learn total RTCP utilization
  - Try to keep total utilization below 5% of network bandwidth

(C) Herbert Haas 2005/03/11 15

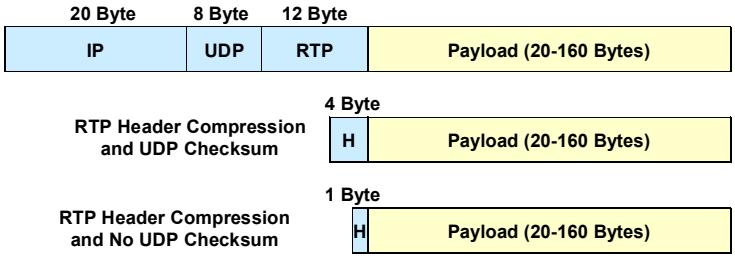
All multicast receivers periodically send RTCP control packets **to the same multicast group address** which is used for RTP delivery. This provides a feedback loop to both the sender and receivers.

Therefore, all receivers use the RTCP packets from their partners to limit the RTCP rate itself and keep the RTCP-based network utilization **below 5%** of the available bandwidth—thus making RTCP very **scalable**!

When the sender receives a RTCP packet, it may adapt to changes in the network (available bandwidth situation and congestion conditions) and keep track of the receivers.

RTP Compression

- **Simple substitution principle**
  - **Only point-to-point !**
  - **Not CPU intensive !**
  - **Might be memory greedy**

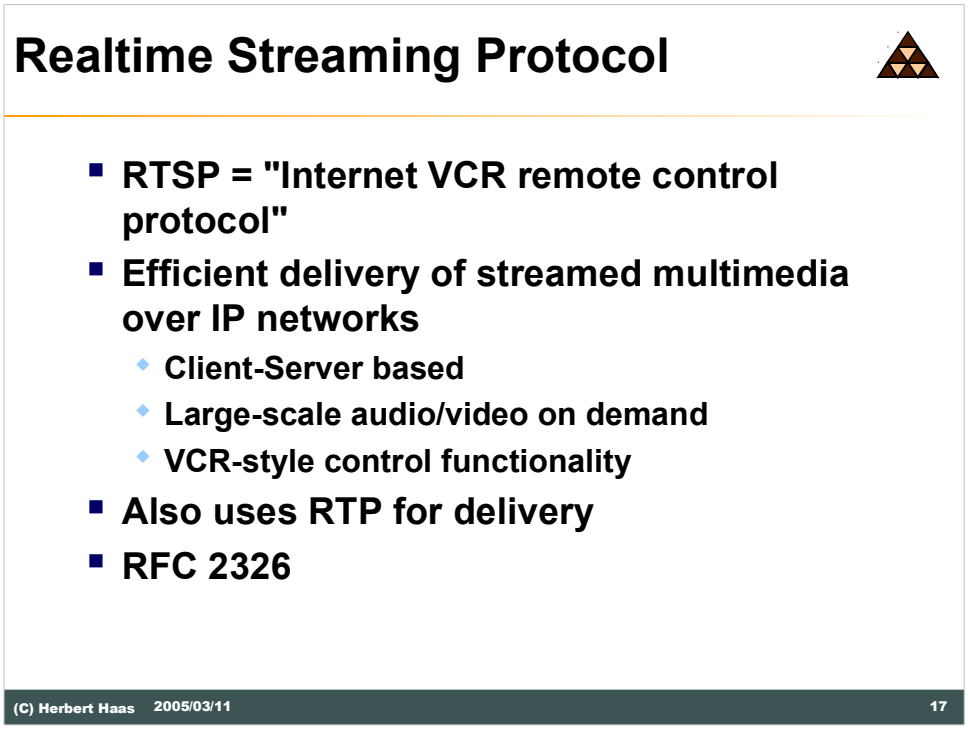


(C) Herbert Haas 2005/03/11 16

RTP compression uses a simple substitution principle. It works **only on point-to-point links** and requires the terminating devices to maintain a substitution table. Each IP+UDP+RTP header combination is replaced by a one or four byte (with UDP checksum) label.

Obviously, this compression method is **not CPU intensive** but might be **memory greedy** if a router must deal with lots of RTP connections (multicast environments or similar).

> Before Cisco IOS Release 12.0(7)T, if compression of TCP or Real-Time Transport Protocol (RTP) headers was enabled, compression was performed in the process switching path. That meant that packets traversing interfaces that had TCP or RTP header compression enabled were queued and passed up to the process to be switched. This procedure slowed down transmission of the packet, and therefore some users preferred to fast switch uncompressed TCP and RTP packets.
>
> Now, if TCP or RTP header compression is enabled, it occurs by default in the fast-switched path or the Cisco Express Forwarding-switched (CEF-switched) path, depending on which switching method is enabled on the interface. Furthermore, the number of TCP and RTP header compression connections was increased to 1000 connections each.
>
> If neither fast switching nor CEF switching is enabled, then if TCP or RTP header compression is enabled, it will occur in the process-switched path as before.
>
> Prerequisite requirements:
>
> •CEF switching or fast switching must be enabled on the interface.
>
> •HDLC, PPP, or Frame Relay encapsulation must be configured.
>
> •TCP header compression or RTP header compression or both must be enabled.
>
> TCP and RTP header compression is performed in the CEF-switched path or fast-switched path automatically. No configuration tasks are required.

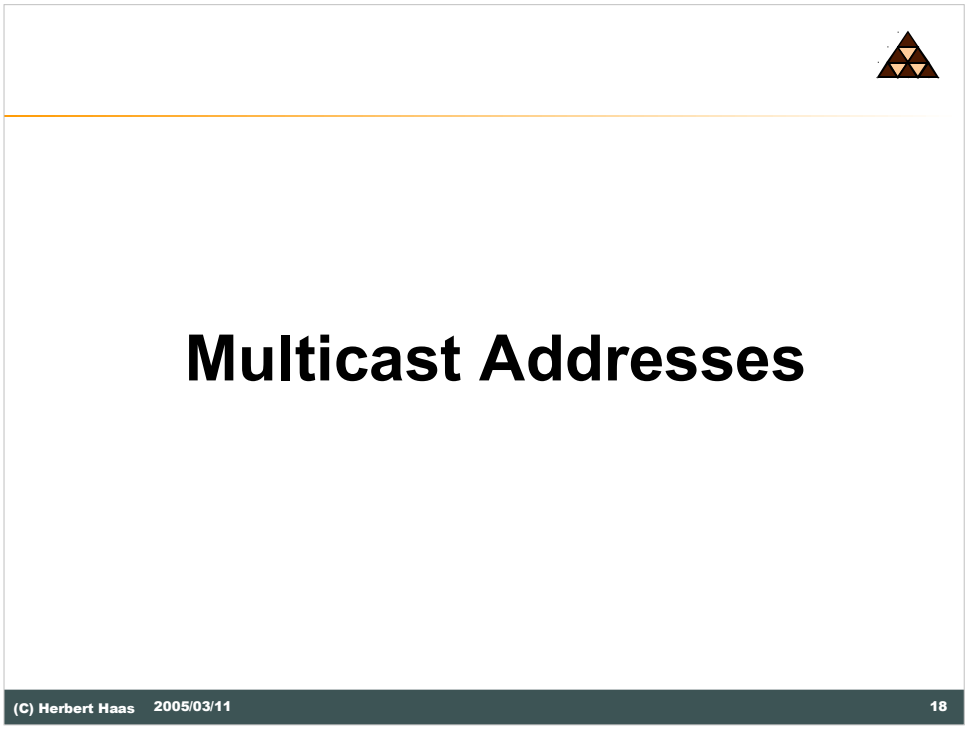


Other than Microsoft's Active Streaming Format (ASF) which is used to stream the content of a file system over a network, RTSP is client server based.

It is designed to address the needs for efficient delivery of streamed multimedia over IP networks and works well both for large audiences as well as single-viewer media-on-demand. RealNetworks, Netscape Communications and Columbia University jointly developed RTSP within the MMUSIC working group of the Internet Engineering Task Force (IETF). In April, 1998, it was published as a Proposed Standard by the IETF.

H.323 and RTSP are complementary in function. H.323 is useful for setting up audio/video conferences in moderately sized peer-to-peer groups, whereas RTSP is useful for large-scale broadcasts and audio/video-on-demand streaming. One could think of H.323 as offering services equivalent to a telephone with three-way calling, while RTSP offers services like a video store with delivery services, a VCR or cable television. RTSP provides "VCR-style" control functionality such as pause, fast forward, reverse, and absolute positioning, which is beyond the scope of H.323 and RTP.

Both H.323 and RTSP use RTP as their standard means of actually delivering the multimedia data. This data-level compatibility makes efficient gateways between the protocols possible, since only control messages need to be translated.



These section covers the IP class D address range (224.0.0.0-239.0.0.0).

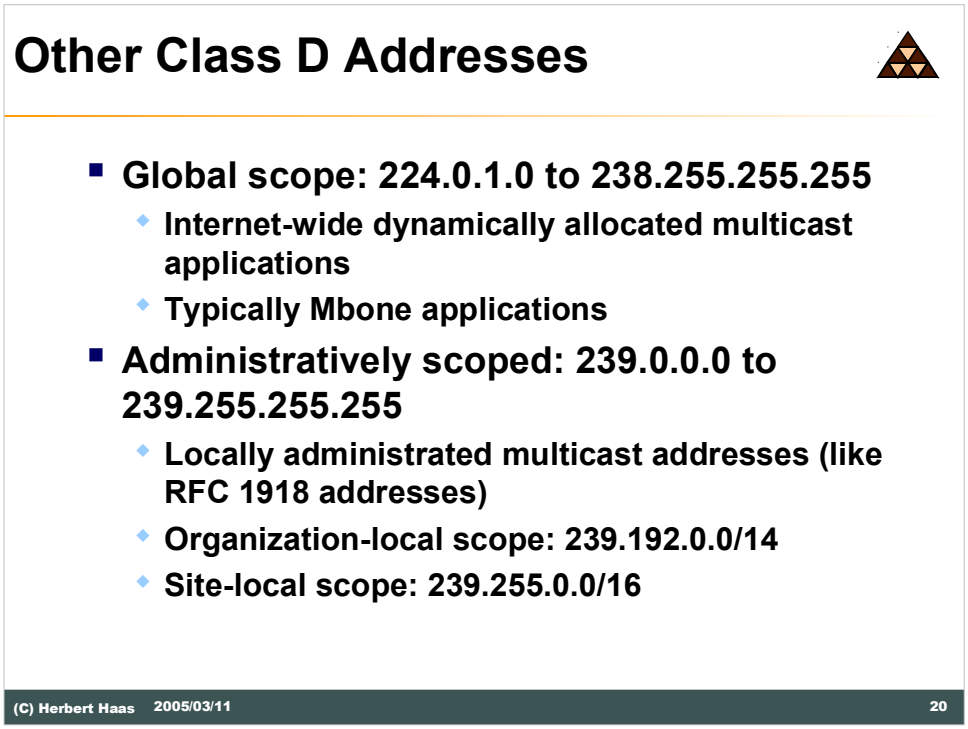
Reserved Class D Addresses

- IANA reserved range 224.0.0.0 to 224.0.0.255 to be *local scope:*
  - 224.0.0.1 = all multicast systems on subnet
  - 224.0.0.2 = all routers on subnet
  - 224.0.0.4 = all DVMRP routers
  - 224.0.0.5 = all OSPF routers
  - 224.0.0.6 = all OSPF designated routers
  - 224.0.0.9 = all RIPv2 routers
  - 224.0.0.10 = all (E)IGRP routers
  - 224.0.0.13 = all PIMv2 routers

Multicasts in this IANA-defined range are never forwarded beyond this IP-network regardless of the actual TTL value (which is typically set to 1).

"ftp://ftp.isi.edu/in-notes/iana/assignments/multicast-addresses" is the authoritative source for reserved multicast addresses.

Note that the address range 224.0.0.0 to 224.0.0.255 is regarded to be local scope. The above listing shows only some reserved "well-known" addresses from this range.

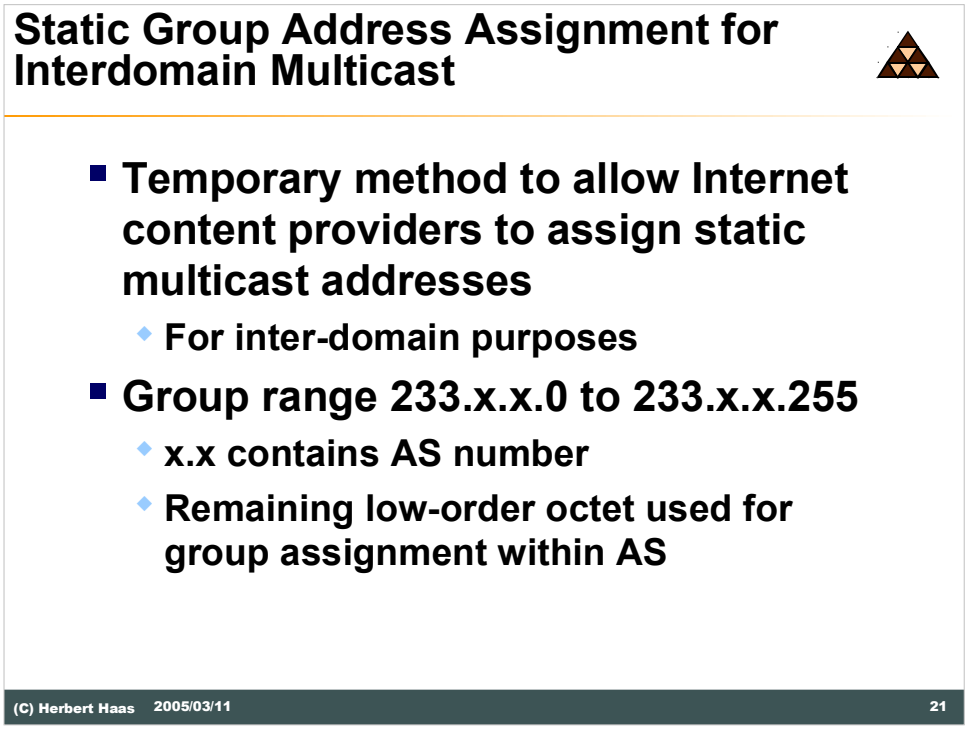


Administratively scoped multicast addresses are "private" addresses (similar to RFC 1918 unicast addresses) and must not be used within the Internet. The administratively scoped multicast address space consists of a local scope range and an organization-local scope**.**

The IPv4 **Local Scope** may grow downward from 239.255.0.0/16 into the reserved ranges 239.254.0.0/16 and 239.253.0.0/16. However, these ranges should not be utilized until the 239.255.0.0/16 space is no longer sufficient.

The IPv4 **Organization Local Scope** 239.192.0.0/14 is the space from which an organization should allocate sub-ranges when defining scopes for private use. The ranges 239.0.0.0/10, 239.64.0.0/10 and 239.128.0.0/10 are unassigned and available for expansion of this space. These ranges should be left unassigned until the 239.192.0.0/14 space is no longer sufficient. This is to allow for the possibility that future revisions of this document may define additional scopes on a scale larger than organizations.

See **RFC 2365** for further information.



One of the methods for static address allocation for multicast groups is defined in Internet standard RFC 2770 titled "GLOP Addressing in 233/8".

Until Multicast Address Set-Claim (MASC) has been fully specified and deployed, many content providers of the Internet require something at the very least to begin address allocation. This necessity is being addressed with a temporary method of static multicast address allocation.

See IETF draft "draft-ietf-mboned-glop-addressing-xx.txt" and RFC 2770.

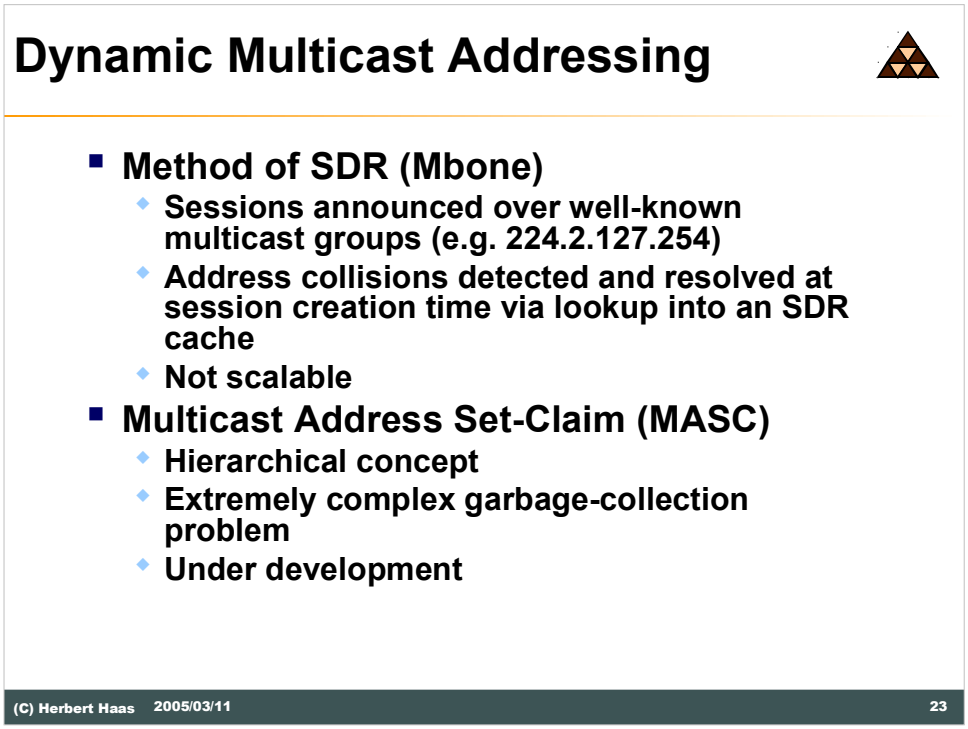
SSM Addressing

- For globally known sources and source-specific distribution trees
  - Across domains
- Group range: 232.0.0.0/8
  - 232.0.0.0 to 232.255.255.255

The increasing demand for interdomain multicast routing led to some interim solutions such as GLOP. But GLOP addressing is restricted to the last byte, which results in 255 uniquely identified groups only.

When the sources (senders) have to be globally known, a special range of multicast addresses can be used for those servers. Additionally the specialized multicast protocol called "Source Specific Multicast (SSM)" can be used, which supports building the distribution tree at the source for any group address from the range 232.0.0.1 – 232.255.255.255.

Defined in IETF draft "draft-holbrook-ssm-00.txt".



The **Session Directory (SDR)** is an important application for the Mbone. SDR detects collisions when creating new sessions and switch to an unused address. This method was sufficient in the old Mbone but today the increasing number of sessions revealed that this method does not scale well.

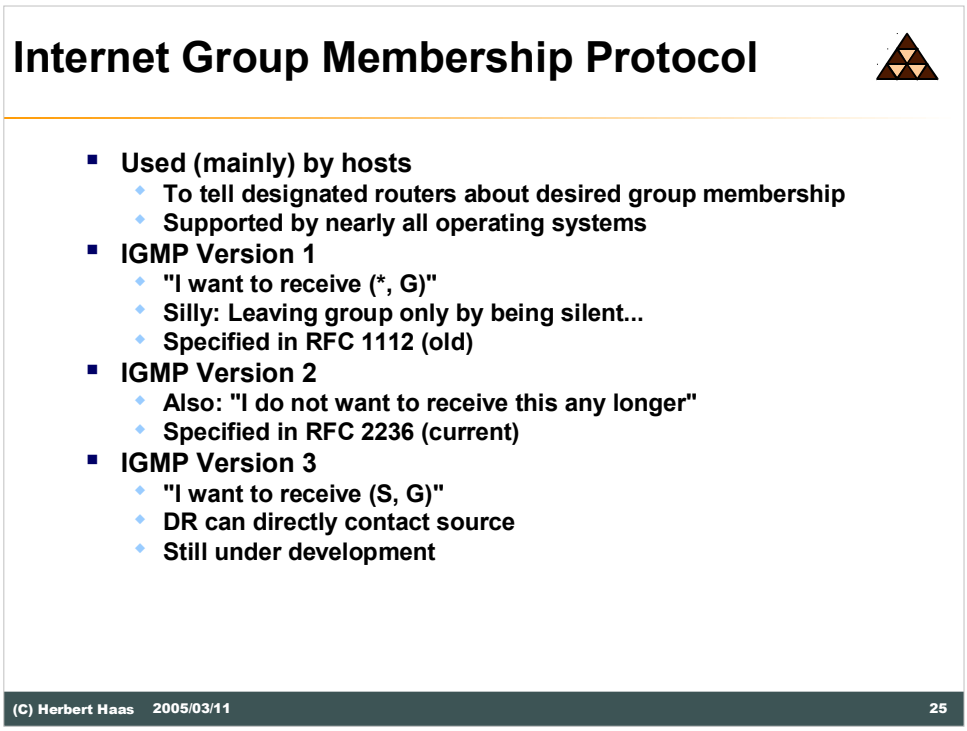
MASC is a new proposal for a dynamic multicast address allocation that is being developed by the Multicast-Address Allocation (malloc) Working Group of the Internet Engineering Task Force (IETF).

MASC requires domains to lease IP multicast group address space from their parent domain. These leases are good for only a set period. It is possible that the parent domain may grant a completely different range at lease renewal time because of the need to reclaim address space for use elsewhere in the Internet. This task is indeed very complex!

MASC is part of the hierarchical **Multicast Address Allocation Architecture (MAAA)** and represents the top level of this architecture. When a certain range of multicast addresses is allocated at the top level, the underlying hierarchies use additional protocols for address assignment. Within a domain (AS or service provider) the **Address Allocation Protocol (AAP)** is used. The **Multicast Address Dynamic Client Allocation Protocol (MADCAP)** is merely a modified DHCP and allows address assignment at leaf segments for the multicast sources. Servers for address allocation within the MAAA architecture are called **Multicast Address Allocation Servers (MAAS).**

See *"draft-ietf-malloc-masc-01.txt"* for detailed MASC principles.

IGMP



The Internet Group Management Protocol (IGMP) is primarily used by hosts to tell the DR about their desire to receive multicast traffic. Upon receiving IGMP messages the DR may retrieve the specified multicast by joining the MDT.

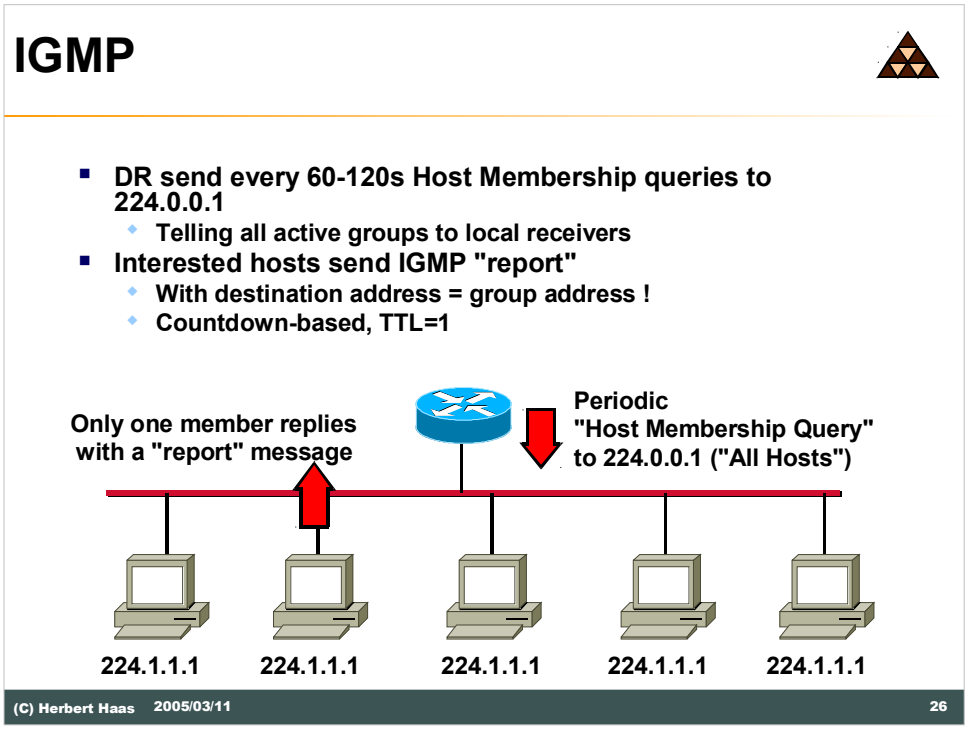
IGMP is carried directly within IP using protocol number 2.

The **initial specification for IGMP** (now considered as v1) was documented in RFC 1112 ("Host Extensions for IP Multicasting", August 1989, Stanford University). Soon several shortcomings of IGMPv1 had been discovered (e. g. hosts leave group by not responding) and this led to the development of IGMPv2.

To tell the whole truth: IGMP Version 0 had been specified in RFC 988 and obsoleted by RFC 1112.

Using **IGMPv2**, hosts can send **leave message** to the router. The router immediately sends a query in order to check if there is really no host wanting to be a member of this group. If there is no answer within three seconds (!) the group is pruned from the multicast tree. IGMPv2 was ratified in November 1997 in RFC 2236 ("Internet Group Management Protocol, Version 2" by Xerox PARC).

IGMPv3 is still under development. Please check out draft-ietf-idmr-igmp-v3-??.txt ...as things change quickly...



The basic principle is this:

The designated router sends periodically a "**Host Membership Query**" using the destination address of **224.0.0.1** ("all hosts"). Note: The **TTL is set to 1.**

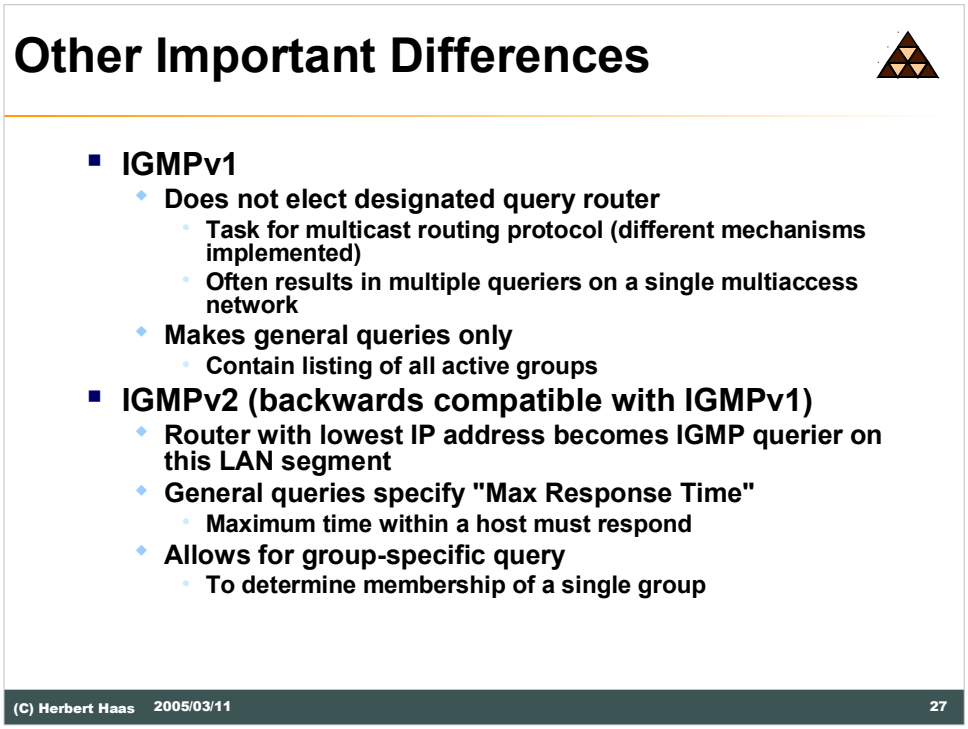
Upon receiving a "Host Membership Query" from the router each host starts a **countdown for each group** it is member of. The countdown is initialized by a random value (IGMP v1: something between 0 and 10 seconds).

Any host reaches the zero value first sends a "**Host Membership Report Message**". Again the **TTL is set to 1**. Any other host of this group can immediately cancel its countdown and does not need to reply. This method saves bandwidth and processing by the hosts.

Using **IGMPv1**, hosts leave group simply by not responding. The DR sends three query messages (one every 60 seconds) and if no host replies this subnet is pruned from the multicast tree. This is indeed silly because during **3 minutes** the whole LAN is flooded with unwanted multicast traffic.

Using **IGMPv2**, hosts can send leave message to the router. The router immediately sends a query in order to check if there is really no host wanting to be a member of this group. If there is no answer within **3 seconds** (!) the group is pruned from the multicast tree.

Note: Join messages can be also sent immediately without being queried by the DR in advance ("asynchronous joins").



IGMPv2 can do group-specific queries to query membership only in a single group instead of all groups. This is much more efficient to determine any left members of a group without asking all groups for a report. This group-specific query is not send to 224.0.0.1 but to the group's address G.

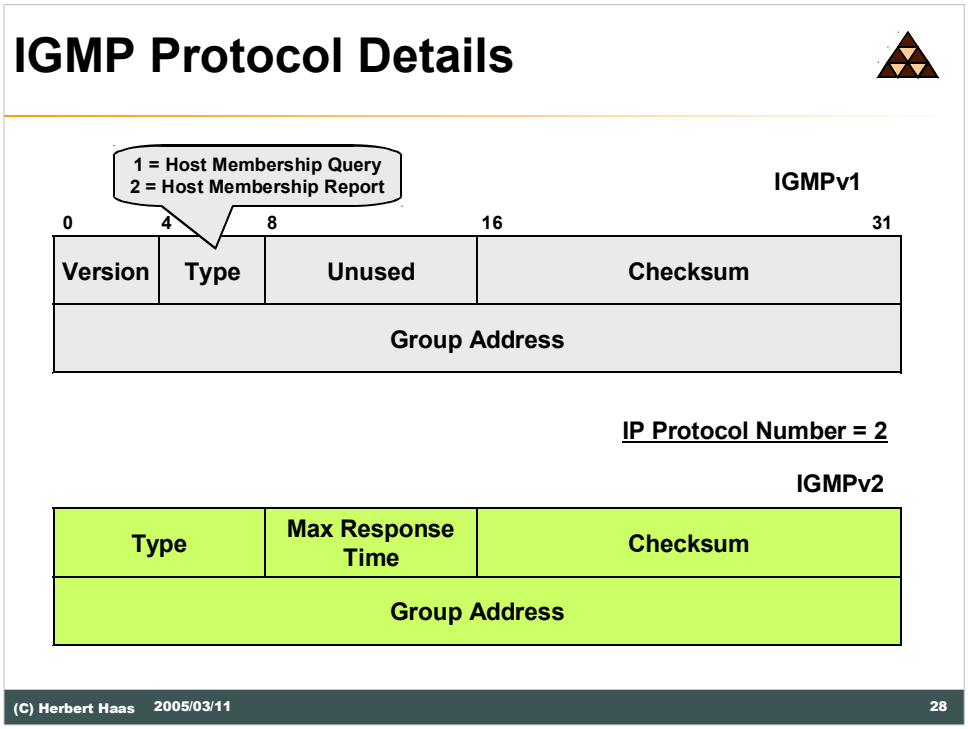
Initially all IGMPv2 routers think they are queriers but must give up immediately when a lower IP address query is noticed on the same LAN segment.

Each time a host leaves the group (by sending the IGMPv2 leave message to the group address) the designated router sends a group-specific query to check whether this was the last host leaving the group.

When **CGMP** is used in the LAN, the IGMPv2 leave message mechanism also helps the router to better manage the CGMP state in the switch.

If an **IGMPv2 host is present in an IGMPv1 environment** (including DR) this host must always send IGMPv1 reports and may suppress IGMPv2 leave messages.

If an **IGMPv1 host is present in an IGMPv2 environment** the DR must wait for the IGMPv1 timeout to be sure if this v1-host wants to enter a group because this host cannot deal with the advanced IGMPv2 query response intervals. Furthermore, the router must ignore v2 leave messages for all groups the v1-host is part of (until the 3-minute timer expires for this host).



Like ICMP, IGMP is a **integral part of IP**. It is required to be implemented by all hosts wishing to receive IP multicasts. IGMP messages are encapsulated in IP datagrams, with an IP protocol number of 2. All IGMP messages are sent with IP **TTL 1**, and contain the IP Router Alert option (RFC 2113) in their IP header. The unused field in version 1 is zeroed when sent, ignored when received.

IGMPv2 Type field:

0x11 = Membership Query. There are two sub-types of Membership Query messages: The General Query, which is used to learn which groups have members on an attached network and the Group-Specific Query, which is used to learn if a particular group has any members on an attached network. These two messages are differentiated by the Group Address.

0x12 = Version 1 Membership Report. This is defined for backwards-compatibility with IGMPv1.

0x16 = Version 2 Membership Report

0x17 = Leave Group

IGMPv2 Max Response Time field This field is meaningful only in Membership Query messages, and specifies the maximum allowed time before sending a responding report in units of 1/10 second. In all other messages, it is set to zero by the sender and ignored by receivers.

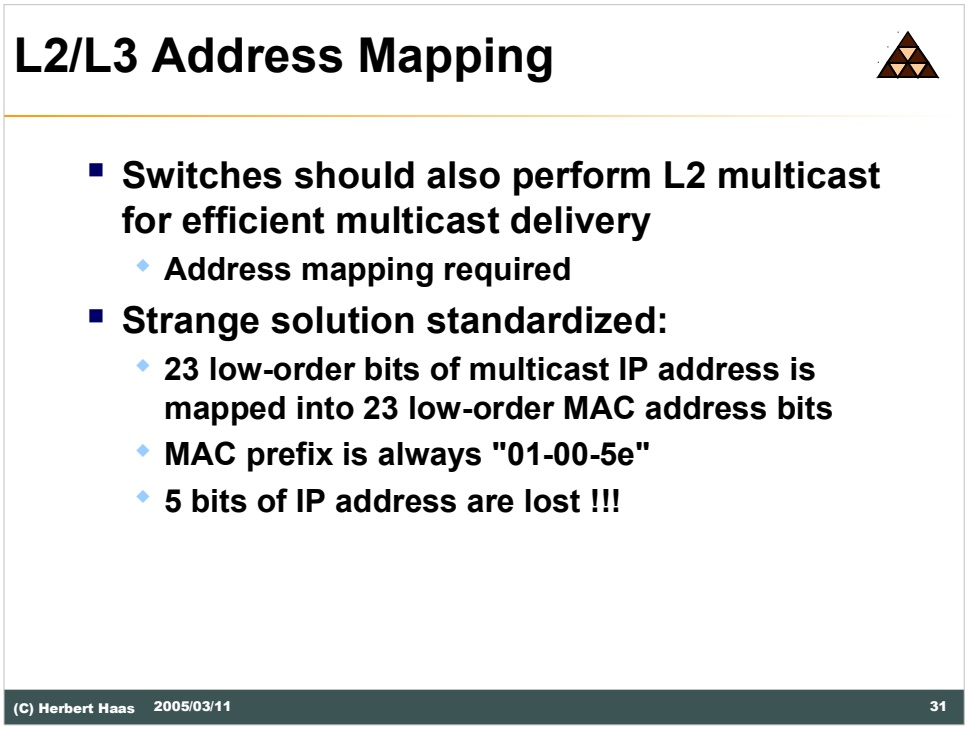
IGMPv3

- Hosts could even send a list of sources
  - Either (S, G) or [(S1, S2, ..., Sn), G]
- Advantages:
  - Leaf routers can build a source distribution tree without RPs
  - LAN switches, which would do IGMP snooping

Using IGMPv3 a host can directly say "I want to receive traffic to group G from source S". Those host could directly connect to the source!

IGMP v3lite is a Cisco specific SSM transition solution toward IGMPv3 for application developers and users that have to rely on host operating systems that do not yet support IGMPv3. Using IGMP v3lite, application developers can support IGMP v3 for SSM (Source Specific Multicast) before the host supports IGMP v3 itself in the operating system. IGMP v3lite works together with Cisco IOS routers in the network.

Layer 2 Multicast



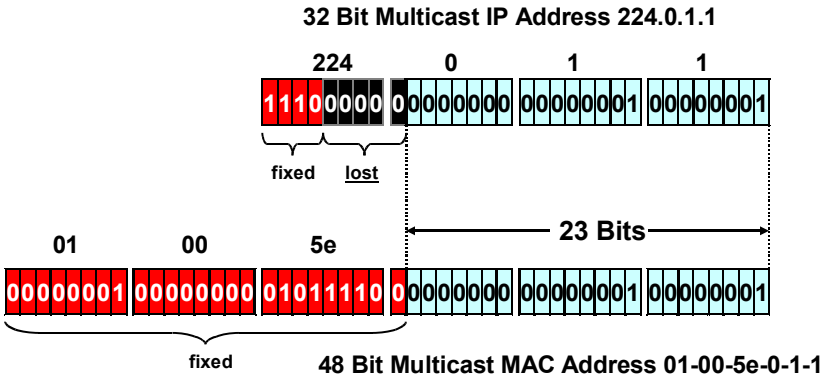
The loss of 5 address bits when mapping L3 multicast addresses to L2 MAC addresses were not originally intended. One of the inventors, Dr. Steve Deering asked for 16 OUIs to map all 28 bits of the Layer 3 IP multicast address into unique Layer 2 MAC addresses.

Unfortunately, the IEEE charged $1000 for each OUI assigned, which meant that Dr. Deering requested actually that his advisor spend $16,000 to continue his research. Because of budget constraints, the advisor agreed to purchase a single OUI for Dr. Deering. However, the advisor also chose to reserve half of the MAC addresses in this OUI for other graduate research projects and granted Dr. Deering the other half.

This action resulted in Dr. Deering having only 23 bits of MAC address space with which to map 28 bits of IP multicast addresses. It is unfortunate that it was not known then how popular IP Muticast would become. If they had anticipated such popularity, Dr. Deering might have been able to collect sufficient funds from interested parties to purchase all 16 OUIs.

Dr. Steve Deering recently joined Cisco Systems, where he is working on the development of very high-speed internet routers. Prior to that, he spent six years at Xerox's Palo Alto Research Center, engaged in research on advanced internet technologies, including multicast routing, mobile internetworking, scalable addressing, and support for multimedia applications over the Internet. He is present or past chair of numerous IETF Working Groups, inventor of IP multicast and co-founder of the Internet Multicast Backbone (the MBone), and the lead designer of the new version of the Internet Protocol, IPv6. He received his Ph.D. from Stanford University.

Address Mapping to Ethernet



- MAC prefix "01-00-5e" indicates L3-L2 mapping
- Only 23 bits had been reserved for Ethernet: 32:1 Overlapping!

(C) Herbert Haas 2005/03/11 32

After IP multicast packets have been routed to the last hop router, i. e. the router which has receivers directly attached to it, the multicast method should be continued on layer 2 if a broadcast capable medium is used. If Ethernet is used, the "**0x01005e**" prefix has been reserved for mapping L3 IPmc addresses into L2 MAC addresses.
Unfortunately, only 23 bits of the IP address can be mapped into MAC addresses, which leads to a 32:1 overlap of L3 addresses to L2 addresses.

That is several L3 addresses can map to the same L2 multicast address! This is also valid for FDDI. Token Ring addresses have other bit order, which lead to much bigger problems but this is not considered here.

For **example**, all of the following IPmc addresses map to the same L2 multicast of 01-00-5e-0a-00-01:

224.10.0.1, 225.10.0.1, 226.10.0.1, 227.10.0.1

228.10.0.1, 229.10.0.1, 230.10.0.1, 231.10.0.1

232.10.0.1, 233.10.0.1, 234.10.0.1, 235.10.0.1

236.10.0.1, 237.10.0.1, 238.10.0.1, 239.10.0.1

224.138.0.1, 225.138.0.1, 226.138.0.1, 227.138.0.1

228.138.0.1, 229.138.0.1, 230.138.0.1, 231.138.0.1

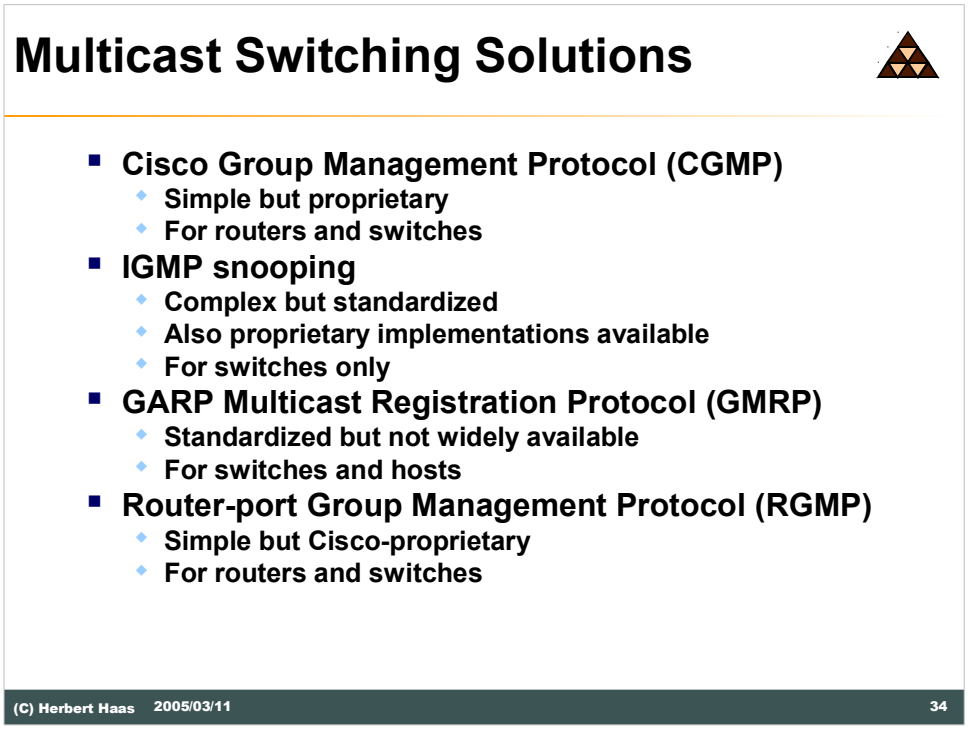
232.138.0.1, 233.138.0.1, 234.138.0.1, 235.138.0.1

236.138.0.1, 237.138.0.1, 238.138.0.1, 239.138.0.1

Multicast Switching

- Normal switches flood multicast frames through every port
  - No entries in CAM table (how to learn?)
  - Waste of LAN capacity
- Some switches allow to configure dedicated multicast ports
  - Not satisfying because users want to join and leave dynamically over any port

Some switches (including Cisco Catalysts) allow to configure dedicated multicast ports. But this solution does not scale well as users change the groups frequently.



CGMP has been created by Cisco Systems and is still a proprietary protocol which can be used by a router to tell a switch the content of IGMP messages which had been sent by hosts.

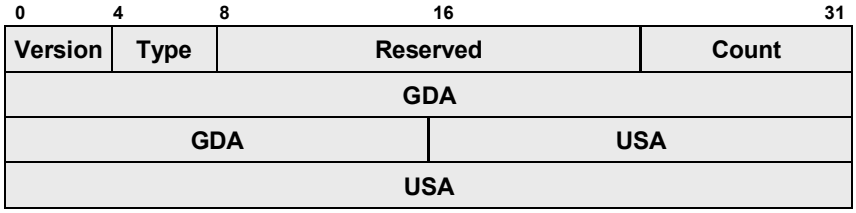
A switch can apply **IGMP snooping** and hereby intercepts IGMP messages from the host to the DR in order to learn the MAC addresses. Switches should be L3 aware otherwise the performance will degrade.

GMRP stands for Generic Attribute Registration Protocol (GARP) Multicast Registration Protocol and uses GARP to register and propagate multicast membership information in a switching domain. GARP is a Layer 2 transport mechanism which allows switches and end systems to communicate various information throughout the switching domain.

RGMP is also a Cisco proprietary solution and requires an environment where one switch is connected to routers only.

CGMP

- Sent by routers to switches
  - Destination address 0100.0cdd.dddd
- Message contains
  - Type field (join or leave)
  - MAC address of IGMP client (host)
  - Multicast MAC address of group
- Now switch can create multicast table
- Low CPU overhead



(C) Herbert Haas 2005/03/11 35

The router translates **IGMP membership** messages into **CGMP join** messages and forwards them to switches.

The CGMP messages contain the Unicast Source Address ("**USA**", the MAC address of the client) and the Group Destination Address ("**GDA**", the multicast MAC address that maps a multicast group IP address).

The switches use the CGMP information to populate the **CAM tables** with the correct multicast entries. The dedicate address **0100.0cdd.dddd** is used to address the **Network Management Processor (NMP)** inside the switches.

In the CGMP **type field** the value 0 denotes a "join" and 1 means "leave".

A "leave" message with a nonzero GDA and an all-zeros USA is used to globally delete the group in all switches. This is necessary after the last member has left the group. A "leave" message with all zeros in both the GDA and the USA fields means that all groups must be deleted in all switches. This occurs when CGMP is disabled on the router or the command **clear ip cgmp** is executed on a router interface.

Ethereal (see http://www.ethereal.com/) is a good GPL-based sniffer (or politically correct: "protocol analyzer") which is also able to decode CGMP.

Note: CGMP does not work in combination with IGMP snooping.

CGMP – Notes (HIDDEN)

- Supported by wide range of routers and switches
- Conflicts with IGMP snooping
- How to learn about all receivers in spite of the report suppression mechanism?
  - Good question...

IGMP Snooping

- Switches must decode IGMP
  - Which traffic should be forwarded to which ports?
  - Read IGMP membership reports and leave messages
  - Either by NMP (slow) or by special ASICs
- The CAM table must allow multiple port entries per MAC address!
  - Also the CPU port (e.g. 0) must be added!
  - Upon high mc-traffic load the CPU gets overloaded!
  - Special ASICs might differentiate IGMP from data traffic to improve performance

(C) Herbert Haas 2005/03/11 37

Before the first host joins a group G, there is no entry in the CAM table for this group's associated MAC address. Therefore the **first** IGMP group membership report is **flooded** to all ports including the switch CPU and the port to the DR.

Now the CPU can enter the MAC address for this group G in the CAM table together with **this host's associated port**, the **port to the DR**, and the **CPU port** (Cisco: 0). Thus, three ports are entered after the first IGMP membership message.

It is evident to include the CPU port in this CAM table entry. Otherwise, the switching engine could not forward any further IGMP message (to this group G) to the CPU. The CPU needs to see all IGMP packets (for this group G) for further IGMP snooping. Remember that normally (without multicast-capable CAM tables) any multicast would be flooded through all ports automatically. But now measures must be taken to assure that the CPU gets all packets!

If **another host** of the same group sends an IGMP membership message, the switch simply forwards this message to all listed ports for this group and adds the corresponding port to the CAM table.

Now consider a high-rate multicast traffic (e. g. 10 Mbit/s videostream) addressed to the group G. Since the CPU port is entered in the CAM table, every packet to group G is also forwarded to the CPU, which scans it for IGMP information. Clearly—at a certain point—the CPU explodes. Therefore IGMP snooping is no elegant solution for high-end switches.

If an **additional ASIC** is implemented which is able to **scan L3 information**, this ASIC could separate IGMP and bulk data packets. Then the CPU only gets IGMP packets. But it is still not elegant!

GARP Multicast Registration Protocol

- IEEE 802.1p GARP (Generic Attribute Registration Protocol) extended for IP multicast
  - Runs on hosts and switches
- Pro-active processing:
  - Hosts must also join to switch using GMRP
  - Switch configures CAM table and notifies other switches
- Incoming mc-traffic can be efficiently switched

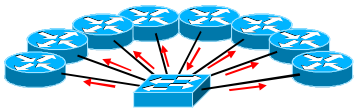
GMRP and GARP are part of the IEEE 802.1p standard and must be supported by operating systems in the hosts and switches. This standard is not yet finished.

Using GMRP a host must also send a "join" to the neighboring switch. This switch configures its CAM table (i. e. sets a filter) and exchanges membership information with neighboring switches.

Any incoming multicast traffic is simply switched according to the CAM table. There is no need to snoop for IGMP packets.

Switch/Router Problems

- Any switch connected to multiple routers must forward *all* multicast traffic to *all* routers!
  - Since routers don't send IGMP membership reports
  - Routers might get lots of unneeded packets!
- Using RGMP a router can tell a switch all multicast groups the router manages
  - Router-only switched topologies only!



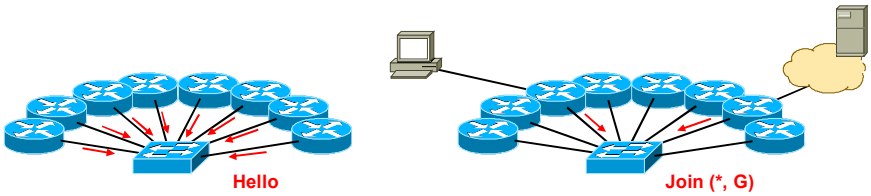
(C) Herbert Haas 2005/03/11 39

Router-Port Group Management Protocol (RGMP) is a proprietary Cisco protocol that allows to restrict multicast traffic that switches send to router ports.

RGMP may be used **only with sparse mode protocols** because they are based on an explicit join to the multicast group.

RGMP Details

- Routers periodically send hello messages to the switch
  - Switch learns about existence of routers
- Routers send RGMP (*, G) joins for groups they belong to
- Well-known address 224.0.0.25
- Restrictions:
  - Not all routers need to support RGMP
  - No directly connected sources allowed



(C) Herbert Haas 2005/03/11 40

RGMP supplements IGMP snooping in that also a router will send report messages. Using IGMP only, a switch can learn about receivers via report messages but routers usually do not send reports.

Routers can use RGMP (*, G) join and leave messages to tell the switch about multicast groups that they want to receive. After a switch receives RGMP (*, G) joins from a router, it only forwards multicast traffic for joined groups.

RGMP does not allow any directly connected sources but it supports directly connected receivers. Traffic to directly connected receivers is restricted using IGMP snooping (which must be turned on).

RGMP enabled switches will not forward any RGMP packet but non-RGMP switches will do because a 224.0.0.X (X=25) multicast address is used. Therefore it is strongly recommended to use non-RGMP switches only as leaf-switches (with an upstream link to RGMP switches).

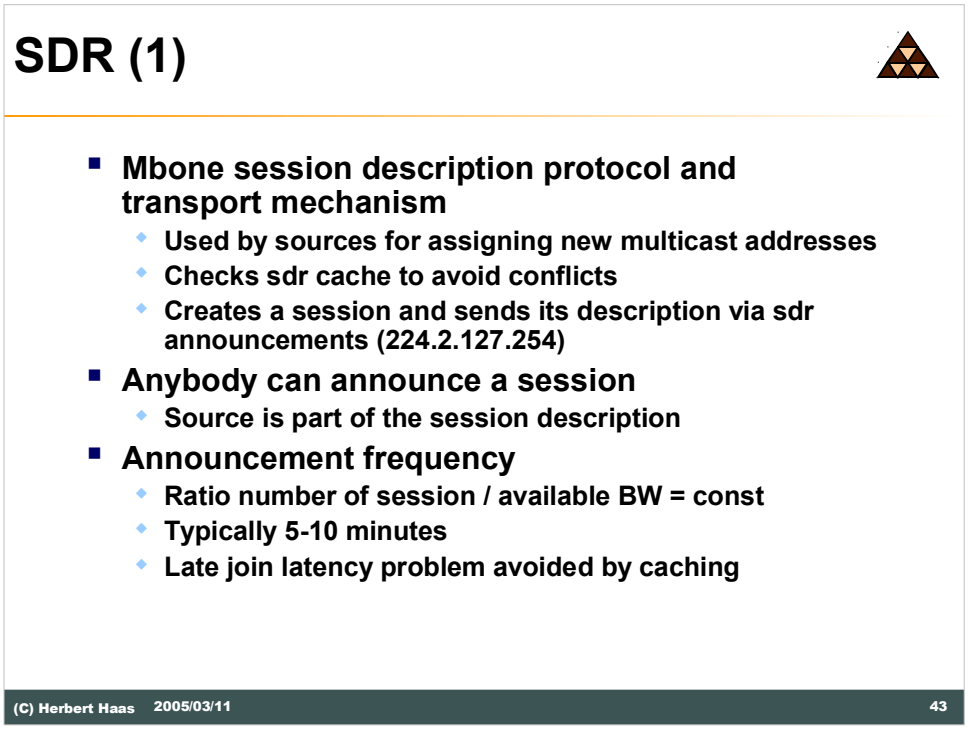
See RFC 3488 (Informational).

Session Information

Session Information

- Potential receivers must be informed about multicast sessions
  - Sessions are available before receiver launches application
  - Might be announced via well-known multicast group address
  - Or via publicly available directory services
  - Or via web-page or even E-Mail

There are several ways to tell potential receivers about multicast sessions. Either specialized protocols are used or well-defined data structures (MIME, XML) which are distributed via a web-page or E-Mail.



At the receiver side, sdr is used to learn about available groups and sessions.

At the sender side, sdr is used to create new sessions and to avoid address conflicts. During session creation the senders consult their sdr caches (note that senders are also receivers) and choose one of the unused multicast addresses. After creating the session, the senders start to announce it using all the information needed by receivers to successfully join the session, including session schedule, codecs, multicast group address and port numbers, contact information, etc.

SDR announcements are typically sent every 5-10 minutes. This **announcement frequency** depends on the number of sessions *to be announced* and on the bandwidth of the *outgoing interface* through which the announcement will be sent. Each SDR application tries to keep this ratio at a constant value.

This might lead to the **late join latency problem** for potential receivers that miss the last announcement. Newly enabled receivers must wait for the next announcement.

Caching mechanisms are used to avoid this late join latency problem. Regardless whether a multicast application is running or not, the operating system (or any other low-level multicast management process) caches all announcements locally. When a multicast application is started, it first scans this cache. Note that also inactive sessions are announced and cached.

Cisco routers can also cache the SDR information but only to create more descriptive outputs. This allows a router—for example—to use a descriptive session name instead of the multicast group address.

SDR (2)

- RFC 2327 only specifies variables but no transport mechanism
  - Session Announcement Protocol (SAP, RFC 2974)
  - Session Initiation Protocol (SIP, RFC 2543)
  - Real Time Streaming Protocol (RTSP, RFC 2326)
  - E-mail (MIME/SDR) and also web pages

The RFC 2327 only defines the standard set of variables that describe multicast sessions **but does not specify the transport mechanism** of these variables. Several transport mechanisms can be used:

The Session Announcement Protocol (SAP) can be used to carry the session info.

The Session Initiation Protocol (SIP) is primarily a signaling protocol for Internet conferencing, Internet telephony, event notification, and instant messaging.

The Real Time Streaming Protocol (RTSP) is basically a control protocol in a multimedia environment, typically used together with RTP/RTCP. RTSP provides VCR-like functions but can also carry information of a multicast session.

Finally using a special **MIME** definition, even **E-Mail** can carry session variables. Therefore, session information can also be stored on web pages because **HTTP** is also MIME aware.

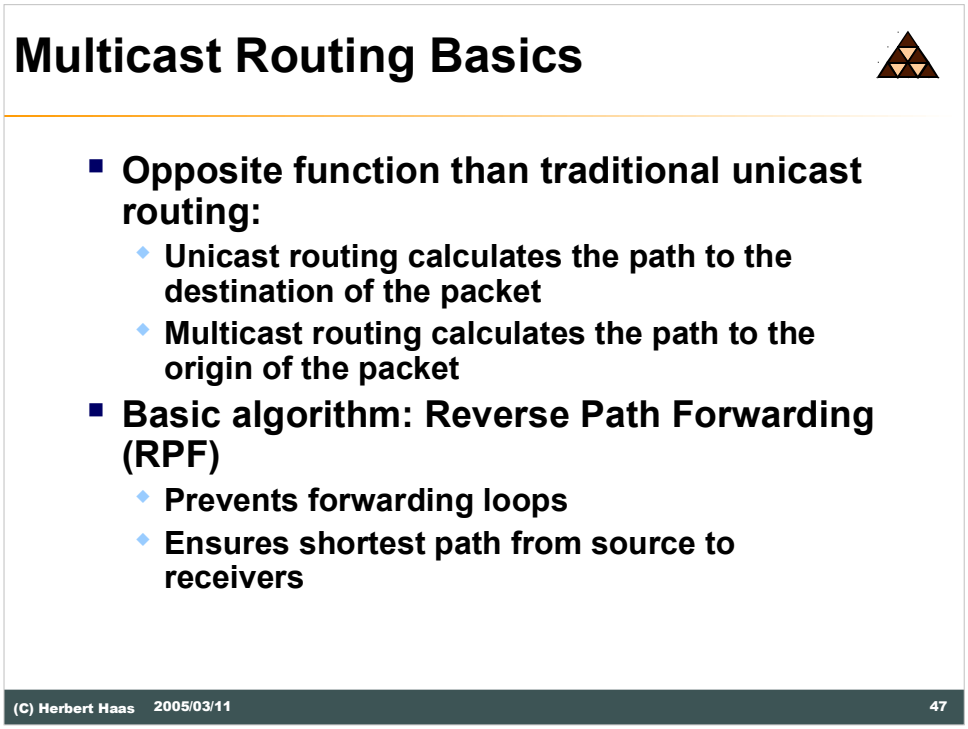
Security

- Receiver identification
  - Generally not needed except for security and feedback mechanisms (QoS)
  - Provided by RTCP
  - Applications might use unicast return messages
- Multicast flows from the sender and from receivers may be encrypted for security reasons
  - If receivers are not known to the sender, the encryption may be done only one way

(C) Herbert Haas 2005/03/11 45

When RTP is used then the co-protocol RTCP can be used to transport identification information from the receivers to the senders either via unicast or multicast. This is very important in enhanced security environments such as in a conferencing environment.

Multicast Routing Basics



The very basic message is: **Multicast routing tries to find the best interface to the source!** On the other hand, traditional unicast routing wants to determine the best interface to the destination.

The closest interface to the source is necessary in order to check whether a multicast packet arrived indeed on the upstream interface—an interface which belongs to the MDT. This check is called "Reverse Path Forwarding" (RPF), which is explained next.

In Other Words...

- **Multicast routing:**
  Which is best path to the source?
- **Prevent multicast storms: Tree!**
- **Routers do**
  "Reverse Path Forwarding" (RPF)
  - **Forwards a multicast packet only if received on the upstream interface to the source**
  - **Check source IP address in the packet against routing table to determine upstream interface**

(C) Herbert Haas 2005/03/11 48

Unicast routing is busy to determine the best path to any destination. Multicast routing works backwards, looking for the **best path to the source**. The problem with IP multicast packets is that forwarding might lead to the same problem as in bridged Ethernet LANs with redundant links: Broadcast Storms (although we should call them "Multicast Storms" here.)

But routers have on big advantage over bridges: They know the topology of the network. Hence Multicast Storms can be easily avoided by applying a simple algorithm known as "**Reverse Path Forwarding**" (**RPF**). Using RPF, each router that receives a multicast packet checks (using its routing table) whether this receiving interface is actually the closest to the source. If this is the case then this interface is an **upstream interface** and the packet can be forwarded through all other interfaces.

RPF Check

- Router forwards multicast packet only if it was received on the upstream interface to the source
  - Then this packet is already on the distribution tree
- Utilizes unicast routing table to determine the nearest interface to the source
  - RPF check fails: packet is silently discarded
  - RPF check succeeds: packet is forwarded according OIL

(C) Herbert Haas 2005/03/11 49

After performing the RPF check it is said that the RPF check **succeeds** when the multicast packet arrived on that interface specified in the unicast routing table to reach the source. Otherwise, the RPF check **fails**.

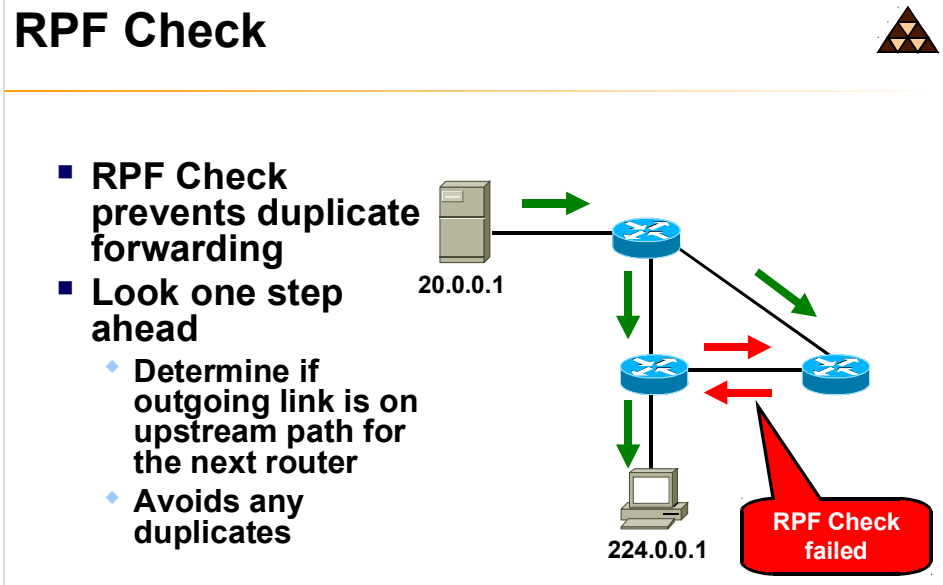
Thus, RPF ensures that multicast packets will follow the **shortest path** from the source to the receivers and ensures that there are **no loops** on that path.

The determined **RPF interface** for a specific source is used until the next RPF check is performed. Note that the unicast topology might change between RPF check events. Therefore the RPF check interval should not be too large.

On **Cisco routers** the RPF check is performed **every five seconds** by default.

Each multicast router must maintain an **Outgoing Interface List** (**OIL** or **oilist**) which contains all downstream interfaces. In the OIL each interface is associated with an **TTL threshold**.

RPF Check



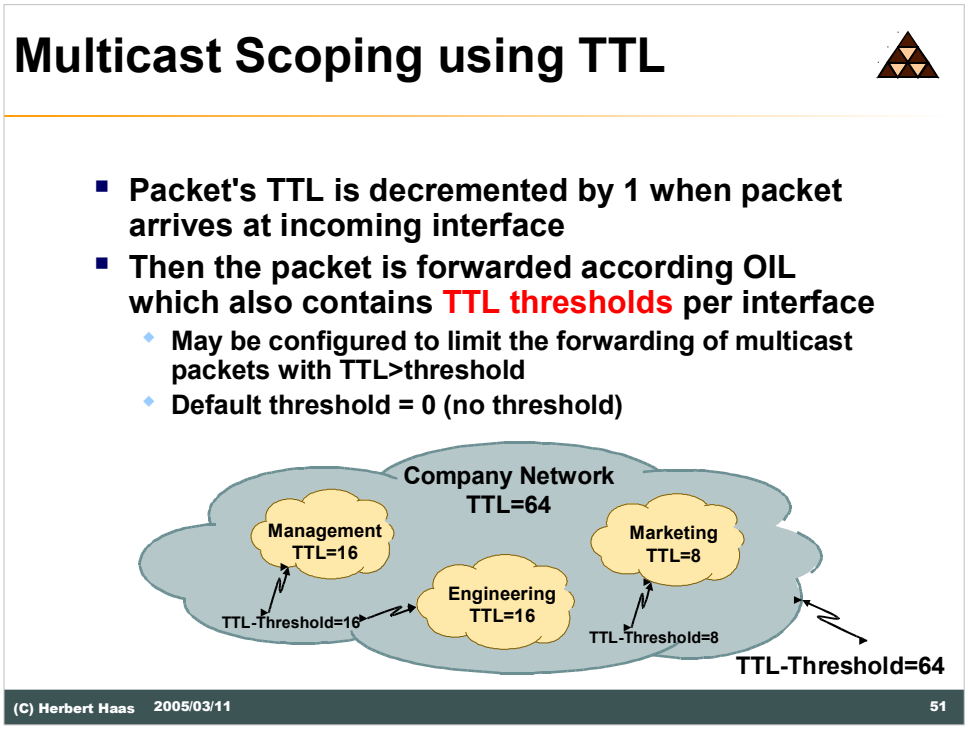
RPF can be enhanced by looking "**one step ahead**". That is, duplicate packets can be avoided if packets are only forwarded on links which are upstream to the next router. This can easily calculated using a link state protocol.

Cisco routers perform a RPF check every 5 seconds by default.

A so-called **"Outgoing Interface List" (OIL),** contains interfaces pointing to

- Multicast neighbors
- Receivers
- Administrative pre-configured interfaces

Using the OIL allows a quick decision on which interfaces the packet should be forwarded. If the OIL list is empty, then a "Prune" message is sent to the upstream router.



Setting **thresholds** on multicast routers allows to define **boundaries** for certain multicast traffic. This is for example useful to disallow external multicast traffic to enter the own network, or to prevent private multicast packets from leaving the own domain.

TTL thresholds may be set on each **interface**. A zero TTL threshold means no threshold is set. When a multicast packet arrives at a router's interface, its TTL is decremented by one. If the resulting TTL is less than or equal to zero, this packet is dropped. (This rule is exactly correct on Cisco routers. Other vendors might define other TTL handling rules.) Then the router determines the outgoing interfaces (using RPF check and OIL).

If a specific interface has a TTL threshold set unequal zero then the packet's TTL is checked against this TTL threshold. Only if the packet's TTL is **greater or equal** than the specified threshold, this packet is forwarded out of this interface.

In the **example** above there are three autonomously managed domains (management, engineering, and marketing) which have their own TTL threshold set on their respective boundaries. For example, multicast packets in the engineering domain will be originated with a TTL of 15 and cannot leave this domain. Additionally, company-wide multicast packets might be sourced using a TTL of 63.

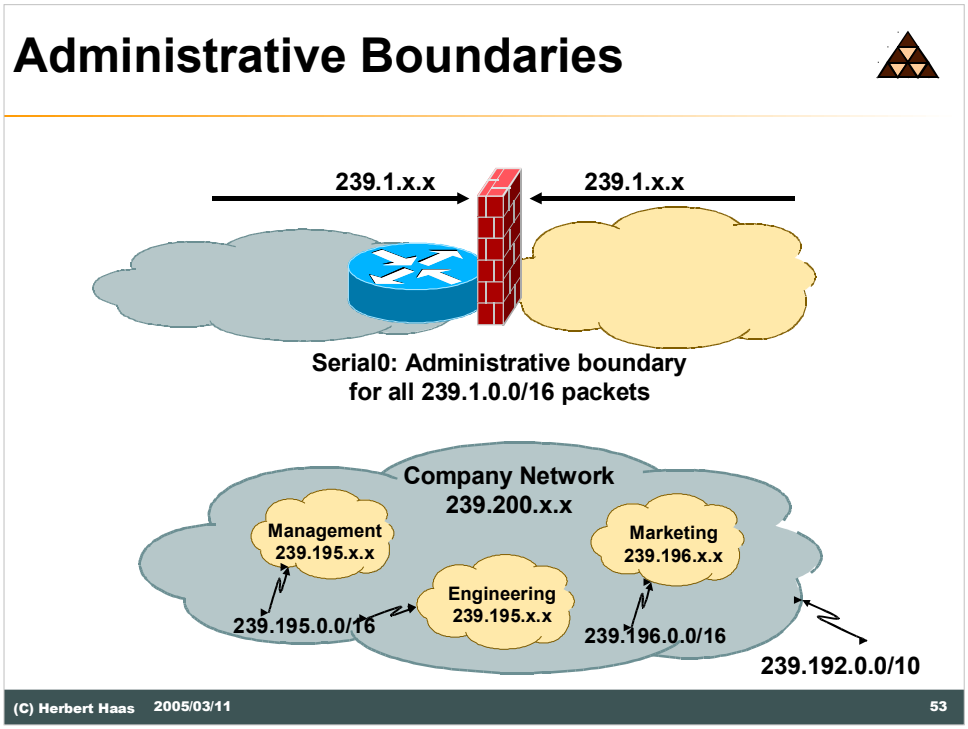
Multicast Scoping using Addresses

- Scoping via TTL thresholds relies on the TTL configurations
  - Might be unknown or unpredictable
- Administrative boundaries can be created using address scoping
  - Traffic which does not match the ACL cannot pass this interface
  - In both directions!

When **TTL scoping** is used together with broadcast and prune multicast protocols, any router discarding multicast packets cannot prune any upstream source anymore. Additionally, TTL-based multicast scoping does not support overlapping zones.

Address scoping allows to establish "administrative" multicast boundaries based on the group address. This method is much more flexible than TTL scoping. Any multicast packet that does not match an ACL—which must be specified—is dropped, no matter from which direction the packet came.

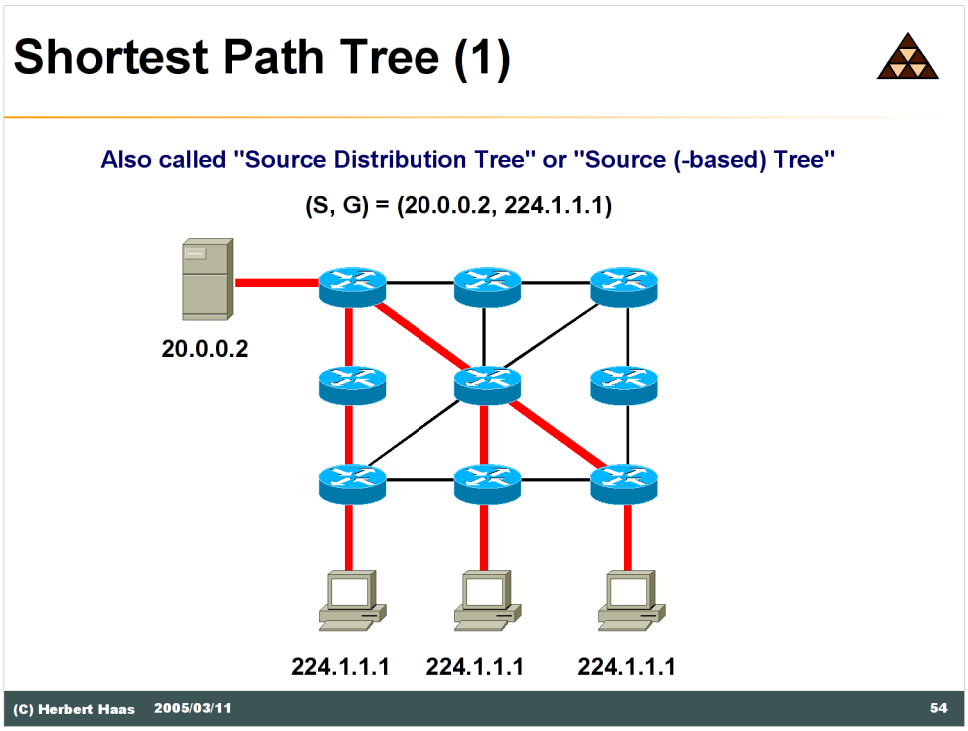
Overlapping zones are now possible to implement and requires to use different address spaces within those zones. However, this might result in a complex administration task.



As shown in the **picture at the top,** multicast packets for a specified group address (or range, as ACLs are used) cannot pass this interface **in neither direction.**

The **bottom example** shows how three administrative domains can be **multicast-isolated** from each other but it is still possible to receive the company's multicast 239.200.x.x anywhere within its boundaries.

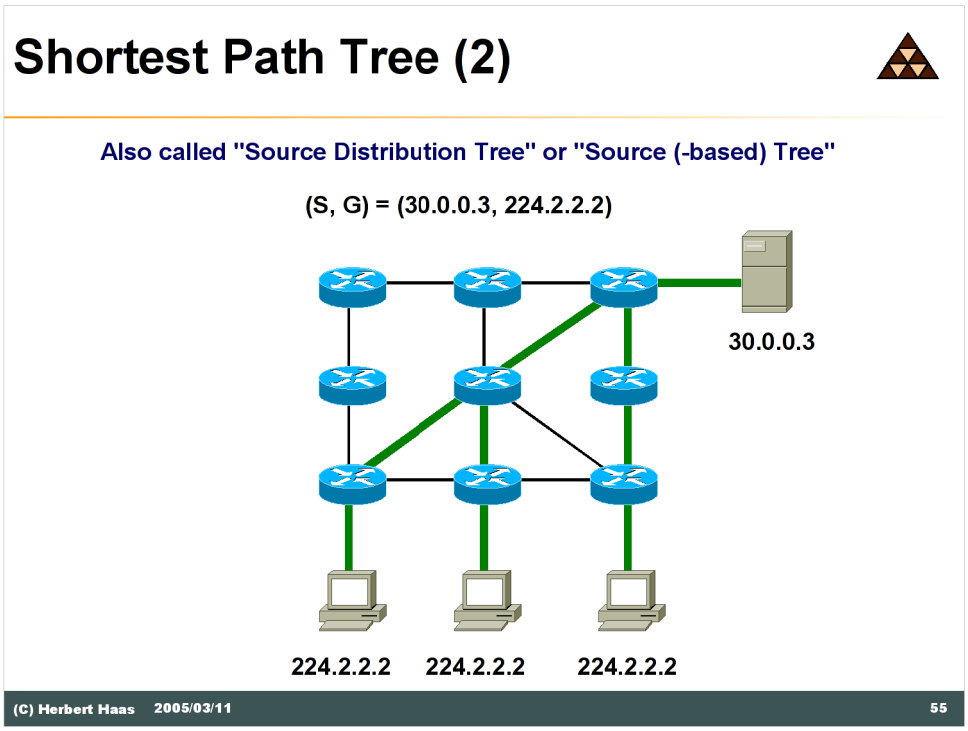
Additionally it can be seen (look at the management and engineering clouds) that several zones can use the **same address boundaries**.



"Shortest Path Trees" (SPT) are also called "Source Distribution Trees" or "Source Trees".

The basic idea is that a separate tree is created for each single source. The picture above shows only one source based tree.

This distribution method consumes much memory in the involved routers—of order O(S · G)—but it provides optimal paths from source to all receivers and minimizes delay.

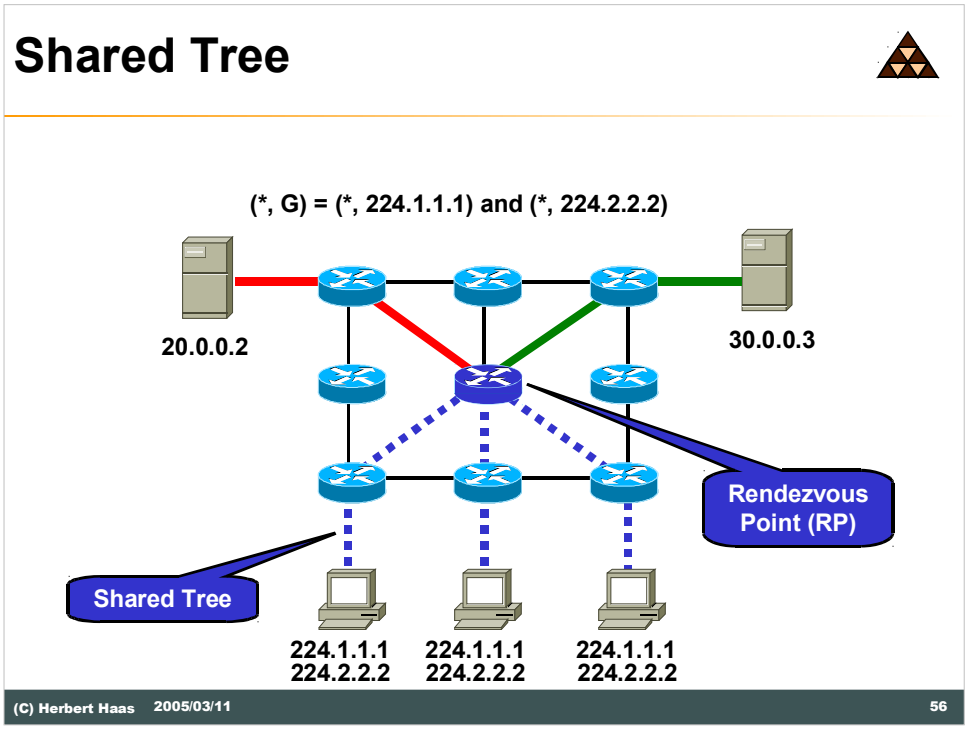


This picture shows another SPT. Each SPT is identified by a pair of addresses, that is, the unicast source address S and the multicast group address G, thus **(S, G).**

The SPT principle is the **most implemented method**, found in DVMRP, PIM-DM, and other protocols. We will discuss them later in this chapter.

Note that each router must maintain **state information for each (S, G)** combination, including timers, interface lists, etc. Suppose there are hundreds of sources and hundreds of groups...

The main point here is that a separate SPT is built for every source "S" sending to group "G". Traffic is forwarded via the shortest path from the Source!



Shared trees utilize a so-called "**Rendezvous Point**" (**RP**), which distributes multicast traffic to its attached receivers. The idea is similar as the supermarket principle: "Customers should not have to visit every manufacturer but rather buy everything at the shop around the corner."

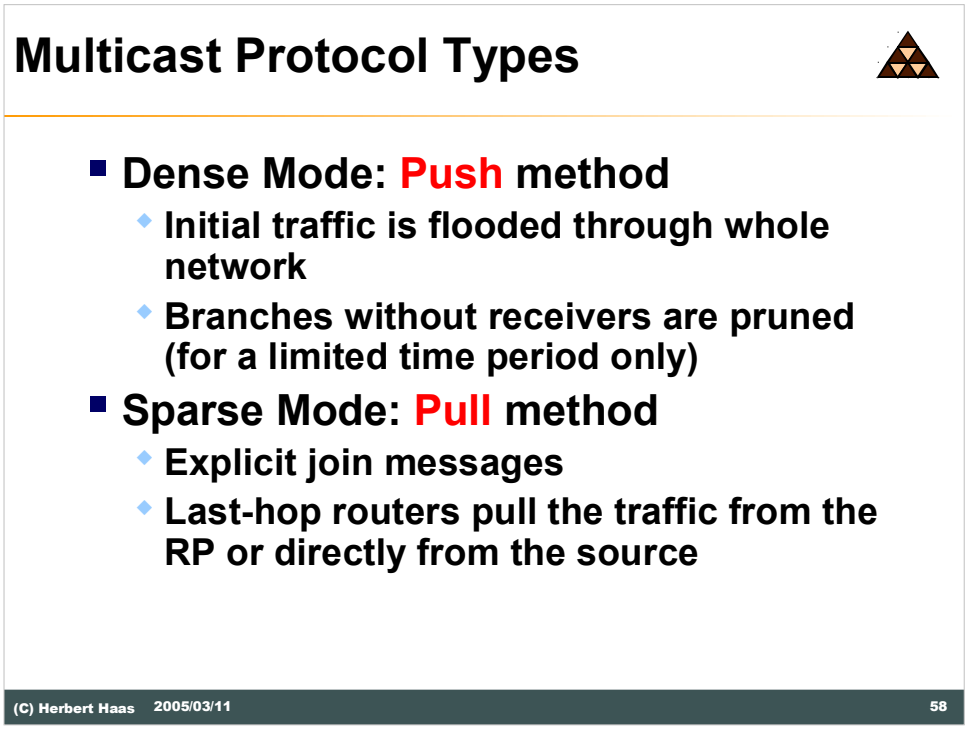
In this sense, the RP acts as supermarket and offers multicast traffic from several sources. **Typically, each RP is a leaf of a SPT**, which is rooted at a source. That is, the shared tree principle is mostly used in combination with a SPT.

Shared trees consume memory of order O(G) but might result in **sub-optimal paths** from the source to all receivers. Furthermore they may introduce extra delay. Thus, only a **clever combination** of both SPT and shared trees might be most efficient. As explained later in this chapter, this led to the development of "**PIM-SM**".

Until now, the student should have noticed, that multicast-enabled routers maintain so-called (S, G) and (*, G) entries in their mroute table.

(S,G) entries: For this particular source S sending to this particular group G, traffic is forwarded via the shortest path from the source.

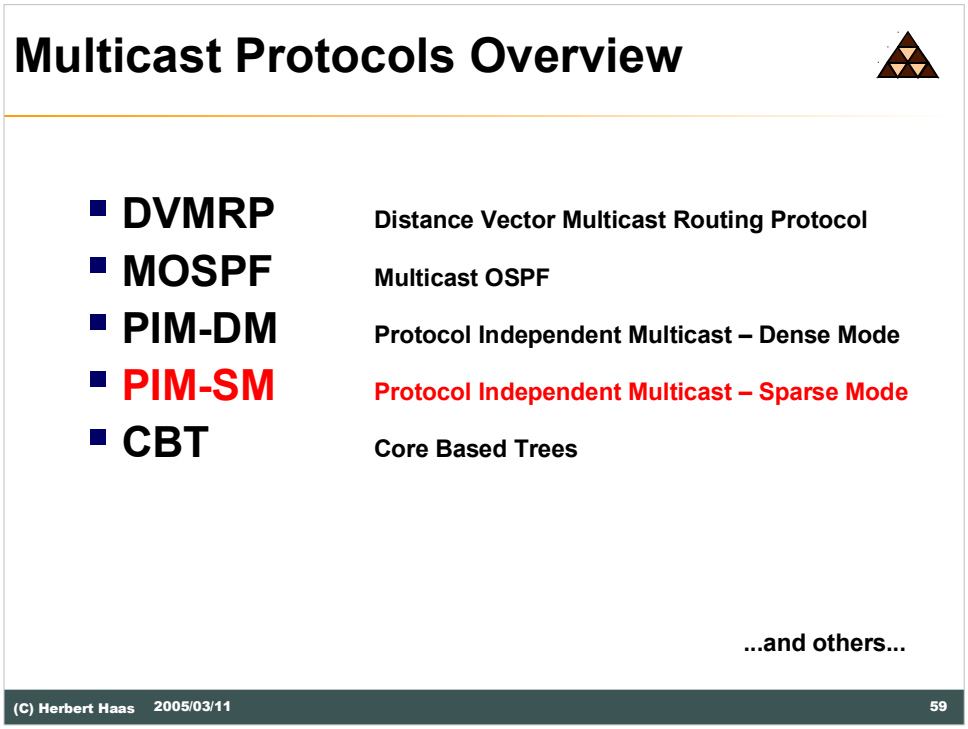
(*,G) entries: For any (*) source sending to this group G, traffic is forwarded via a meeting point for this group.



Multicast routing protocols are either **dense mode** or **sparse mode**.

The **dense mode** principle uses a "**push**" method to create the distribution tree. Multicast packets are flooded throughout the network and each router creates its OIL using the RPF check and "**prune**" messages to cut off unnecessary branches of the tree. That is, after the initial flood, branches without receivers are pruned. But after a timeout, traffic is flooded throughout the network again. Typically every 3 minutes a flood and prune occurs.

The **sparse mode** principle uses the opposite method in that routers which want to be part of the tree must send explicit "**join**" messages. Thus, the sparse mode supports a "**pull**" method for tree establishment. **Note: Branches without receivers never get any multicast traffic!**



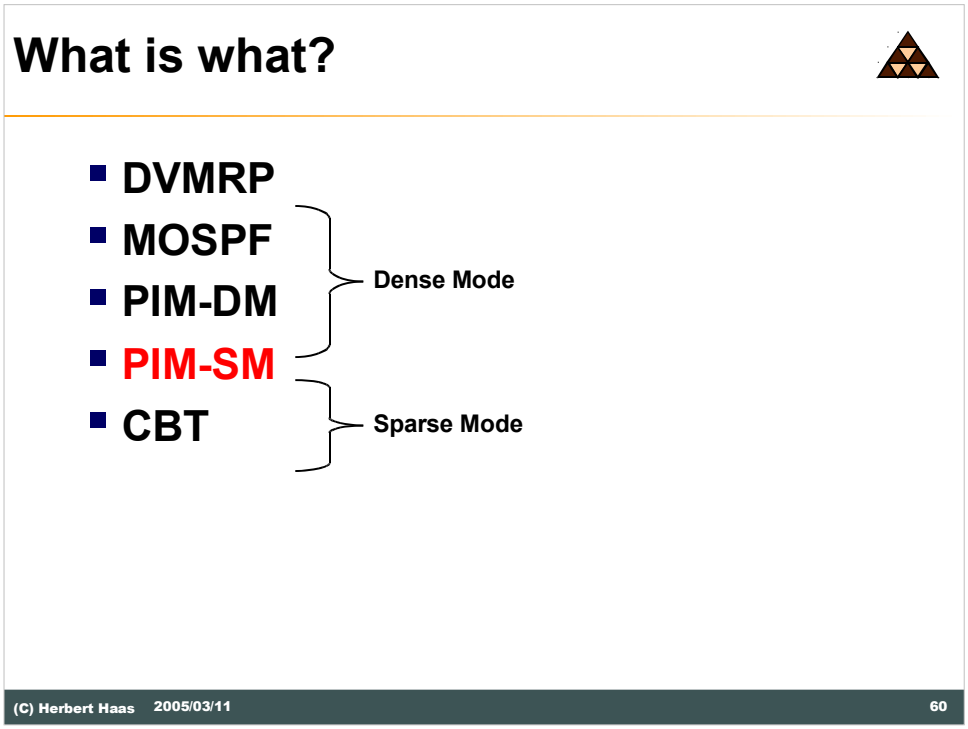
DVMRP: Version 1 (RFC 1075) was used in the early MBONE and is obsolete and unused today. DVMRPv2 is the current implementation and is used through-out the MBONE, although it is only an "Internet-Draft". Version 3 is under development.

Although **MOSPF** (RFC 1584) is the only multicast routing protocol which is found on the RFC standard track, most experts think (even John Moy) that there is more research needed. Actually MOSPF is not really implemented anywhere.

PIM-DM (Internet-draft) is useful for small networks only, as it is not really scalable.

PIM-SM (RFC 2362- v2) is the most sophisticated and useable multicast routing protocol, supporting any underlying unicast routing protocol. Cisco recommends PIM-SM for today's multicast applications.

Other proposals include CBT (RFC 2189), OCBT, QOSMIC, SM, etc., and are mostly of academic interest. New technologies might be expected for the next years.



Dense mode operation had been used for the **easiness**, with which all multicast data reach end users. Using **sparse mode** operation each user has to explicitly request needed data, e.g. if these messages are lost, the rest of the network is not influenced. But special measures have to be taken to assure the delivery of multicast data to all users.

When **interconnecting** sparse and dense mode domains there is still the problem of legacy **dense mode receivers**. No data is forwarded from the SM to the DM domain because routers in the DM domain do not have any forwarding states for any (*) sources, groups and interfaces.

Even if the receiver knows the group and wants to join, the IGMP join message is discarded by the nearest router as it does not have any state for the group.

For Cisco-based dense mode domains there is a workaround using an **`ip igmp helper-address`** command. For mrouted domains there is only the possibility by starting to send data to the group: data is flooded over the DM domain including the border router, forwarding states are created and the SM domain is learning about receivers wanting that data.

Actually this problem applies for **SDR announcements** only. All the other widely used MBONE tools run RTCP, which always sends some control messages to the multicast group address—that is you are always a sender whenever you join any group using these tools. Here the problem of DM non-pruners is completely and democratically enough solved.

DVMRP – Facts

- Dense mode protocol (Prune and Graft)
- Distance Vector announcements of networks
  - Similar to RIP but classless
  - Infinity = 32 hops
- Creates Truncated Broadcast Trees (TBTs)
  - Each source network in the DVMRP cloud produces its own TBT
  - Source Tree principle

DVMRP is quite similar to RIP but it also carries subnet masks for each network and allows for 31 hops—**32 hops** is considered "unreachable".

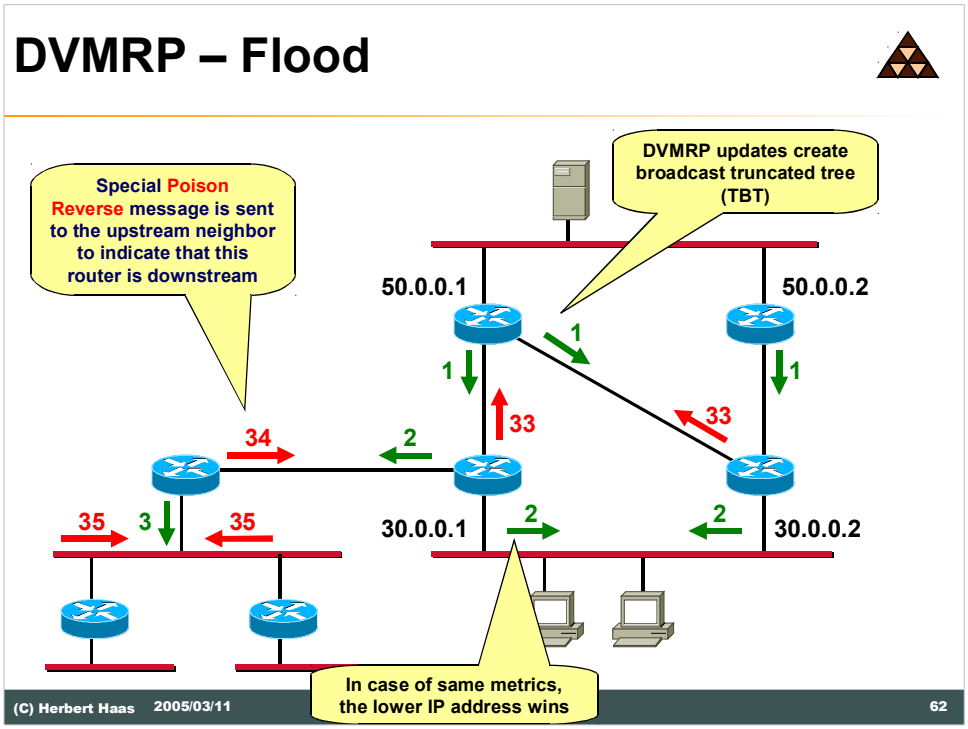
DVMRP is basically a **dense mode** protocol and therefore floods immediately the whole network with traffic, while routers create a tree using RPF. Soon, prune messages cut down the tree to a necessary size. Therefore this tree is called a **"Truncated Broadcast Tree" (TBT).**

Note that prune messages do not destroy the tree, they just stop the traffic. Prunes periodically time-out and therefore cause a reflooding of packets.

DVMRP routing information is carried **inside of IGMP** (IP protocol 2) packets. The IGMP type code for DVMRP is 0x13. That is, analyzing DVMRP packets requires sniffing of IGMP packets and further decoding. However, Ethereal can do that easily...

If two routers share the same Ethernet segment, then that router with the **lower IP address** on that segment will forward multicast traffic. This is determined through routing updates between the routers.

DVMRP is similar to PIM DM because both protocols use the broadcast and prune mechanism and an **unicast routing table** for RPF checks. DVMRP builds its own unicast routing table while PIM DM utilizes an underlying unicast routing protocol to build a multicast routing table.



This picture shows the creation of a SPT using DVMRP.

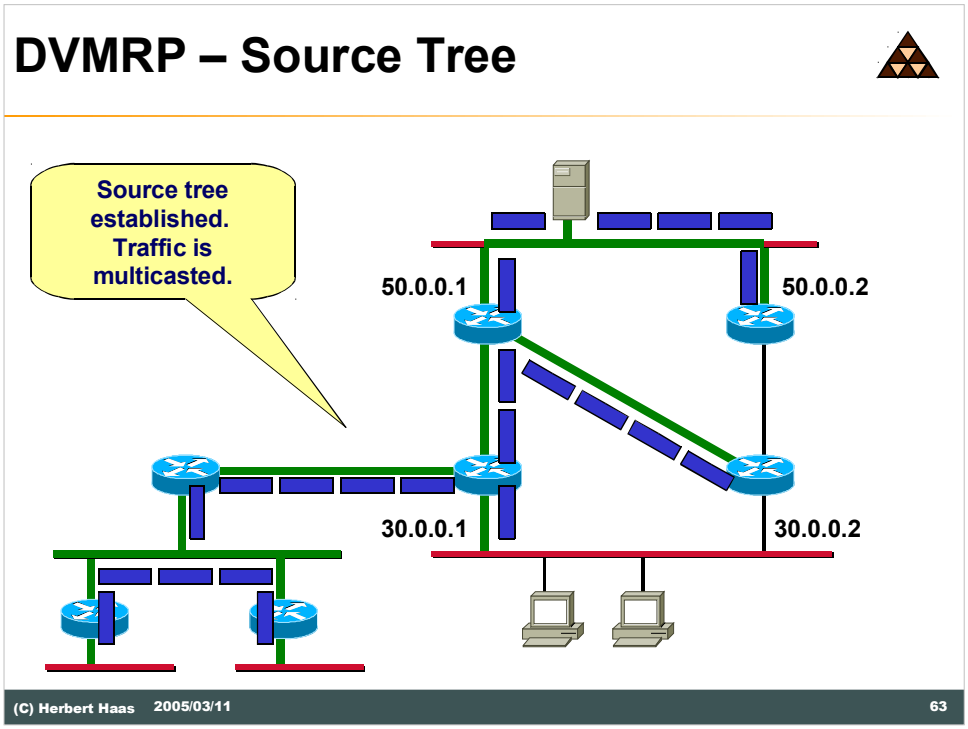
A TBT is built for each source subnet. The source subnet interface is a dedicated interface.

Each router simply announces the distance to any source network, like any other routing protocol. But note that some routers receiving such updates reply with a **special poison reverse message**, indicating, that they are indeed "downstream" in the tree.

The poison reverse message contains a distance of **32 plus the received distance** in the previous announcement of the neighbor.

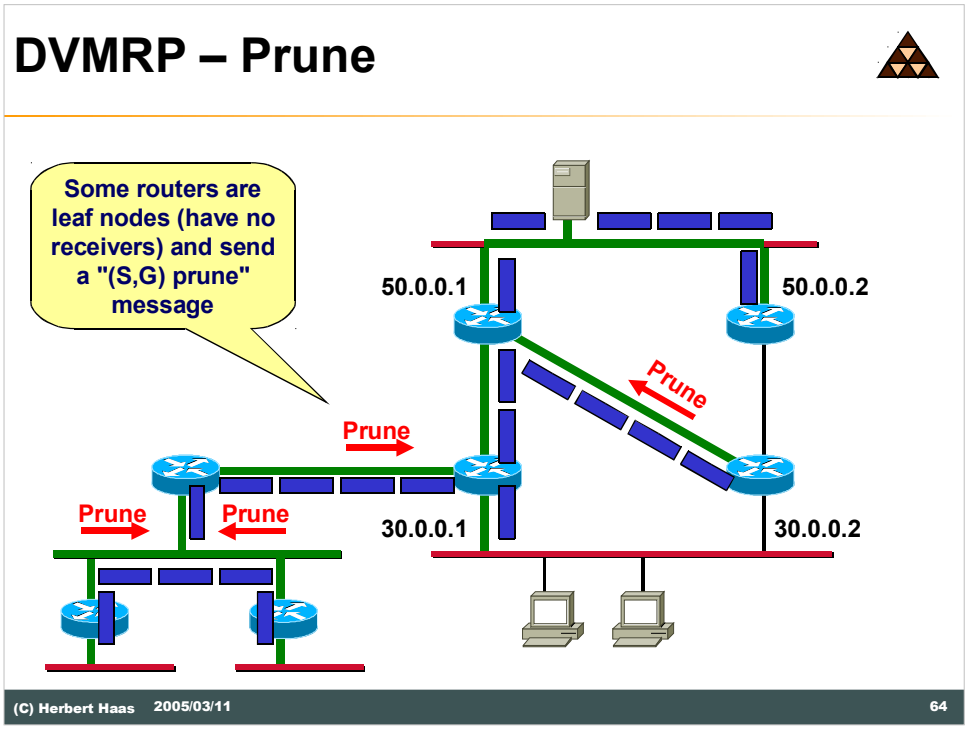
If DVMRP updates are received on two different interfaces, only the interface closer to the source is considered as "upstream". If there is more than one upstream interface the IP address of the sender (connected to the announced source network) is used as tie breaker: **the lowest address wins**.

Also, if two (or more) routers are attached to the same LAN segment and announce the same distance to each other, the router having the higher address will stop sending.



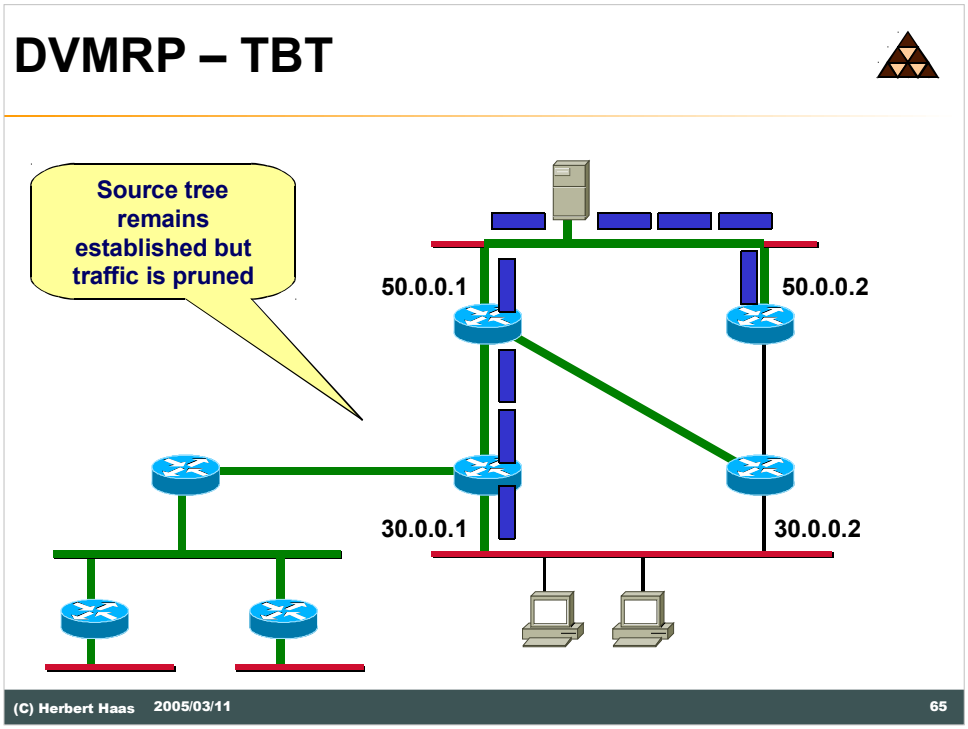
This picture shows how the multicast SPT has been established and traffic is forwarded downstream.

Side-note: The old MBONE uses currently dense mode operation, where data are flooded everywhere and each end user has to refuse explicitly to accept them (pruning). If the prune message is for any reason lost or not sent, whole network suffers from continuous, un-needed data flow.



Now, some routers notice that there are no receivers attached to them. Therefore they send a "Prune" message upstream, in order to truncate the tree.

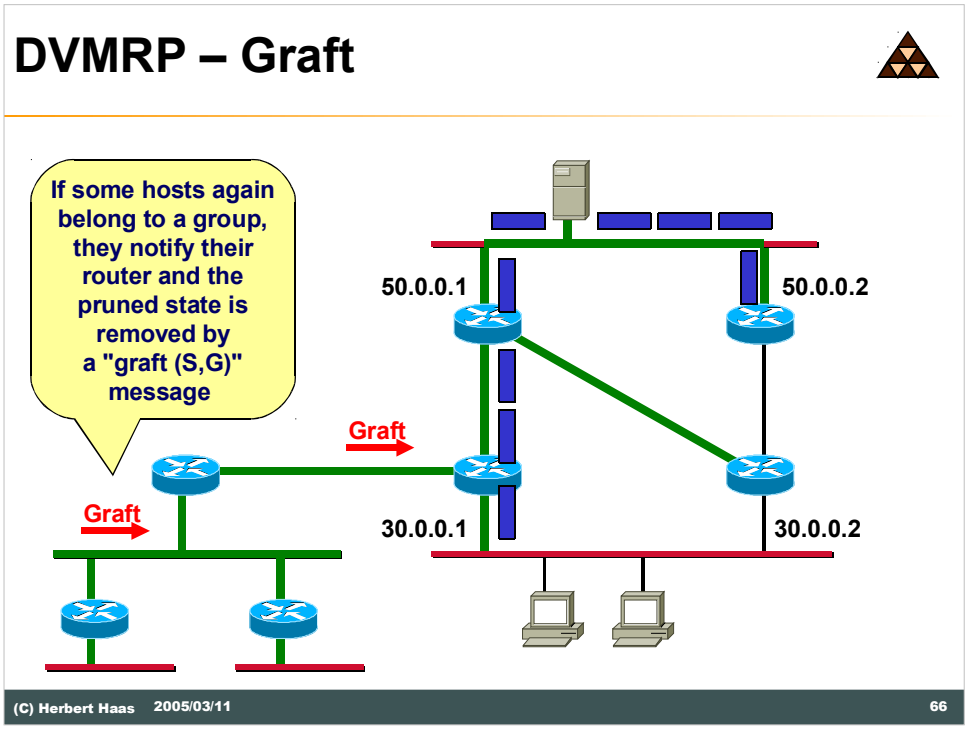
Note that a first hop router (which is directly connected to a source) will never send a prune message upstream (i. e. to the source).



After the pruning, the tree is truncated to its smallest useable size.

But note: although we call it a "Truncated Broadcast Tree" (TBT), the tree is not really destroyed! Only the forwarding of the traffic as been turned off. The tree is still there!

Each upstream router which received a prune message maintains a "state" for a certain period (3 minutes per default). After expiring, the traffic is again flooded and another prune message might be sent.



If some group appears again at a router which recently sent a prune message, then the router can remove the "state" on its upstream neighbor, by sending a "Graft" message, which specifies (S, G).

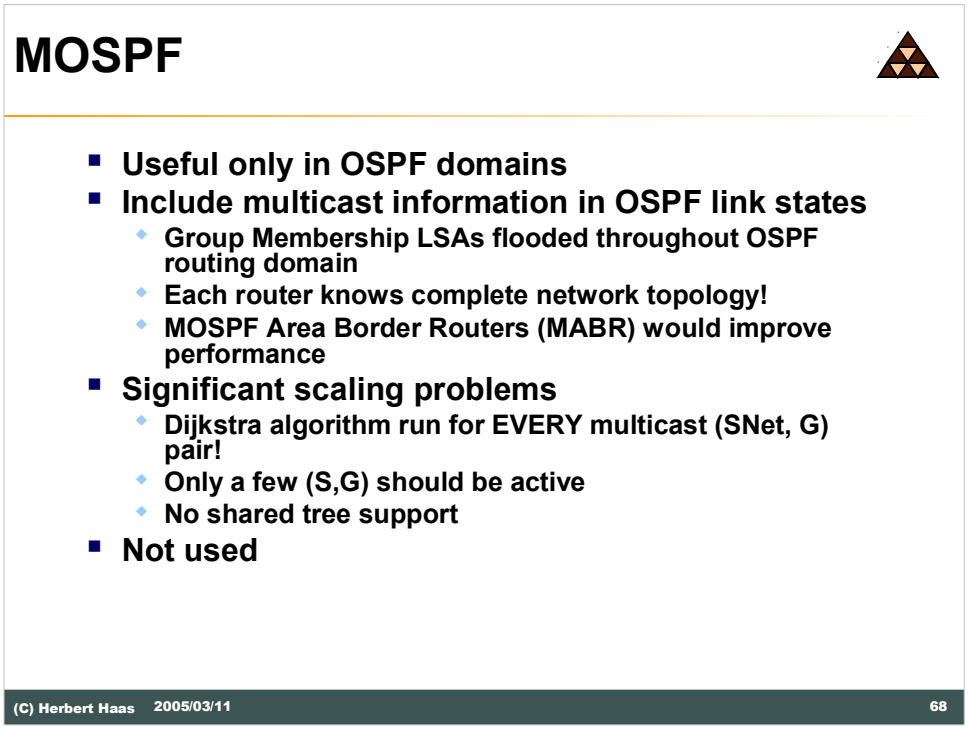
A graft message is acknowledged by the router and then forwarded to the next router. Soon the multicast traffic again flows downstream the tree.

DVMRP Facts

- **Significant scaling problems**
  - **Slow Convergence (RIP-like)**
  - **Significant amount of multicast routing state information stored in routers**
  - **No support for shared trees**
  - **Maximum number of hops < 32**
- **Used in the MBONE**
  - **Today worldwide available and accessible**
  - **Virtual network through IP tunnels**

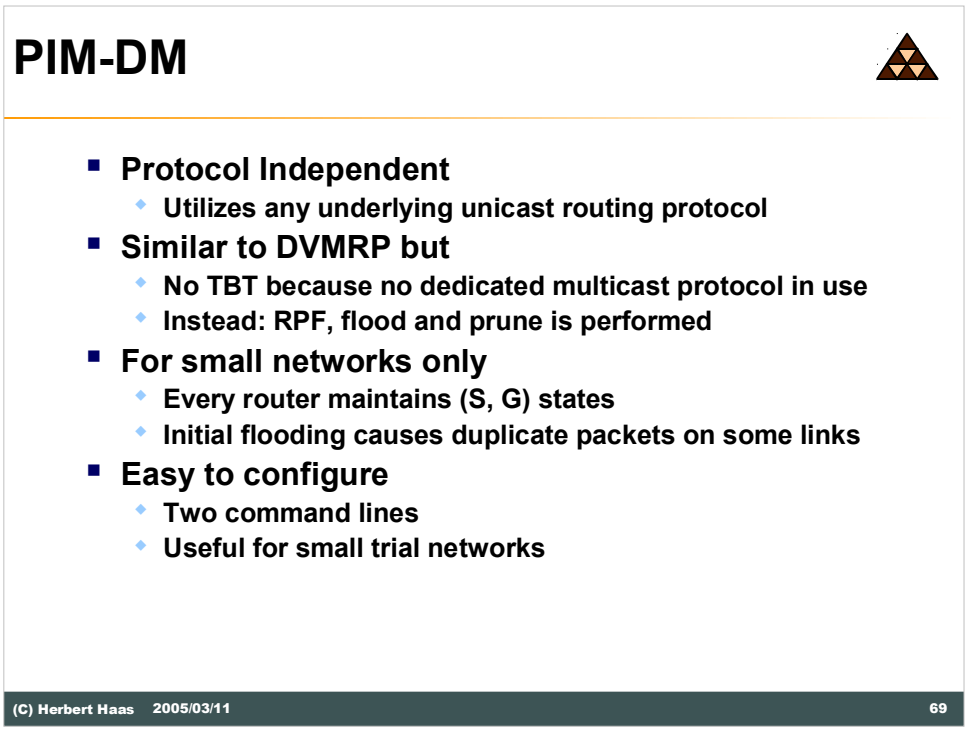
Every router has to store the (S,G) information, which is very memory demanding. Therefore DVMRP does not scale well. Furthermore, shared trees are not supported at all, and the maximum path length is limited by 32 hops. DVMRP has been used in the MBONE.

The **MBONE** (multicast backbone) is used to transmit conference proceedings and for desktop video conferencing. Multicast routing and forwarding is provided by tunnels between dedicated devices. The MBONE caused significant disruption to the Internet when popular events were active.



MOSPF denotes the Multicast Extension to OSPF and is described in RFC 1584. It only works in OSPF domains and suffers from significant scaling problems. A Dijkstra's SPF rerun is necessary on flapping links and changes of group membership.

MOSPF is not supported by any vendors today (AFAIK).



The Protocol Independent Multicast - Dense Mode (PIM-DM) supports any underlying unicast routing protocols, including static, RIP, IGRP, EIGRP, IS-IS, BGP, and OSPF.

When a PIM-DM router receives multicast traffic via its (upstream) RPF interface it forwards the multicast traffic to **all** of its PIM-DM neighbors.

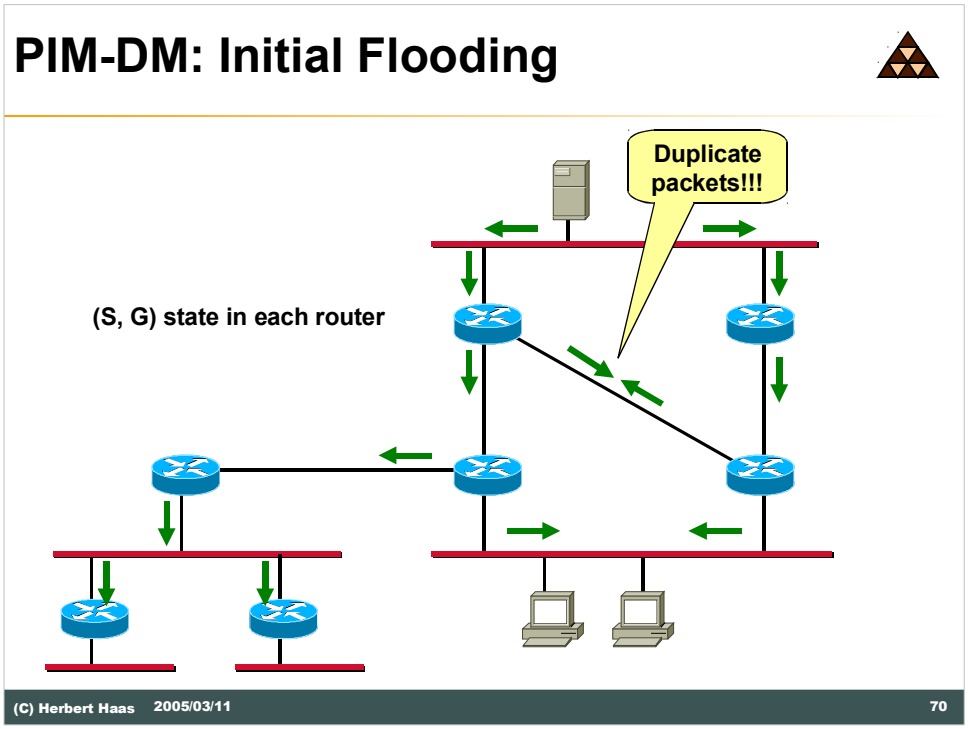
But then, the next-hop routers might receive packets also on **non-RPF interfaces!** Clearly this silly method results in **duplicate** packets on some links. These non-RPF flows are normal for the **initial flooding** of data and will be corrected by a PIM DM pruning mechanism.

Special "**assert**" messages are used to prune another routers interface. Flood and prune is performed every **3 minutes**. If the metric is equal, then the highest IP address on an interface wins.

Note: PIM-DM can be used together with DVMRP.

PIM-DM is easy to configure, there are only two commands necessary.

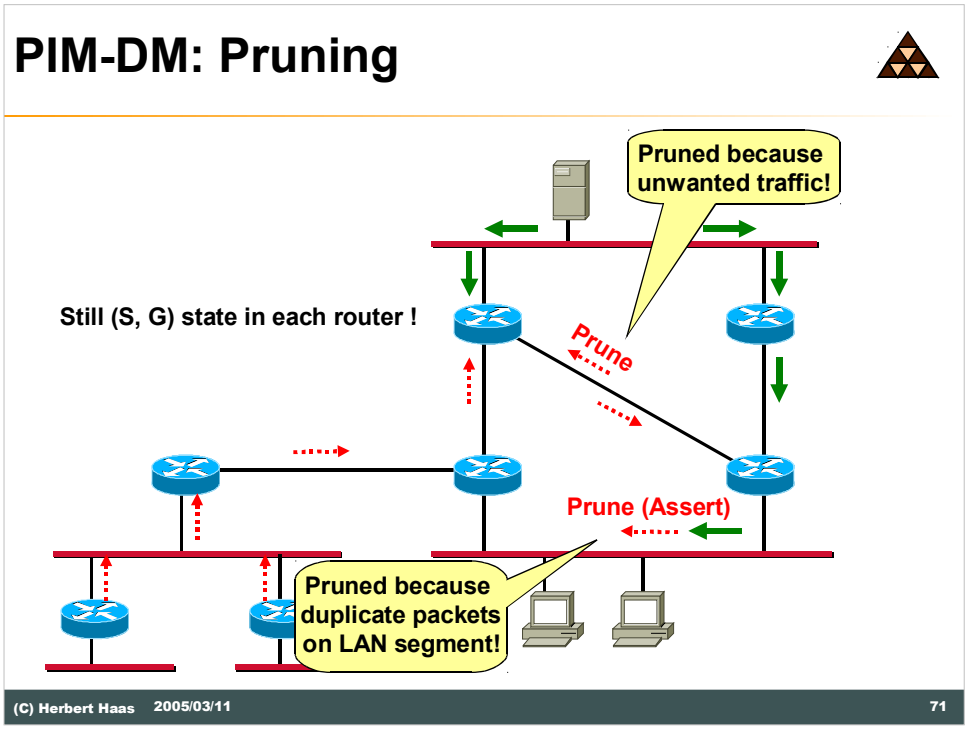
PIM-DM is an Internet draft.



The example above shows some routers which receive packets on non-RPF interfaces. The routers will discard these packets because only packets received through the upstream interface are considered as good packets.

Duplicate packets can occur on some links during the initial flooding of data and will be removed by a PIM DM pruning mechanism, following in the next step.

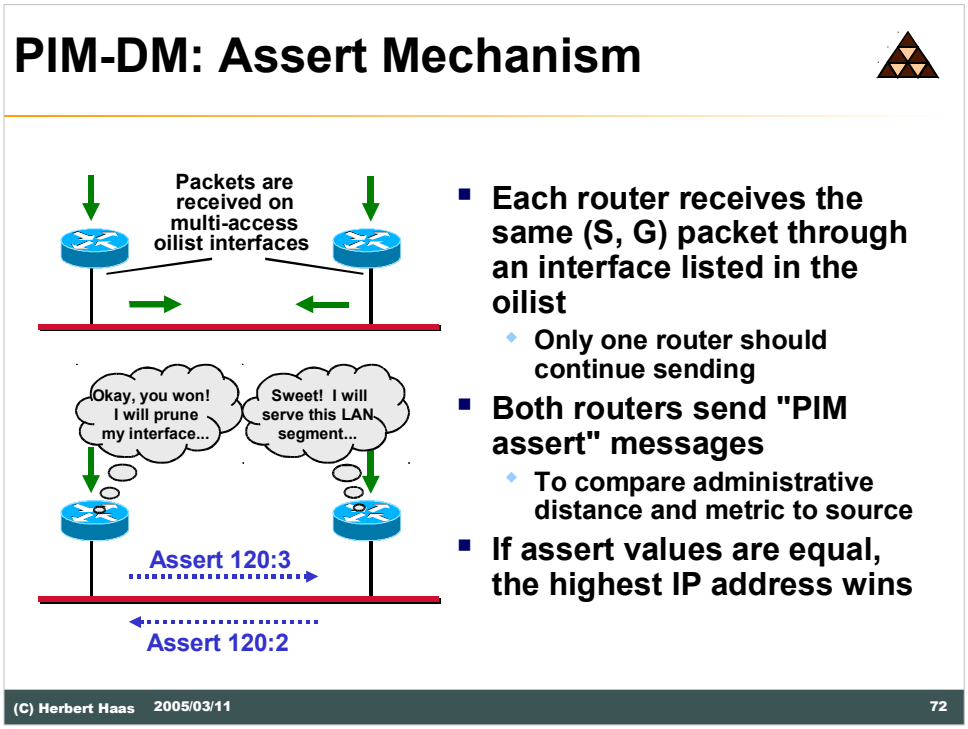
Also note, that each router must maintain a (S, G) state.



Pruning occurs after the initial flooding (which is done **every 3 minutes** by default) and serves for two purposes: First, branches can be cut off when there are no further receivers downstream; secondly, a router can use a so-called **assert** message to stop another router from sending packets to its non-upstream (i. e. non-RPF-) interface. The latter method forces the other router to prune its own interface.

Again: The prune state lasts three minutes by default. Then a new flooding occurs over all links!

Note, that each router must **still** maintain the (S, G) state.



The PIM assert mechanism is used to eliminate duplicate flows on the same multi-access segment. Other than DVMRP (which establishes a TBT in advance using a dedicated multicast routing protocol), the assert mechanism is only performed when duplicate packets appear on this link.

When a router receives a (S, G) packet via a **multi-access interface** which is listed in the (S, G) **oilist**, then it will send an **assert message**, telling the other router a so-called **assert value**.

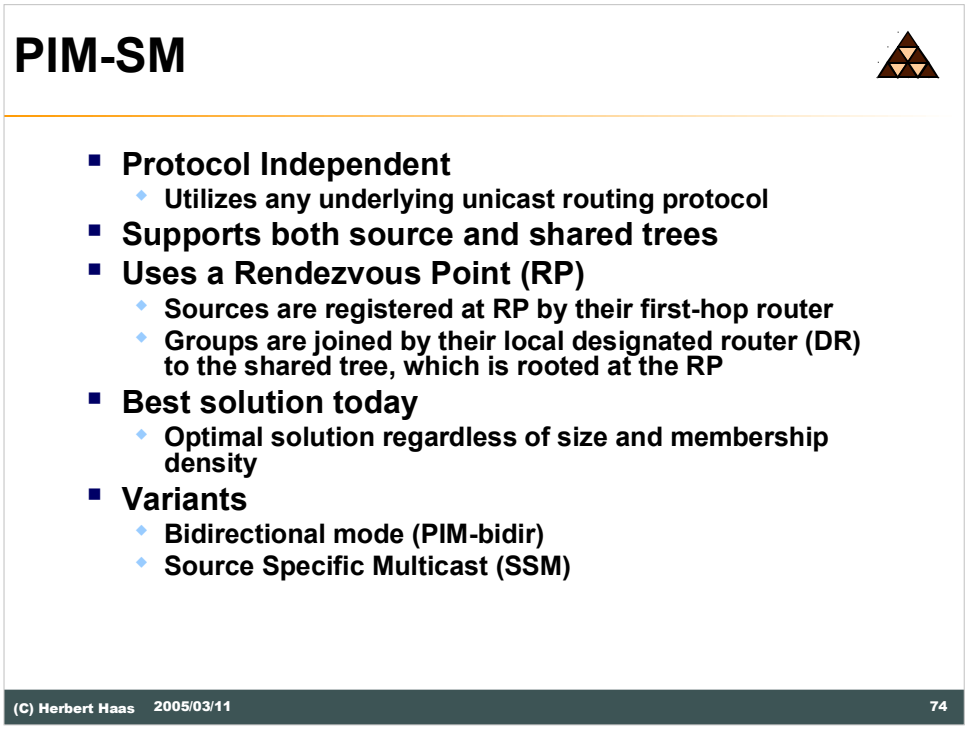
The **assert value** contains both the **administrative distance** of this router and the **metric** toward the source. The administrative distance is evidentially the high-order part of this assert value. Obviously the other router sends also an assert message.

Now both routers compare these values to determine who has the best path (i. e. lowest value) to the source. If both values are the same, the **highest IP address** is used as tiebreaker. Losing routers prune their interface, whereas the winning router continues to forward multicast traffic onto the LAN segment.

I wanted to insert some slides about CBT but this would be only of academic interest. CBT is **not used anywhere** in the real world and it has never been implemented on Cisco routers as far as I know. Sometimes people say that CBT had been only implemented on PowerPoint. But *we* won't do that.

For those of you who are very curious, just a few words here:

- CBTv3 is an Internet-Draft, and there is only an experimental RFC 2189.
- CBT is quite similar to PIM-SM but bidirectional
- Currently there are no switchover mechanisms as provided by PIM-SM



The Protocol Independent Multicast – Sparse Mode (PIM-SM) has been defined in **RFC 2362** and is the most useful multicast protocol today. PIM-SM relies on a explicit pull concept. Traffic is only forwarded to receivers that ask for it (i. e. send a **join** message).

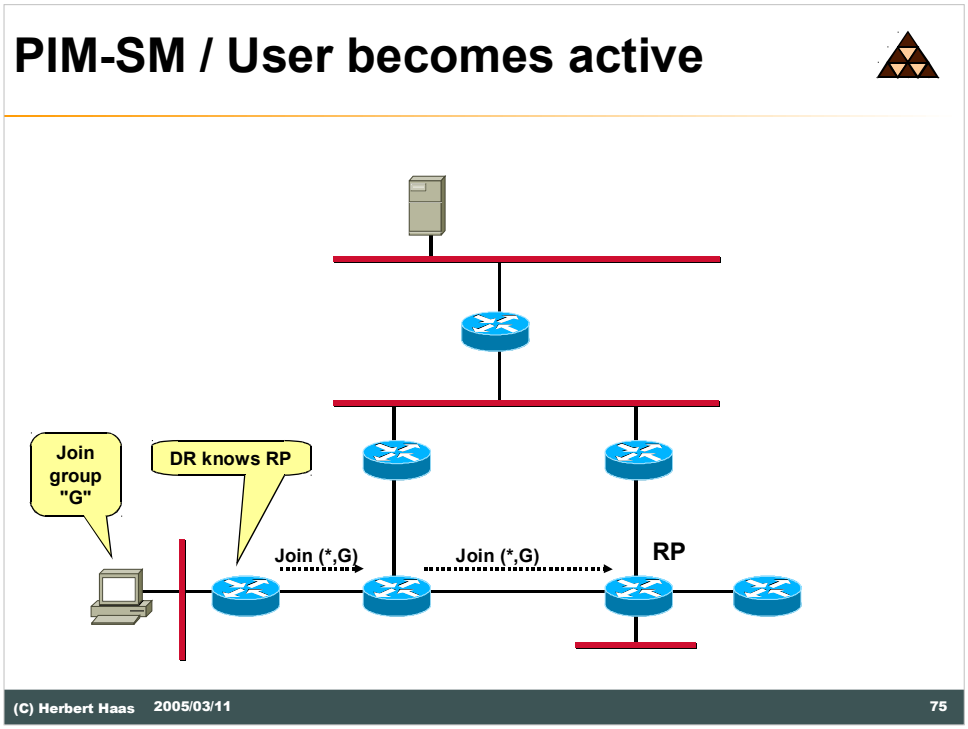
PIM-SM utilizes a **Rendezvous Point (RP)** which roots a shared tree to the groups. The groups are joined by their local designated router (DR) to this shared tree. Basically, PIM SM uses shared distribution trees, but it may also switch to the source rooted distribution tree.

Sources are registered to RPs by so called **register** packets, which are created by the first hop routers, closest to the source. A single copy of the multicast packet is sent through the RP to the registered receivers. Group members are joined to the shared tree by their local designated router. A shared tree that is built this way is always rooted at the RP.

By the way: PIM-SM is the **only solution recommended by Cisco**.

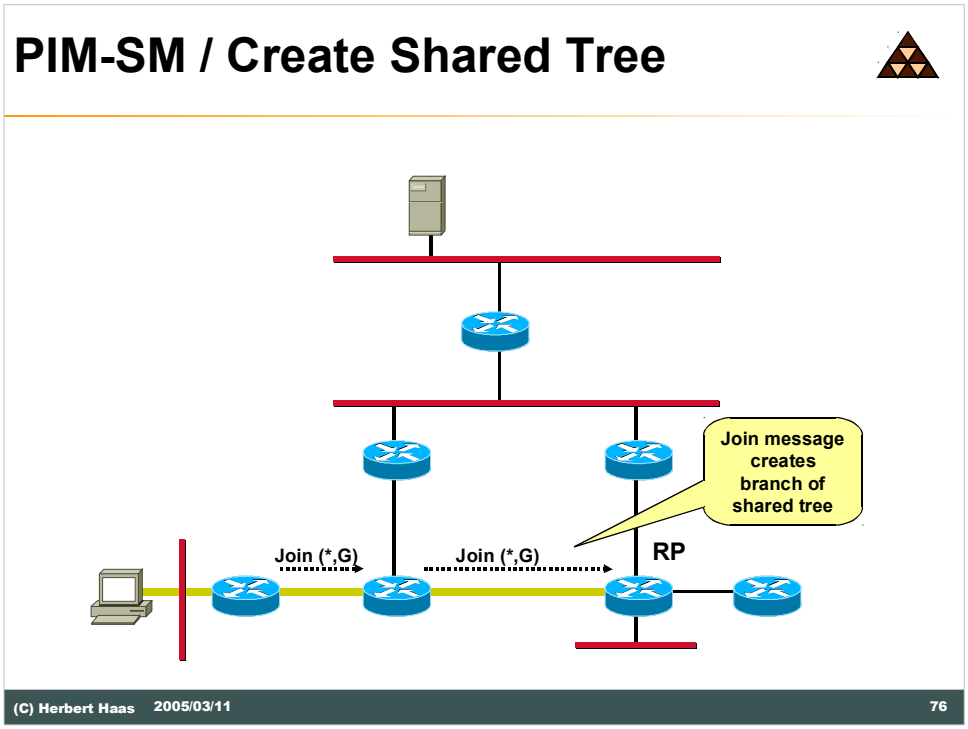
The **bidirectional PIM mode** (PIM-bidir) had been designed for many-to-many applications such as needed for conferencing and whiteboarding purposes.

The **Source Specific Multicast (SSM)** is a variant of PIM-SM that only builds source specific shortest path trees. This solution does not need an active RP and uses the source-specific group address range 232/8.

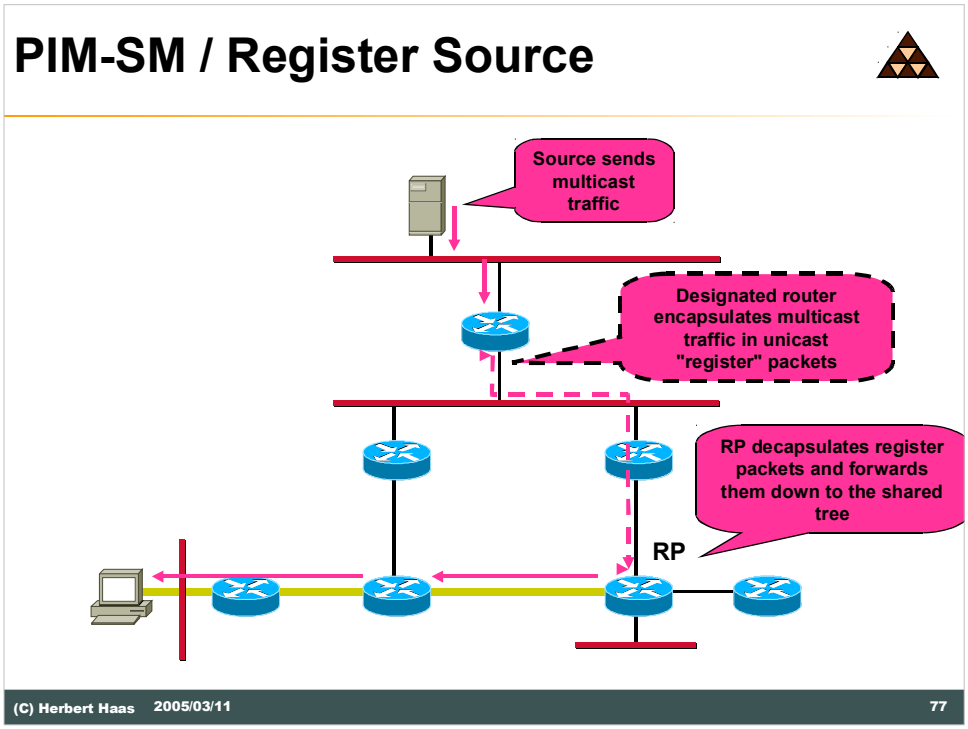


User joins group: The picture above shows how a receiver tells its **designated router (DR)** that he becomes active and wants to listen to group G. This is done using IGMP on the local LAN segment.

The DR sends a **Join (*, G)** to the **RP**. Obviously the DR must know the IP address of the RP. Obviously the DR does not need to know the IP address of the source. Obviously the human receiver should at least know what he wants to listen to.

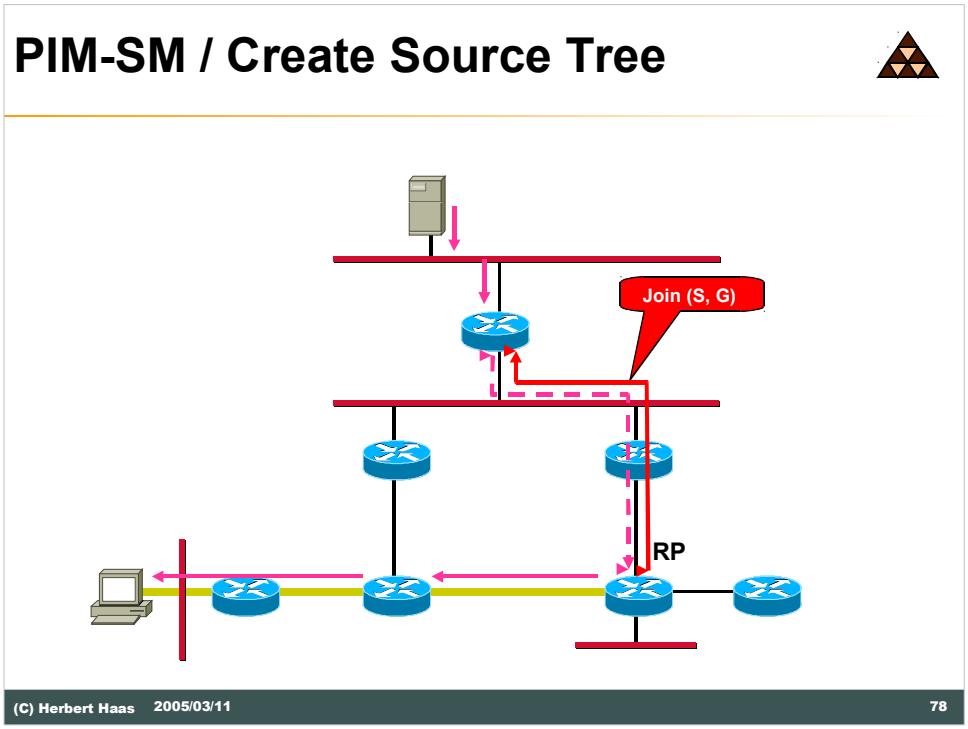


Shared tree to RP: This (*, G) join message is forwarded hop-by-hop toward the RP and hereby a branch of the **shared tree** is established. Now multicast traffic for group G may flow down the shared tree to the receiver.



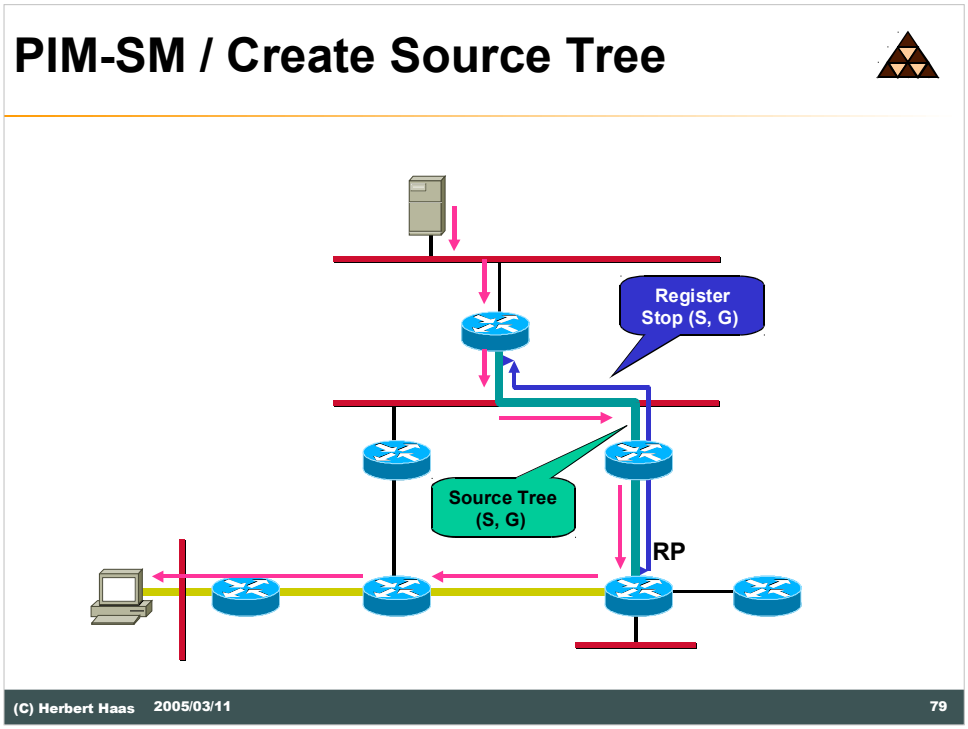
DR registers at RP: The source (for G) becomes active and sends multicast packets, which are encapsulated by the first router (DR) into unicast packets. These "**register**" packets are sent to the RP. Obviously this DR must also know the IP address of the RP.

The RP decapsulates this packets and forwards the multicast packets (which had been carried inside the register packets) downstream to the group G.

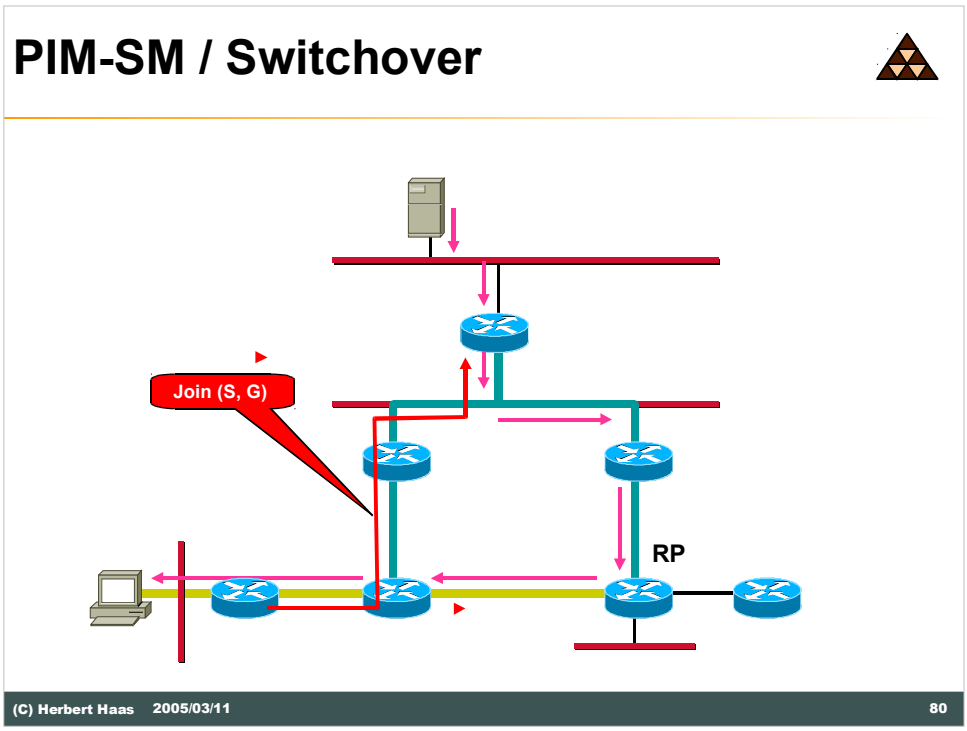


RP joins SPT: Now the RP creates a **shortest-path tree (SPT)** by sending an (S, G) join toward the source. Now (S, G) states are created in all routers along this new SPT path.

Note: Also the RP must maintain a (S, G) state.



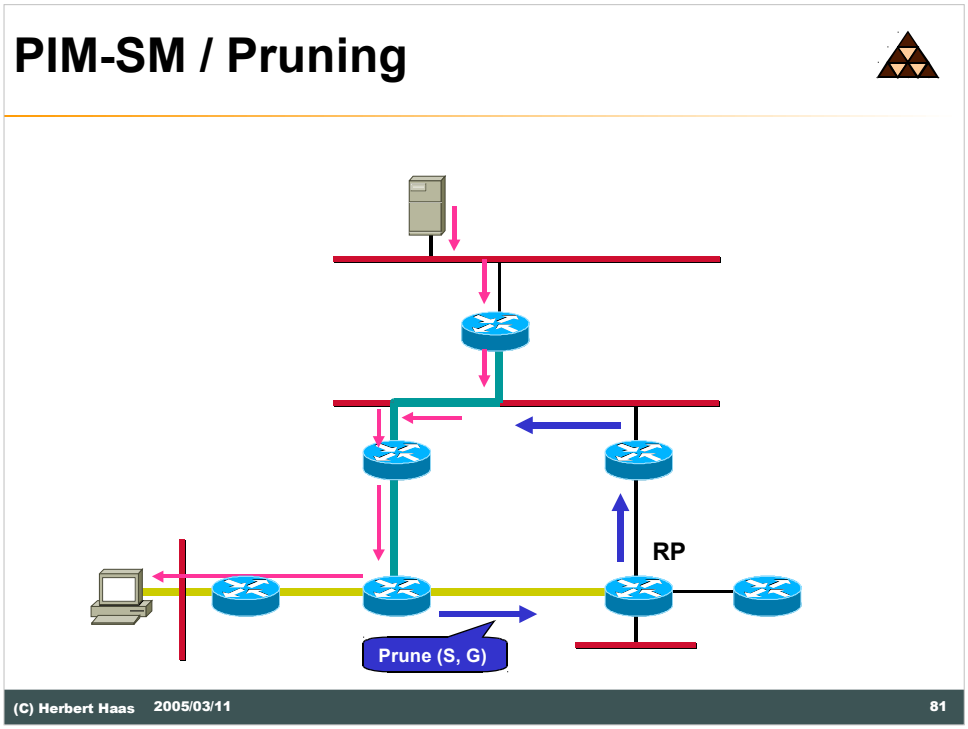
RP stops registering: As soon as native multicast packets arrive at the RP (over the newly established SPT) the RP sends a **"Register Stop (S, G)"** message to the first-hop router, in order to stop the sending of unnecessary register packets.



Shortcut: PIM-SM is able to **switchover** to the **shortest connection** to the source. That is, last-hop routers (i.e. routers with directly connected members) can switch to the Shortest-Path Tree and bypass the RP if the **traffic rate** is above a configured threshold called the **"SPT-Threshold".**

Note: The default value of the SPT-Threshold in Cisco routers is zero.

Therefore the default behaviour for Cisco PIM-SM leaf routers is to immediately join the SPT to the source as soon as the first packet arrives via the (*,G) shared tree.



Disconnect from RP: Now, special **(S, G) RP-bit prune messages** are sent up the shared tree to prune only the (S, G) traffic from this shared tree. This prune is important to avoid duplicate packets.

RP may disconnect from source DR: When the (S, G) prune (with RP-bit set) arrives at the RP **the RP sends (S, G) prune messages back toward the source** to stop the unnecessary (S, G) traffic.

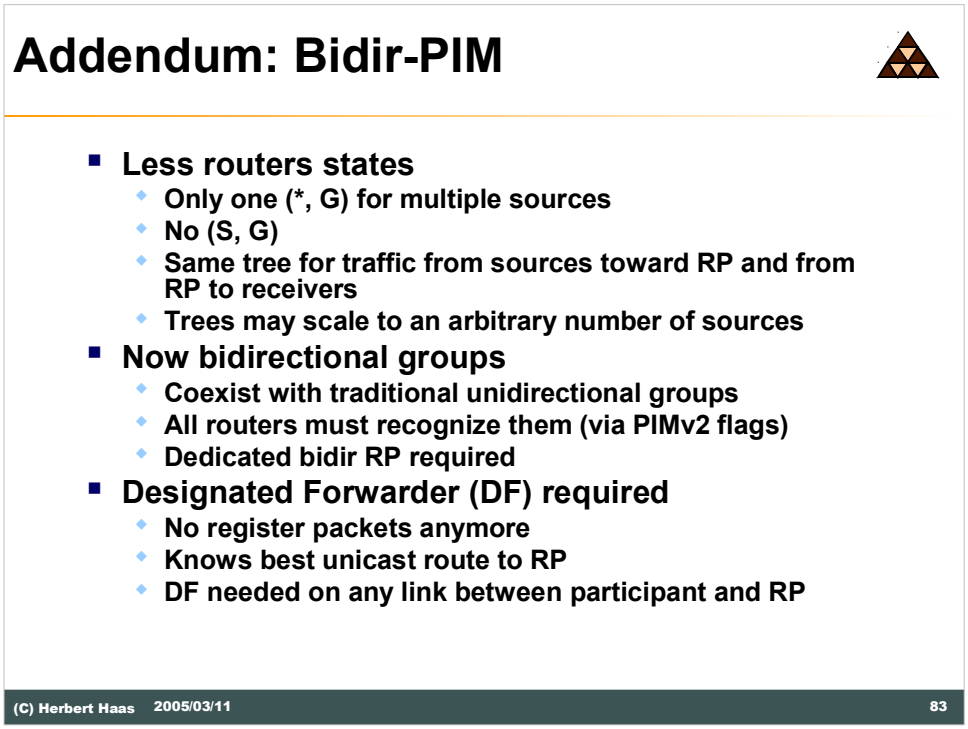
Note: Of course the RP may only do this if the RP has received an (S, G) RP-bit prune via all branches, i. e. **no receiver on the shared tree wants to receive the (S, G) traffic from the RP anymore.**

PIM-SM Summary

- Now we learned:
  - PIM-SM can also create SPT (S, G) trees
  - But in a much more economical way than PIM-DM (fewer forwarding states)
- PIM-SM is:
  - Efficient, even for large scale multicast domains
  - Independent of underlying unicast routing protocols
  - Basis for inter-domain multicast routing used with MBGP and MSDP

Please consider the following issues:

- PIM-SM can be efficiently used for both sparse and dense distribution of multicast receivers.
- There is **no need to flood** multicast traffic at any time.
- On the other hand a RP is needed at least for the initial setup of a MDT.
- PIM-SM can also work together with DVMRP.



Bidir-PIM was introduced with Cisco IOS version 12.1(2)T (5/00). Traditional PIM-SM is unidirectional that is the traffic from sources to the RP is encapsulated in register packets. But this encapsulation and de-capsulation consumes a significant amount of CPU power. Additionally, the SPT which is built between RP and source (initiated by the RP) requires (*, G)(S, G) entries on routers between RP and source.

Using a many-to-many multicast model (where each participant is both receiver and sender) the (*, G) and (S, G) entries appear everywhere along the path from participants and the associated RP. This results in a significant RAM and CPU overhead and may become a significant issue for example with stock trading applications where thousands of stock market traders perform trades via a multicast group.

Bidirectional PIM avoids both encapsulation and (S, G) states. The trick is to ensure that the path taken by packets flowing from the participant (source or receiver) to the RP and the reverse will be the same—only (*, G) states are necessary!

Note: Regular PIM SM groups may coexist with bidirectional groups.

A **Designated Forwarder (DF)** is needed on every link and knows the best unicast route to the RP. The DF forwards both downstream and upstream traffic (from link to RP).

Like in normal PIM-SM the receivers send (*, G) Joins which are forwarded by the last-hop DR toward the RP which is serving the group. But now the DF acts as DR. When a router receives a join message for a bidirectional group the router must determine if it is the DF for this link and for this group. The router either inspects (*, G) state or RP DF election information when there is no (*, G) entry. The shared tree is established between the receiver segments and the RP.

Addendum: PIM-SS

- Source-Specific Multicast (SSM)
  - Much simpler when sources are well known
- Immediate shortcut receiver to source
  - No need to create shared tree
  - DR sends (S, G) join directly to source
  - No MSDP needed for finding sources
- IGMPv3 needed!
  - Or IGMPv3 lite
  - Or URL Rendezvous Directory (URD)

(C) Herbert Haas 2005/03/11 84

The PIM-SS provides all benefits of PIM-SM but **avoids shared trees**. Instead source-specific shortest-path trees (SPTs) are built immediately upon receiving a group membership report for a specified source. SSM is particularly recommended in cases where there is a single source sending to a given group (one-to-many applications).

Note that there is no need for RPs for SSM groups because the discovery of sources is done via some other method (for example web-based directory etc.)

Before SSM, it was necessary to acquire a unique IP multicast group address for any service a source would provide. This was necessary to ensure that different sessions would not collide with each other on the same shared tree.

But when using SSM, traffic from each source is uniquely forwarded only via a SPT and **different sources may use the same SSM multicast group addresses.**

Receivers must have **IGMPv3** implemented in order to send a (S, G) to the DR which forwards an appropriate PIM-join messages to the source. **IGMPv3 lite** is a lightweight interim solution to implement SSM.

The **URL Rendezvous Directory (URD)** communicates (S, G) information via HTTP **redirect** messages (TCP port 659). That is, the browser of the receiver host is redirected by a website to the well-known port 659 with the multicast group and source address as parameters. The DR scans the traffic for this port and therefore learns about the address information.

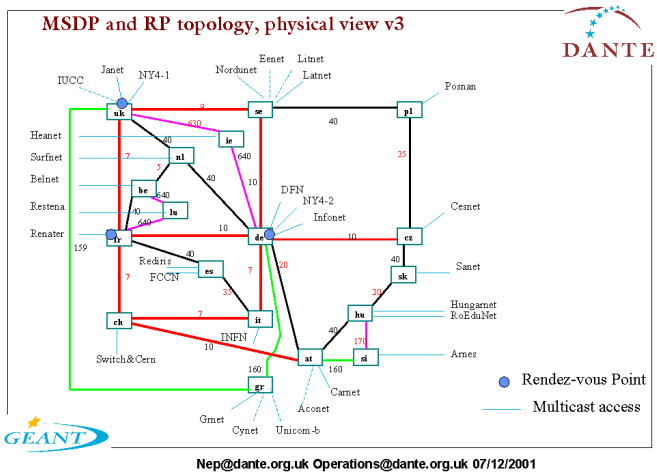
The PIM-SS is still a draft proposal (draft-bhaskar-pim-ss-00.txt).

SSM – Notes

- Take care that no shared tree uses the same group address
  - SSM protocols cannot avoid address collisions
  - Register/Join packets to 232/8 should be filtered

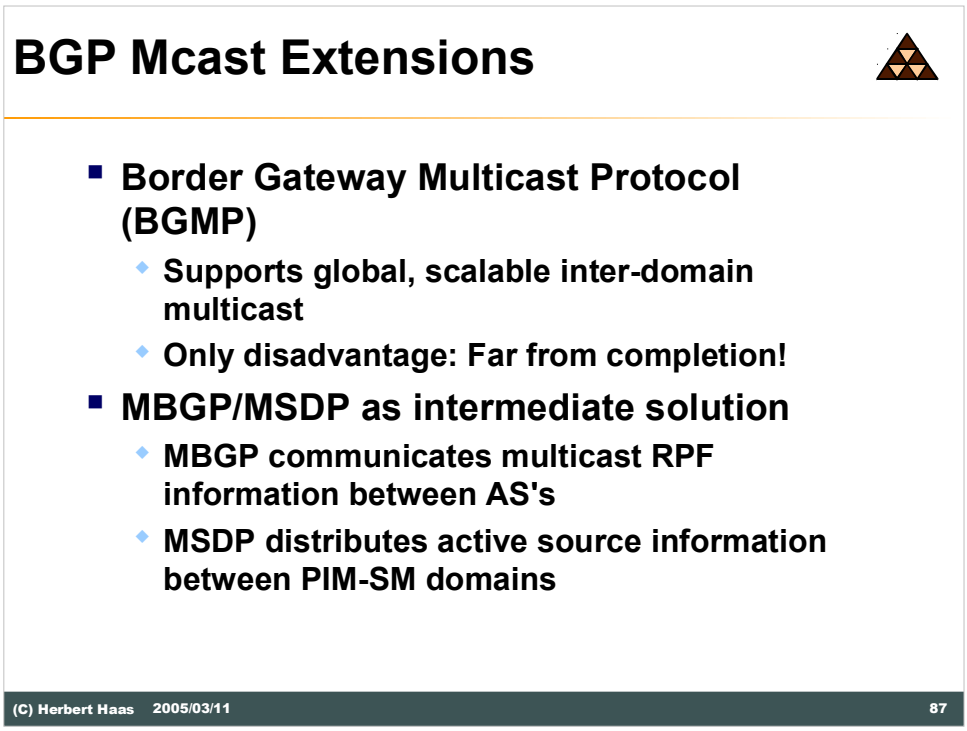
The dedicated address range **232/8** had been reserved exclusively for SSM SPTs. That is, no other router may build a shared tree for any group having an address from this range ("global well-known sources").

Inter-domain Multicast Routing



Nep@dante.org.uk Operations@dante.org.uk 07/12/2001

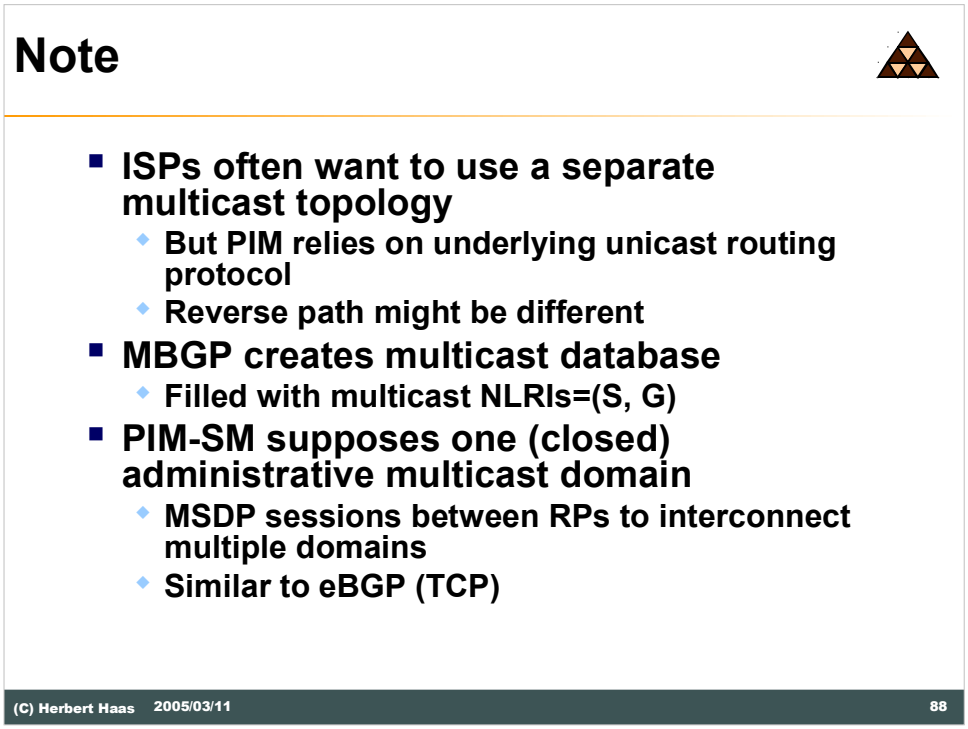
The GÉANT network will provide multicast on all production routers. On the backbone the multicasting is entirely native and sparse mode using PIM-SM. Multicast runs on the same physical infrastructure together with unicast data. Most of the connected National Research Networks have also native connections. All connections between all participants are done via PIM-SM/MSDP/MBGP. The previous TEN-155 Multicasting topology is adapted to the GÉANT topology in the backbone.



The **Border Gateway Multicast Protocol (BGMP)** is far from completion because of its complexity. Today—if really needed—the **combination MBGP/MSDP** is used as **intermediate solution**.

MBGP allows to exchange multicast RPF information between Autonomous Systems (AS). The only difference from ordinary BGP-4 is different NLRI code in BGP messages, allowed by RFC-2283, which are so-called MBGP multicast NLRIs. In other words MBGP is only an extension to BGP. Since MBGP cannot build multicast distribution trees, an additional protocol is used: MSDP.

MSDP is utilized by a PIM-SM domain to tell another PIM-SM domain that active sources exist. Then the routers of the other PIM-SM domain can send (S, G) joins to interconnect sources and receivers in distant domains via inter-domain branches of the SPT.



Routers, which communicate with each other using RFC-2283 BGP extensions ("MBGP") can exchange routing information of several protocols (similar to IS-IS, when configured to carry IP routing information). This functionality of BGP can be used to **exchange reachability information of multicast sources**.

In the current DVMRP Mbone, tunnels are used to bypass non-multicast capable routers in the Internet. MBGP instead creates separate multicast routing tables. Therefore, **different unicast and multicast topologies** may exist: Some parts of the network can be used by unicast only, some by multicast only.

When a router is sending unicast data it never looks into the multicast table. But when the router wants to perform a RPF check for multicast packets it can use both tables. When the multicast and unicast topology is identical, the multicast table is indeed useless, but if the topologies are different this second (multicast) table is used to solve the RPF problem.

PIM-SM supposes the existence of **one administrative domain** having one RP where receivers can request multicast data. But in the real Internet everybody wants to control his own domain. Therefore a new protocol was proposed: MSDP, which allows to communicate active multicast sources among several RPs from different domains.

MSDP runs over TCP similarly to eBGP. There is no need for full mesh TCP connections.

MSDP

- MSDP peering from source RP to
  - Border routers
  - Other AS's RP
- If MSDP peer is a RP and has a (*, G) entry
  - This means there exists some interested receiver
  - Then a (S, G) entry is created an a shortcut to the source is made
  - Furthermore the receiver itself might switchover to the source

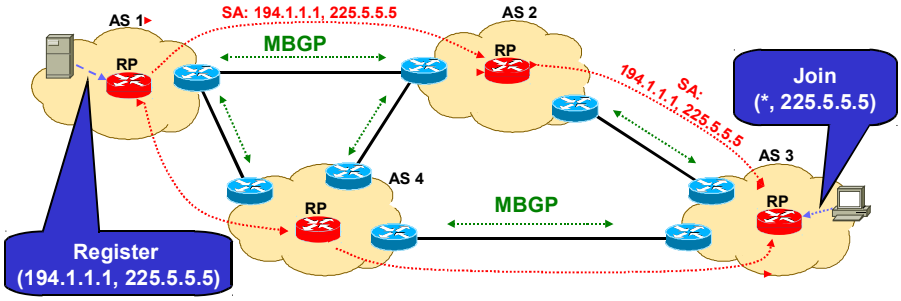
In the PIM Sparse mode model, multicast sources and receivers must register with their local Rendezvous Point (RP). Actually, the closest router to the sources or receivers registers with the RP but the point is that the RP knows about all the sources and receivers for any particular group. RPs in other domains have no way of knowing about sources located in other domains. MSDP is an elegant way to solve this problem. MSDP is a mechanism that connects PIM-SM domains and allows RPs to share information about active sources. When RPs in remote domains know about active sources they can pass on that information to their local receivers and multicast data can be forwarded between the domains.

The RP in each domain establishes an MSDP peering session using a TCP connection with the RPs in other domains or with border routers leading to the other domains. When the RP learns about a new multicast source within its own domain (through the normal PIM register mechanism), the RP encapsulates the first data packet in a Source Active (SA) message and sends the SA to all MSDP peers. The SA is forwarded by each receiving peer using a modified RPF check, until it reaches every MSDP router in the interconnected networks—theoretically the entire multicast internet.

If the receiving MSDP peer is an RP, and the RP has a (*,G) entry for the group in the SA (there is an interested receiver), the RP will create (S,G) state for the source and join to the shortest path tree for the state of the source. The encapsulated data is decapsulated and forwarded down that RP's shared tree. When the packet is received by a receiver's last hop router, the last-hop may also join the shortest path tree to the source. The source's RP periodically sends SAs, which include all sources within that RP's own domain.

MBGP/MSDP (1)

- ASs establish multicast peering using MBGP
  - Via special Multicast RPF NLRI types
  - Used by PIM-SM to send (S, G) joins
- MSDP tells all RPs about *active* sources
  - Using Source Active (SA) messages
  - Containing (S, G) information



(C) Herbert Haas 2005/03/11 90

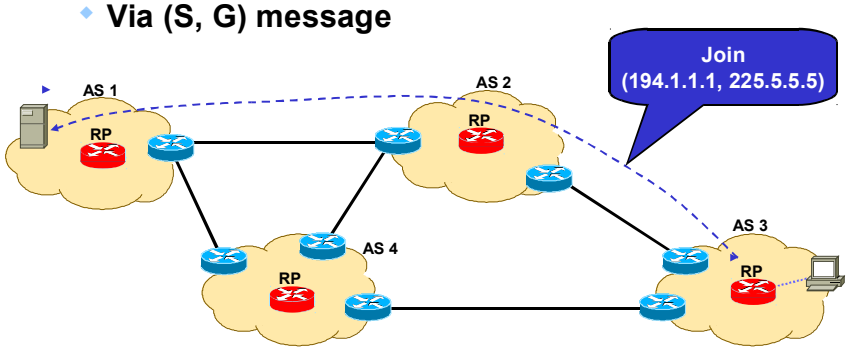
Routers on the borders of domains establish **MBGP peering** and exchange multicast RPF NLRI which is used by PIM SM to determine which way to send (S, G) joins.

RP-routers establish **MSDP peering** and exchange information on active sources and groups.

Note: Since BGP now has to deal with both inter-domain unicast NLRI and inter-domain multicast NLRI the resulting inter-domain unicast traffic paths may differ from inter-domain multicast traffic paths.

MBGP/MSDP (2)

- Receiver joined local RP
  - Via (*, G) message
- Local RP joins source directly
  - Via (S, G) message



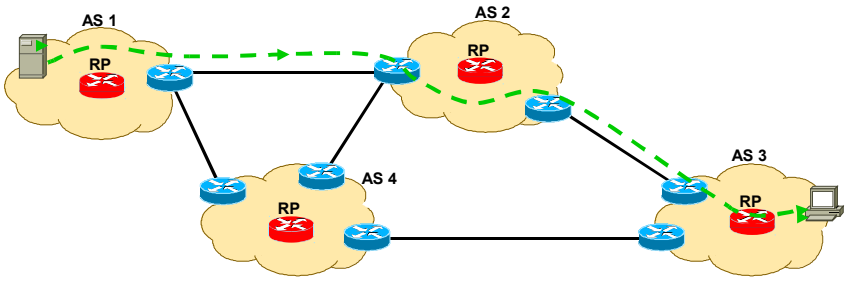
(C) Herbert Haas 2005/03/11 91

The source sent register packets—hereby declaring (S, G)—and the receiver sent (*, G) join packets to the local RP.

Since the local RP learned about (S, G) via the previous SA messages, the RP can **directly join** the SPT which is rooted at the **remote DR** by sending a (S, G) join.

MBGP/MSDP (3)

- Multicast traffic flows directly from the source to the receiver
  - Along a SPT downstream (to perhaps multiple receivers)
- Note: DRs and intermediate routers are omitted for simplicity!



Now the DR of the source can send the multicast traffic directly to the receiver's RP over a SPT.

Reliable Multicast

What is this? Who needs it?

- **Reliable transmission means: no single bit gets lost over MDT !!!**
- **Traditional multicast can't guarantee that—and doesn't need to!**
  - **Audio and video does not bother**
- **But important for *data-based* applications**
  - **Whiteboarding**
  - **Efficient Usenet updates**
  - **Database synchronization**
  - **etc...**
- **Also real-time demands**
  - **Financial data delivery**

(C) Herbert Haas 2005/03/11 94

Traditional multicast had been designed to distribute **bulky voice and video** streams most efficiently. Clearly those are **realtime** protocols which additionally must be transmitted **isochronously**.

Isochronous means: a piece of voice or video data (typically some 30-200 Bytes) are only relevant at one **instant** of time—but a few seconds later it is useless. This time relates to the **playback-buffers** within the multicast applications and in case of interactive applications this time also depends on the affordable **response time**.

For example during a Voice over IP (VoIP) conference, it is important that the total round-trip-time (RTT) is no longer than 0.5-1 seconds—otherwise the users would get angry, as they cannot debate in a reasonable way.

But note that it is absolutely **irrelevant** if, let's say 0.01% of all transmitted bytes gets **damaged** due to higher sun spot activity. You won't notice!

But also pure **data applications** would like to utilize IP multicast technology. For example **Usenet** ("News") data could be efficiently updated to all servers that belong to this world-wide network. Here, it is very important that each single bit is transmitted reliably without being corrupted.

Soon we'll see that this is accomplished not by deploying all-optical networks and quantum entangled photons (joke-alert!) but with a simple but specialized **reliable multicast protocol**.

Reliable Multicast (1)

- Remember: IP multicast is UDP based!
  - No guaranteed packet delivery!
  - No congestion control
  - Not intended for data transactions!
- RTP/RTCP only cares for
  - Duplicates
  - Sequence
- Reliable multicast requires UDP-based acknowledgements
  - TCP cannot do multicast by nature (too much overhead, state variables, buffers, timers, ...)
- Security issues for financial data delivery etc.!!!

Best effort delivery results in occasional packet drops. Many real-time multicast applications such as video and audio streaming may be affected by these losses. On the other hand it is clearly useless to request retransmissions of each lost data.

However, some compression algorithms may be severely affected by even low drop rates; this causes the picture to become jerky or to freeze for several seconds while the decompression algorithm recovers.

Duplicate packets may occasionally be generated as multicast network topologies change.

Reliable Multicast (2)

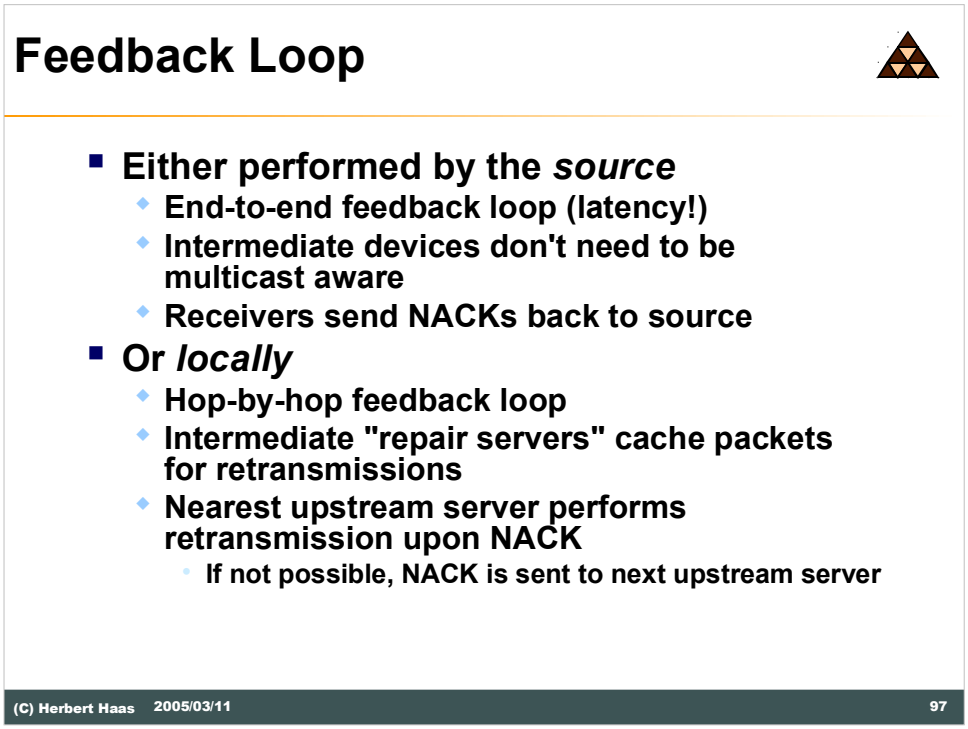
- Guaranteed data delivery is provided by reliable multicast protocols
- Still UDP based *of course*
  - But ACKs are additionally implemented: *Feedback loop*
  - Data recovery mechanisms
  - Congestion control mechanisms

Remember that **TCP** cannot help to implement a reliable multicast protocol as TCP supports only **unicast** transmissions.

This is because TCP maintains peer-specific timers and buffers in order to process a very complex algorithm that supports reliability, high performance, and network fairness.

Therefore also reliable multicast must be **UDP based**. But now additional higher-layer functionality is introduced. The most important function is the **feedback loop** which is fundamental for data recovery mechanisms.

Optionally (and recommended) are **congestion control mechanisms** which significantly enhance the performance of reliable multicast implementations. Note that packets are typically dropped when congestion occurs.

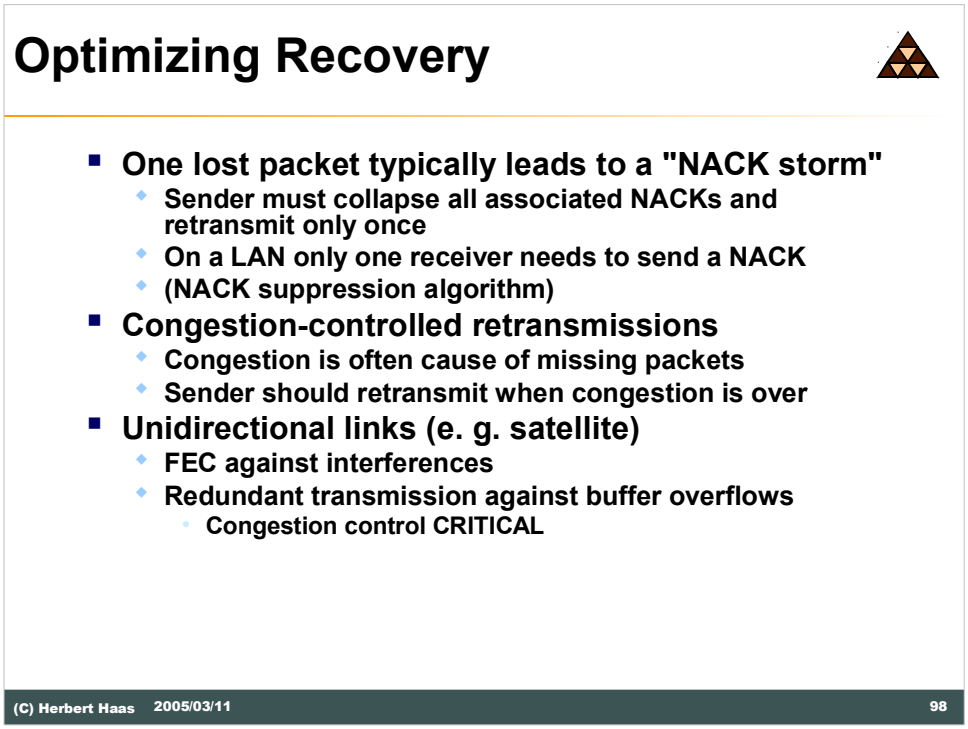


There are **two basic methods** used to implement reliable multicasting:

The first method requires all receivers to send a negative acknowledgement (NACK) back to the source. Thus the source alone is responsible for retransmissions. The intermediate routers do not need to be reliable multicast-aware. This method simply employs an **end-to-end feedback loop.** Obviously, this can lead to significant reparation delays when the path between source and receiver is long.

Note: Other than a normal acknowledgement (ACK) a **NACK** is only sent when a packet is missing. Remember that normal ACKs are sent (e. g. with TCP) for each packet that arrives properly. But this would not scale! Imagine thousand of receivers sending thousand of ACKs back to the single poor source...this would kill it! Instead only NACKs are sent when packets are missing.

The second method requires to employ **special servers for retransmission** within the path between source and receivers. These "repair servers" are also multicast receivers and copy each packet in their **cache**. When the cache is full the next packets overwrite the oldest—like a FIFO principle. If a receiver sends a NACK upstream, the first server which finds the missing packet in its cache will perform the retransmission. Otherwise the NACK is forwarded upstream to the next server. Hence the feedback loop is provided on a **hop-by-hop basis.**



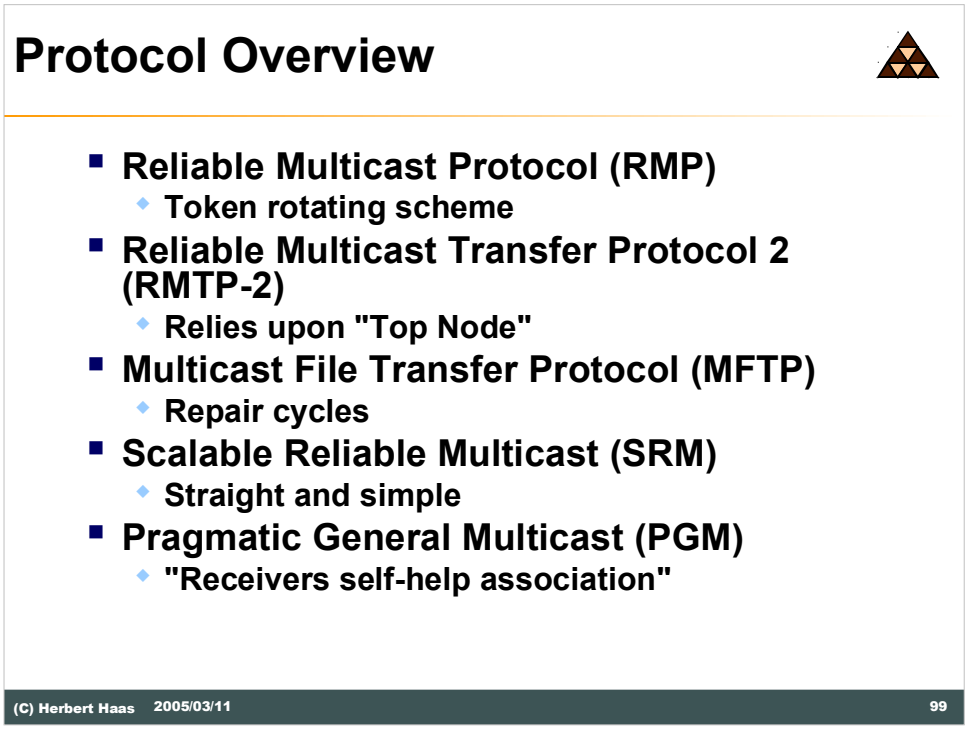
When a packet is dropped at a certain point in the MDT, every receiver residing along the downstream path will send a NACK. The source will be overwhelmed by NACKs—but all NACKs request to retransmit the same packet! That is, the source must logically **collapse** all corresponding NACKs and retransmit only this (single unique one precious) packet.

On a LAN the so-called **NACK suppression algorithm** could be implemented. It works similar as with IGMP report suppression: Upon missing a packet every concerned station starts a countdown which is initialized with a random value. The station whose countdown first expires explodes...no...sends a retransmission of course!

Nevertheless, sources should be able to perform **congestion control**. In most cases a buffer congestion on some helpless router is the real cause for a lost packet. It makes the problem even worse if the retransmission occurs during the congestion.

Finally, a feedback loop makes no sense in some cases (asymmetric bandwidth) or is even impossible with **unicast** (e. g. satellite) links. Here the source must introduce sufficient **redundancy** into the multicast traffic so that the receiver could restore the missing information by itself.

Two possibilities: **Forward Error Correction (FEC)** methods (such as Hamming Codes or Reed-Solomon Codes) can be used to mitigate sporadic bit errors caused by interferences or similar. If whole packets are dropped because of buffer overflow reasons then it might help to send the same packets again and again after some typical congestion period. But without effective congestion control, the multicast transmission is permanent endangered.



The **Reliable Multicast Protocol (RMP)** relies on a rotating-token scheme to ensure reliability and message order. Missing packets are signaled using NACKs via multicast to all receivers. Only stations having the unique token are allowed to multicast an acknowledgment for the recently received packets. RMP is comparable with SRM.

The **Reliable Multicast Transfer Protocol 2 (RMTP-2)** requires a trusted "top node" available for each sender. This top node issues a permission and control parameters for the sender and provide a single point of control/monitor for network managers.

The **Multicast File Transfer Protocol (MFTP)** provides reliable non-real-time bulk data transfer. At first a source transmits the whole data volume and then collects all NACK packets from all receivers. By applying a logical OR operation on the NACK packets the source determines the collective need for repairs (NACK collapse). Then the source starts a summary-retransmission and so on.

The **Scalable Reliable Multicast (SRM)** enables reliable multicast delivery of data packets but without any sequence or delay guarantees.

The **Pragmatic General Multicast (PGM)** ensures reliable multicast delivery and guarantees correct packet sequence and no duplicates. When a packet is lost the affected receiver continuously sends NACKs until the next upstream router replies with a NACK Confirmation Message (NCF). Since this NCF is sent via multicast downstream, all other receivers see it and may perform a local recovery by sending the missing packet. PGM is one of the most promising solutions.

RMP

- Useful for real-time, collaborative applications
- NACKs are sent to multicast address
  - Assures NACK suppression
  - Allows any member to perform retransmission
- Token rotation scheme
  - Owner of token may send ACK referring to recently received packets
  - Allows late joined members to inform about missing packets
- Retransmission to multicast group

(C) Herbert Haas 2005/03/11 100

RMP has been built for online **collaboration** applications with (soft) **real-time** demands. NACKs (and all other packets) are always sent to a multicast group address in order to prevent other members of sending the same NACK (**NACK suppression**) and to invite any member to perform the retransmission.

Furthermore, a **token rotation scheme** has been introduced to provide additional reliability, especially for **late joiners**.

Every time the token is passed to a member, this member sends an ACK to all other members (again addressing this multicast group) which **refers to all recently received packets**. Thus, late joined members can figure out which packets they had missed and can request for retransmission (as soon as they get the token). Obviously those late-joiners can only be served as long the requested packets are available in a cache.

Also the retransmissions are sent to the multicast group address, which might cause **duplicate** packets. Therefore, the receivers must be able to detect and eliminate duplicates.

RMTP

- Useful for bulk data distribution
- Hierarchically structured
- Periodic status messages:
  - Sent by leaf receivers to their designated receivers (DR)
  - Relayed via higher layer Designated Receivers up to the Sender
- Local retransmission and late joins possible
- Caching mechanisms in Designated Receivers

The Reliable Multicast Transport/Transfer Protocol (RMTP) represents actually a whole **family** of similar protocols which are mainly used for reliable bulk data distribution such as file transfers.

An RMTP environment is **hierarchically** organized whereas each "layer" relays status messages vertically from receivers toward the source. Those status messages are periodically sent by the receivers upwards to the next-level designated receiver(s) and so on. This way it is assumed that some near receiver could perform the desired retransmission based on its **cache**.

Again, late joiners could be served. The RMTP protocol is slightly similar with PGM.

MFTP – 1. What is it?

- Useful for non-realtime bulk data distribution only
  - Developed by StarBurst Communications and Cisco Systems
  - Internet-draft February 1997
- Also includes diagnostic tools
  - Multicast ping (senders learn group population)
- Good scalability (thousands...)
- Flexible transport
  - Unicast, multicast, or broadcast dependent on number of receivers and medium

MFTP had been developed by StarBurst Communications (www.starburstcom.com, now acquired by Adero, Inc. in March 2000) and improved by Cisco. MFTP had been developed to transport **bulk data** such as file transfer in an efficient and reliable way—but not too fast. Similar as traditional FTP, MFTP also consists of a **Multicast Control Protocol (MCP)** and a **Multicast Data Protocol (MDP).**

Although MFTP is not the fastest protocol it is **scalable** to thousands of receivers over one-hop networks such as satellite links. Furthermore MFTP is **flexible** regarding the underlying medium. Depending on the number of receivers a sender my also choose **unicast** instead of multicast. If the underlying network does not support multicast the sender may also choose **broadcast**.

MFTP – 2. How does it?

- Server announces transmission and waits for receiver registration
  - Hereby learning population
  - Announcement contains filename and size
  - Well-known multicast group address for announcements
  - Registration suppression on LANs
- Then data is sent and NACKs collected
  - NACKs are collapsed, retransmission *afterwards*
  - Several retransmissions if necessary (slow but reliable)

The source **announces** any transmission in advance to a well-known multicast group address by specifying the filename, the file size and other common file parameters. Note that the draft standard does not specify the multicast address itself (just well-known to users) but the **UDP port 5402**.

During a given registration time all interested **receivers will register** and the **source learns the population size. If multiple receivers reside on the same LAN segment a registration** suppression is performed as in other protocols already mentioned.

Note the simple basic principle: The announcements are sent to a (well-known) **public group** address to which everybody listens so that interested receivers may join but the actual data is then sent to a **private group** address.

During transmission the source collects NACKs but the actual retransmission is done **after** the complete file had been transmitted once. The source patiently repeats the retransmissions until all packets are received correctly by all receivers.

Again, the source **collapses** all NACKs by a logical OR operation.

MFTP – 3. Protocol Details

- File is sent in blocks
  - Some 1000 packets per block
  - Consists of Data Transmission Units (DTUs)
  - Source sends status request message after each block
- NACKs are sent after each block
  - Containing bit-map indicating bad DTUs
  - Unicast
- ACKs could be sent but are typically turned off to reduce traffic
  - Only one ACK at the session end is required

All data of a file is sent in blocks of same length. Each block is further subdivided by one or more Data Transmission Units (DTUs), which consists of several IP packets. After each block the source sends a **status request** message indicating that whole block has been sent. At this time every receiver checks if data is missing and if so a NACK can be sent which contains a bit map indicating block/DTU/packet numbers of the missing data.

The source does not wait for the NACKs but continues to send the next block and so on. Only at the end of the file the source starts to retransmit all requested packets—as already mentioned.

Also ACK messages are defined but the source does not expect them. Typically they are not sent in order to reduce the amount of traffic. Only at the end of the transmission, all receivers must send **one ACK** to signal that the whole transmission succeeded.

MFTP – 4. Three Group Models

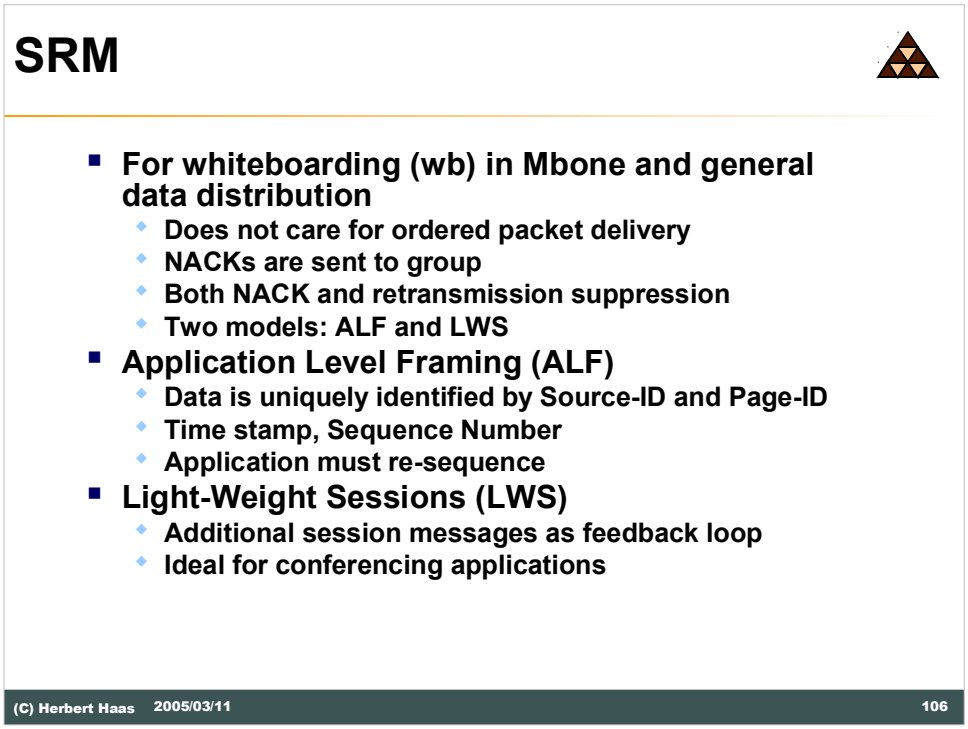
- Closed groups
  - Members are known by source
  - Only those members may register
- Open limited groups
  - Unknown members
  - Source expects registration
- Unlimited groups
  - No registration expected

MFTP allows the multicast service provider to define three different types of groups.

All members of a **closed group** must be known by the source. This model allows for dedicated authorization and is typically applied only for a small number of receivers. Here the source specifies the receivers within the announcement.

Members of an **open limited group** are not specified by the announcement. Any receiver may join the source but must register to the source. The number of receivers is typically limited.

Members of an **unlimited group** do not need to register and even the source sends no announcements at all. There are no limits in group size.



SRM is used by the whiteboarding tool (wb) of the Mbone toolset and some other distributed interactive applications such as simulations or distributed computing environments.

NACKs are sent to the **group** (and not directly to the source) to invite any receiver to start the retransmission. **Both repair packets and retransmissions** are not sent immediately but after expiration of a **suppression timer**.

Two SRM models had been defined, **Application Level Framing (ALF)** and **Light-Weight Sessions (LWS).**

ALF applies an identifier, a sequence number and a time stamp to each packet which allows a receiver to easily identify and NACK lost data.

LWS simply establishes a session between source and receivers and provides special session control messages (which are exchanged between source and receivers). These **session messages** are used by the receivers to tell the source which packets had been received and the source uses them to check the receiver states. Note that NACK messages are used **independently!** When a NACK gets lost the source still notice outstanding packets by tracking the session messages. Of course a source might not track back to the very beginning of a session but rather an **actual time frame** is considered. Thus also **late-joiners** can be served depending on the time frame used by the caches.

Optionally nodes can estimate the distance to a sender using session messages. The average bandwidth utilization (typically below 5%) of the session messages is either preset by a reservation protocol (such as RSVP) or adaptively controlled by a congestion control algorithm.

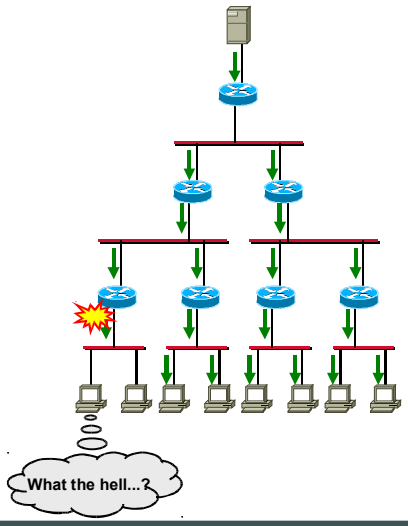
The Pragmatic General Multicast (PGM) is one of the most flexible and **scalable** solutions—and implemented on **Cisco routers!**

On Cisco devices the "**PGM Router Assist**"-feature must be enabled. Those routers will not perform any retransmissions by themselves but support great assistance in **efficient NACK forwarding/filtering** and searching an appropriate retransmitter, which is either the source itself or another receiver which has enough cached data.

Note that PGM is basically layer-3 independent but Cisco IOS only supports PGM over IP.

PGM – Basic Principle (1)

- Source sends ordered data (ODATA) containing
  - Transport session identifier (TSI)
  - Sequence number (SQN)
- Source sends also Source Path Messages (SPM)
  - Interleaved with ordered multicast data
  - Provides an upstream path
  - *Not shown in the picture*



(C) Herbert Haas 2005/03/11 108

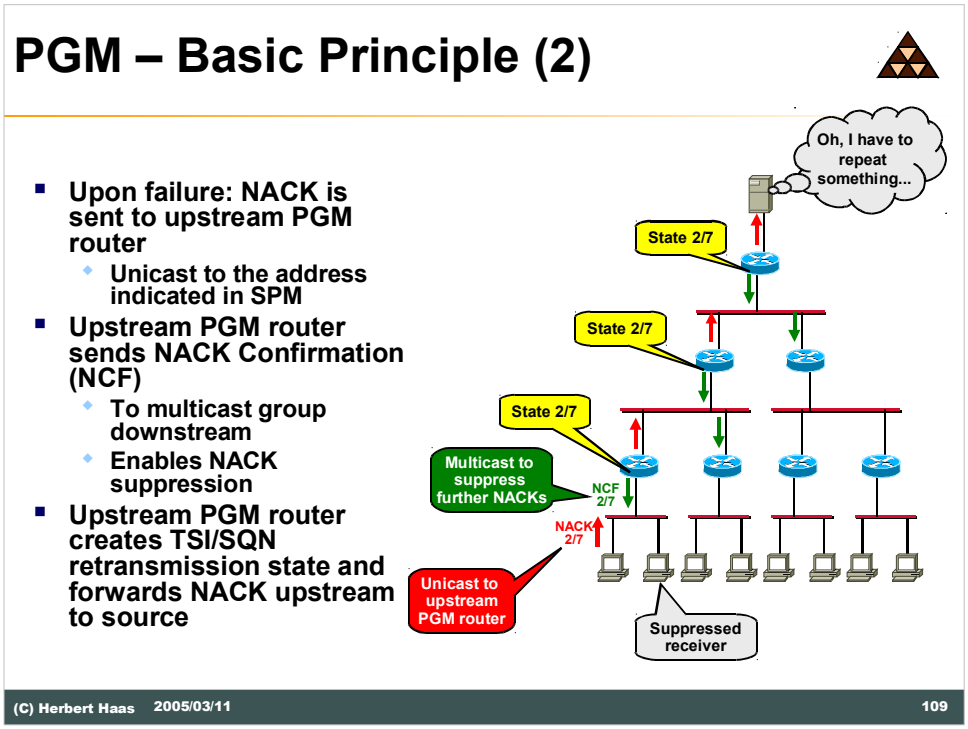
The source sends ordered data (called ODATA by the draft) which is identified by a **Transport Session Identifier (TSI)** and a sequence number (**SQN**). By identifying each session by a TSI label, any number of sources can be handled by PGM. The SQN then further identifies a packet within the TSI-labeled session.

Note: Any **NACK** will refer to a **TSI/SQN pair**.

The receiver learns the information about the next upstream hop from the **Source Path Message (SPM)**, which is periodically interleaved with the data. Each PGM network element inserts its interface IP address (and removes the previous address) through which the SPM message is sent downstream. Therefore, each downstream PGM network element can maintain state information which can be used to send unicast NAKs upstream to the source.

SPM messages have a separate sequence number and must also be NAK-ed if not received.

Note: There must be at least one PGM-enabled router between any source and any receiver.



NACKs are sent **unicast** to the upstream router, which had been learned by the SPM. Each NACK contains TSI/SQN information.

When a router receives a NACK, this router replies with a **NACK Confirmation Message (NCF)** which is sent as multicast through the same interface which received the NACK so that other receivers can suppress their NACKs. PGM-enabled routers never propagate NCFs.

Then this router forwards the NACK upstream and creates a **state** for the TSI/SQN pair. This state allows the router to **filter** any additional NACK and to forward any retransmission. The router continuously propagates the NACK upstream toward the sender until it also receives a NCF from an upstream PGM router. This NACK/NCF process is repeated between each pair of PGM enabled routers until the source itself receives the NACK.

PGM also supports local recovery: Any local receiver may respond with a "**NCF-Redirect**" option and hence becomes a **Designated Local Retransmitter (DLR)** which retransmits the requested data from its own cache. The router forwards all subsequent NACKs directly to this DLR and not upstream.

PGM – Options

- Late joining
  - Sources indicate whether lately joined receivers may request all missing data
- Time stamps
  - Receivers tell urgency of retransmissions
- Reception quality reports
  - Sent by receivers for congestion control
- Fragmentation
  - To confirm to MTU
- FEC
  - To reduce selective retransmissions

(C) Herbert Haas 2005/03/11 110

PGM also supports some **application-friendly options** such as late joining, time stamps, reception quality reports, and others.

The **late joining** option allows a source to tell lately joining receivers whether or not they may request all missing packets.

Additionally **time stamps** can be used in NACKs to allow receivers to tell any PGM device "how urgent" the missing data must be retransmitted.

Reception quality reports may be used in NACKs to support congestion control. This is inserted by the receivers and utilized by the source.

PGM supports data **fragmentation** in order to conform with the maximum transmission unit (MTU) supported by the network layer.

Furthermore, **FEC** can be enabled to reduce the number of selective retransmissions.

Summary

- Multicast routing requires creation of spanning trees
  - Avoid multiple packets
  - Avoid multicast storms
- Source-based and Shared trees
- Push and Pull methods
- IGMP to announce group membership
- Current favourite: PIM-SM
- Also reliable multicast solutions available
  - PGM is most important