Analysis and engineering of the infrastructure required to support multicast traffic at an Internet Exchange Point

Honours Thesis

by

Gerd Besch

Supervisors

Prof. J. Anton Illik

Fachhochschule Furtwangen,
Germany

Ted Hardie, Ph.D.

Equinix Inc.,
USA

submitted: May 2001

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, daß ich die vorliegende Diplomarbeit selbständig und ohne unzulässige fremde Hilfe angefertigt habe. Die verwendeten Quellen und Hilfsmittel sind vollständig zitiert.

Santa Clara, den _____ _____

Gerd Besch

Gerd Besch
2119 Avenida De Las Flores
Santa Clara, CA 95054
USA

email: gerd.besch@gmx.net

Acknowledgements

I am very grateful to Equinix for giving me the opportunity to write my thesis in a great company in the Golden State California.

I am also very grateful to my colleagues in R&D Ted Hardie, Lane Patterson, Diarmuid Flynn and last but not least Jay Adelson for their help and support during the last nine month.

Thanks to Min Zhu and Nitin Jain from Foundry Networks for implementing and debugging the PIM-snooping code and to Marshall Eubanks from Multicast Technologies and Kevin C. Almeroth from UCSB for their advice.

Special thanks to my colleague Ian Cooper for his valuable advice on the question “How to write a *flawless* thesis?”. The answer is: 42.

Abstract

This thesis discusses using “PIM-snooping” as a solution for handling multicast traffic at a Multicast Internet Exchange Point (MIX) on a shared Gigabit-Ethernet switch fabric. Existing MIXs essentially treat multicast traffic as broadcast traffic. Switch ports are flooded due to the lack of information on how to efficiently replicate multicast traffic only onto ports where receivers for specific multicast groups exist.

Protocol Independent Multicast Join/Prune messages can be exploited to provide necessary multicast containment.

Preface

This thesis was conducted at Equinix Inc., USA [1] as the final work in order to obtain my honours degree from the University of Applied Sciences (Fachhochschule) in Furtwangen, Germany [2].

Equinix describe themselves as [1]

> *"... the only company whose sole purpose is to provide a home for the Internet where content providers, ASPs and e-commerce companies can come together and choose best-in-class partners to accelerate their business growth and allow a faster, more reliable Internet. Only Equinix Internet Business Exchange™ (IBX™) centers are designed to provide a neutral environment of unlimited growth that gives customers the power of choice to select the best ISPs, carriers, site and performance management companies, and other key partners to meet their changing needs."*

Equinix also provides dedicated space (also known as the "Sandbox") within each of its IBX™ centers because the Equinix Research and Development philosophy is [3]

> *"... strongly guided by the belief that emerging technologies must be tested in realistic environments before they can be deployed operationally in the Internet."*

All router and switch tests for this thesis have been conducted within the "Sandbox".

Contents

List of Figures

1 Introduction

Internet Exchange Points (IXPs) are neutral meeting places where independent Internet Service Providers (ISPs) exchange **unicast** traffic via a central switch fabric.

Over the last few years more an more ISPs enable **native IP multicast** in their networks. If order to provide the same traffic exchange service for multicast as it is already available for unicast traffic "special" **Multicast** Internet Exchange Points (MIXs) have been installed within the premises of various IXPs. These MIXs usually consist of a **separate** switch fabric that is explicitly used for multicast traffic exchange.

During my internship at the London Internet Exchange (LINX) [4] from March 1999 to September 1999 I worked on a project to implement a MIX [5] in the LINX premises in Telehouse, London (Docklands). I continued to work on this project together with three fellow students during one term (September 1999 to February 2000) at the De Montfort University in Leicester, UK [6].

By the end of this project it turned out that the major problem of handling multicast traffic at an IXP is that the dedicated multicast core switch at the MIX is not 'multicast capable' and therefore **broadcast** multicast traffic onto every switch port which basically turns the switch into a hub.

The sole solution on this problem at that point was a Cisco [7] **proprietary** protocol called Router-Port Group Management Protocol (RGMP) [8]. However, proprietary solutions are not acceptable at an IXP because this would force IXP participants to use proprietary hardware. A new technology to solve the "broadcast problem" was inevitable.

The problem that had to be solved was to gather the necessary information in order to efficiently replicate multicast traffic onto ports where receivers for a specific multicast group exist.

The reception of protocol information at a MIX that can be exploited to provide multicast containment is restricted to certain protocol messages passing through the MIX core switch.

In a Local Area Network (LAN) environment there is already a multicast containment switch feature available that exploits Internet Group Management Protocol (IGMP) messages in order to restrict flooding of multicast traffic from switches to hosts called “IGMP-snooping”.

Based on the same strategy like IGMP-snooping of using IGMP control messages received by a switch to gather information of multicast receivers on specific switch ports this thesis proposes a similar approach called “PIM-snooping” which is based on the same concept of IGMP-snooping but uses PIM control messages instead of IGMP messages to provide a scalable multicast containment solution at a MIX.

2 Multicast Basics

2.1 Unicast

IP communication in the Internet today is mainly based on **unicast** communication. This means one IP host is sending an IP packet to another specific IP destination host. The router on the local subnet and each intermittent router on the way from the source to the destination host forwards the IP packet based on its routing table entries.

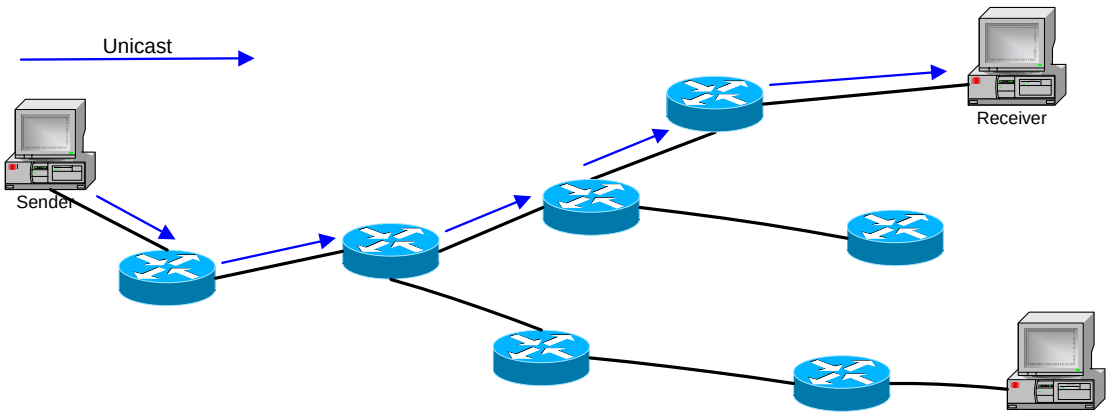

Figure 1: Example of unicast communication

2.2 Broadcast

If a host wants to send IP packets to all hosts on its local subnet it uses **broadcast** packets.

As an example, a host with an IP address of 141.28.2.5 with a subnet mask of 255.255.255.0 has a broadcast address of 141.28.2.255.

```
IP          141.28.2.5     10001101 | 00011100 | 00000010 | 00000101
                           [------------- subnet -------]   [-host-]
Mask        255.255.255.0  11111111 | 11111111 | 11111111 | 00000000

Broadcast   141.28.2.255   10001101 | 00011100 | 00000010 | 11111111
Address
```

Broadcast traffic is restricted to a local subnet and will not be forwarded by a router connected to that subnet. All host IP stack implementations are able recognize broadcast packets and process them accordingly. The sole purpose of broadcast is to communicate with all network components on a local subnet.

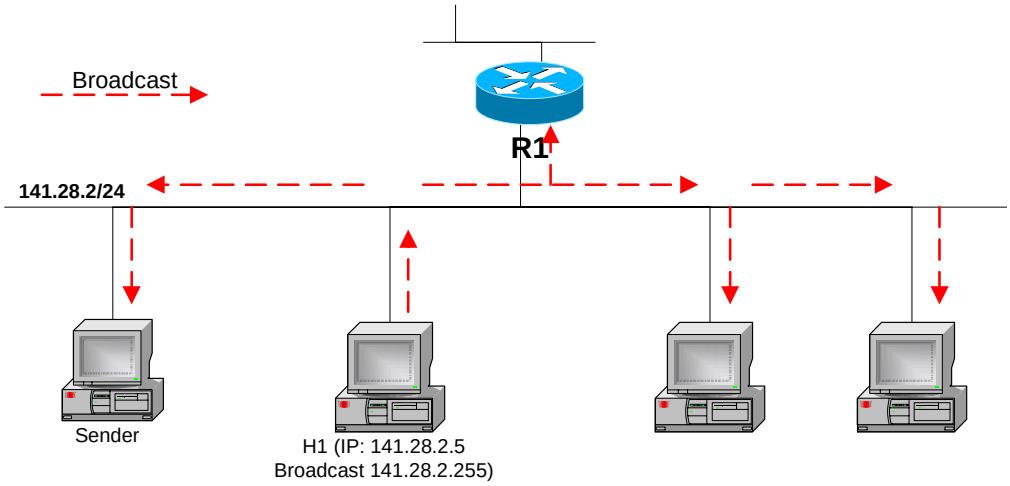

Figure 2: Example of broadcast communication

2.3 Multicast

Back in 1989 Steve Deering [9] first described what kind of extensions in a hosts IP stack would be necessary to provide native IP Multicast capabilities and how multicast works:

> *"IP multicasting is the transmission of an IP datagram to a "host group", a set of zero or more hosts identified by a single IP destination address. A multicast datagram is delivered to all members of its destination host group with the same "best-efforts" reliability as regular unicast IP datagrams … The membership of a host group is dynamic; that is, hosts may join and leave groups at any time. There is no restriction on the location or number of members in a host group. A host may be a member of more than one group at a time. A host need not be a member of a group to send datagrams to it."*

The most important features of multicast are:

- The introduction of sending to, or receiving traffic from a “host group” defined by a **single** IP destination address in the IP Multicast address range (see section 2.7.1) where a “host group” is an **arbitrary** group of IP hosts which can join or leave a group at any time
- A host **does not** have to be a member of a group to **send** IP Multicast datagrams to it. Therefore, **any** host can send multicast traffic to **any** host group at **any** time.
- Multicast packets are **replicated** in the network (by routers/switches) when ever a new branch of the multicast distribution tree is needed (see chapter 3)

Therefore, in order to provide native IP Multicast all hosts that want to receive or send IP Multicast traffic and all intermediate network components are required to be **multicast enabled**.

The requirements for end-node hosts are [10]:

- Support for IP Multicast transmission and reception in the TCP/IP protocol stack
- Software supporting IGMP to communicate requests to join a multicast group(s) and receive multicast traffic (see section 2.6)
- Network interface cards which efficiently filter for LAN data link layer addresses mapped from network layer IP Multicast addresses (see section 2.7.3)
- IP Multicast application software such as video conferencing

An example of a source sending traffic to a specific multicast group is shown in Figure 3.

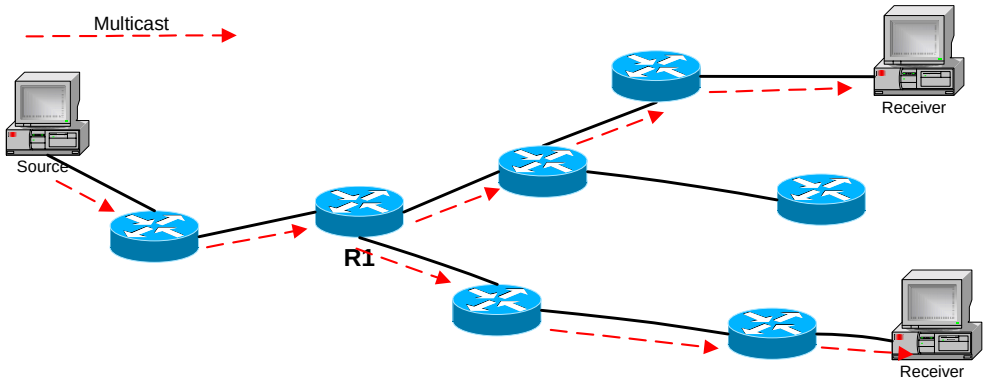

Figure 3: Example of a source sending to a multicast host group

2.4 Service Models

Currently two different service models for multicast exist. The “traditional” Any-Source Multicast (ASM) service model as described by Deering [9] and a new service model that was lately introduced called “Source-Specific Multicast” (SSM). Both service models are described in the following sections.

2.4.1 Any-Source Multicast

The name “Any-Source” Multicast (ASM) comes from the fact that in the ASM service model sources are treated generally. This means that for the identification of a multicast host group the IP address of the source (sending host) is not important whereas in the Source-Specific Multicast (SSM) service model (see section 2.4.2) sources are treated specifically. ASM is like SSM a receiver-driven concept and the receivers are unknown to the sender. An ASM source is not able to retrieve any information about the identity of receivers or the number of receivers.

It is currently being discussed in the “multicast community” on how to name the multicast service model defined by RFC 1112 [9]. In some

documentations the term “Internet Standard Multicast” (ISM) is used whereas in others the term ASM. Both terms refer to the same multicast service model.

In the ASM service model multicast traffic is sent to a “host group”. A host group uses one IP out of the Class D address range as IP destination addresses for all packets sent to a specific group and uniquely identifies a multicast packet in the ASM model (the source address is not important as in the SSM service model described in section 2.4.2). The Class D address range was assigned by the Internet Assigned Number Authority (IANA) [11] and is described in more detail in section 2.7.1.

Multicast applications normally use the User Datagram Protocol (UDP) [12] as transport protocol which means best-effort packet delivery. TCP on the other hand uses a “build-in” congestion avoidance mechanism that causes TCP to backoff and slow-start if a congestion within the network occurs [13]. With UDP this is not the case. Detection of a packet loss must be handled by upper layer protocols e.g. RTP [14]/RTCP [15].

If multicast applications want to make connections reliable they either have to use a reliable Layer 4 transport mechanism like the Transmission Control Protocol (TCP) [16] or some other higher transport layer protocol.

The following figure shows a basic example of Host A sending a multicast packet onto the local LAN with the IP destination address of 224.2.2.1. Host B and Host C are on a separate LAN interconnected by router R1:

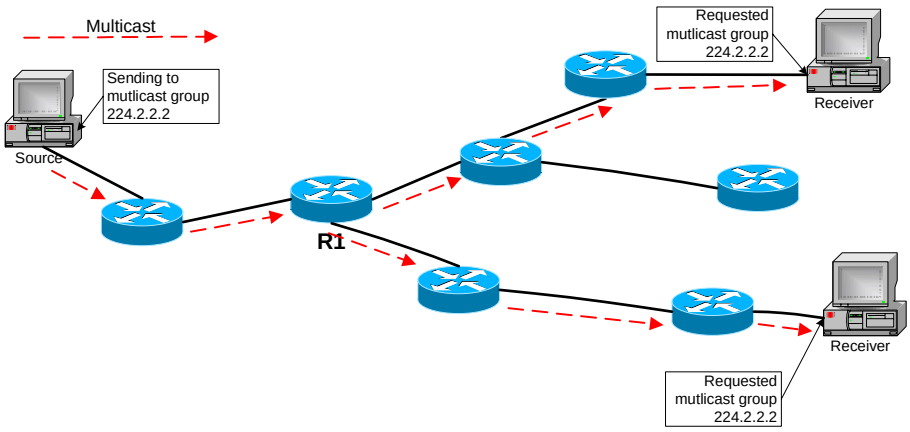

Figure 4: Example of multicast using the ASM service model

How sources are discovered or new source "reserve" multicast group addresses in the ASM model is described in section 4.3.1.

2.4.2 Source-Specific Multicast

Apart from the ASM service model described in section 2.4.1 a new multicast service model has lately been proposed by Holbrook called "Source-Specific Multicast" (SSM) [17] where the term "Source-Specific" comes from the fact that in SSM sources are treated **specifically**, rather than all sources generally like in the ASM service model as shown in Figure 5.

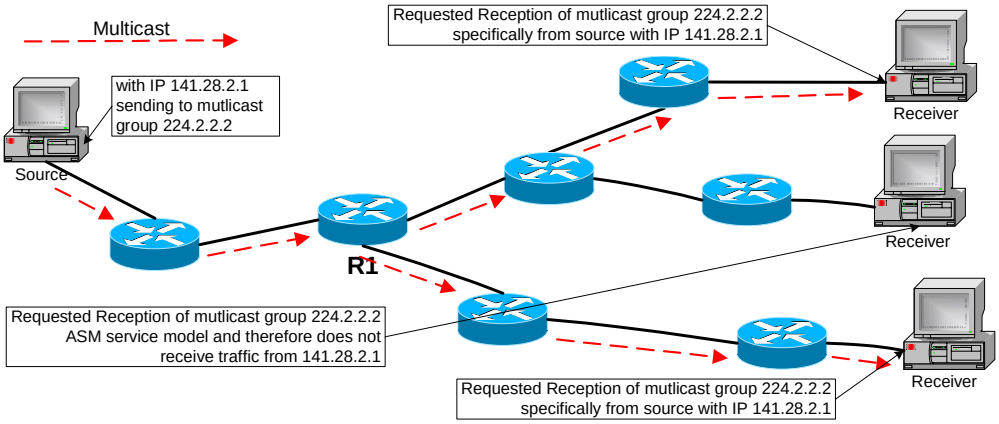

Figure 5: Example of multicast using the SSM service model

Lately SSM was incorporated into the **new** PIM-SM (see section 4.3) protocol specification [18] where SSM (in the spec called PIM-SSM) is described as follows:

> *"The Source-Specific Multicast (SSM) service model can be implemented with a strict subset of the PIM-SM protocol mechanisms. Both regular IP Multicast and SSM semantics can coexist on a single router and both can be implemented using the PIM-SM protocol. A range of multicast addresses, currently 232.0.0.0/8 in IPv4, is reserved for SSM, and the choice of semantics is determined by the multicast group address in both data packets and PIM messages."*

The main differences between the ASM and SSM service model are:

- The SSM multicast service model is based on the same receiver-driven packet delivery model as ASM but uses the multicast group address (G) and the IP address of the source (S) (called an **(S,G) pair**; pronounced “S komma G”) to **uniquely** identify a particular multicast packet whereas in ASM a multicast packet is uniquely identified **only** by its multicast group address (G)
- IANA has assigned the address space **232/8** out of the Class D address space to be used **specifically** for SSM and must **not** be used by applications based on ASM
- The SSM service model supports **only one-to-many delivery** whereas ASM provides one-to-many and many-to-many multicast packet delivery
- SSM uses **only** shortest-path tress and **no** shared trees whereas ASM uses shared-trees **and** shortest-path trees (see section 3)
- Sources in SSM are discovered “out of band” e.g. via a website and not via a globally distributed directory of sources like in ASM
- SSM needs IGMP version 3 (see section 2.6.3) to work properly whereas ASM uses IGMPv1 or IGMPv2

The following list compares the terms used in the two different service models ASM and SSM as described by Holbrook [17].

Service Model:	Any-Source (ASM)	Source-Specific (SSM)
Network Abstraction:	Group	Channel
Identifier:	G	S,G
Receiver Operations:	join, leave	subscribe, unsubscribe

Note: Some documentations translate the abbreviation SSM to "Single-Source Multicast" instead of "Source-Specific" which implies that this multicast service model is based on a **single source** sending to multiple recipients. However, the two terms refer to the same service model.

In SSM the receiver must be specific from which source he wants to receive multicast traffic whereas in ASM it is possible to receive traffic from **any** source.

Because a receiver has to specifically "subscribe" to a multicast channel identified by a unique (**S**ource, **G**roup) pair the IGMP protocol had to be extended to provide this feature. The latest version of IGMP is IGMP version 3 (see section 2.6.3) and is the host membership protocol that is used in conjunction with SSM because it provides the ability to send membership reports that specifically request the reception of multicast packets from a channel (**one** source sending to **one** multicast group).

A receiver can also receive multicast traffic from various sources simultaneously and of course can send to various multicast groups. Therefore it is possible to establish a many-to-many communication by using as many one-to-many channels as needed. Many-to-many communication in SSM is established by having the host to

- "subscribe" to **all** channels from where it wants to receive multicast traffic (identified by an (S,G) pair using IGMPv3)
- start sending multicast traffic to a multicast groups (G) (all receivers in turn subscribe to this group G received from source S → (S,G) pair)

If SSM should be implemented in a network the following conditions have to be met:

- All intermediate multicast routers have to be upgraded to PIM-SM Version 2 (see section 4.3)
- **Only** the DR (the hosts first hop router) has to support IGMPv3 (see section 2.6.3) as the group membership protocol for the local subnet
- If a DR receives an IGMPv3 message for a particular (S,G) group it has to send a join directly towards the source of the multicast traffic because SSM only supports shortest-path trees (see section 3.1) and does **not** support the concept of shared-trees (see section 3.2)

2.5 Scalability Benefits

The major benefit of multicast is its scalability in broadband delivery of video or voice based streaming data. As shown in Figure 4 and Figure 5, in both service models a source is sending traffic only once towards the downstream router.

Form the perspective of the source it makes no difference if there are just a few receivers for that multicast traffic or if there are e.g. 10,000 receivers for a particular multicast group because multicast packets are replicated along the multicast distribution tree when ever this is needed.

In a unicast scenario a source would have to send the same packets to each single receiver which could exceed the available network bandwidth. Using multicast the packet is only sent **once** along each link between routers and hosts.

E.g. if a source is sending a 100 kBit/s video stream to 10,000 receivers then the source would have to send in the case of

- unicast: 100 kBit/s * 10,000 receivers = 1,000,000 kBit/s (at least) onto its first hop router

- multicast: 100 kBit/s * 1 = 100 kBit/s because the packets are replicate within the network towards receivers

The scalability pros of multicast can be summarized as follows:

- Streaming server CPU loads can be significantly reduced and network bandwidth can be saved
- One-to-many or many-to-many communication is possible

LaMaster notes [19]:

> *"Why multicast? Multicast is often the only way to send high-volume, real-time data from a single sender to a large number of recipients. NASA has high-bandwidth applications that require speeds in the millions of bits per second (Mb/s). If you had to send out large files or real-time data streams to a million receivers without multicast, you would have to send out a million streams of the same data. That method doesn't scale to large numbers of receivers. … The ability to do native multicast on today's higher speed routers is fairly new. In the future, all networks will be multicast-enabled."*

2.6 Internet Group Management Protocol

The Internet Group Management Protocol (IGMP) was first defined by Deering [9] as a protocol that

> *"… is used by IP hosts to report their host group memberships to any immediately-neighboring multicast routers. … Like ICMP, IGMP is a integral part of IP. It is required to be implemented by all hosts conforming to level 2 of the IP multicasting specification."*

Multicast capable hosts use IGMP to report the need of reception of a particular multicast group to their locally connected multicast subnet routers. These multicast subnet routers in turn use this information to maintain a list of multicast groups (associated with a timeout value) for which they have to forward multicast traffic.

The IGMP mechanism is based on a Query and Response mechanism. A "Query" is sent by **one** router on a local subnet whereas host responses are sent by hosts on the local subnet and are called "Reports" because hosts report their multicast group membership with this messages.

IGMP messages are sent within IP packets and use an IP protocol number of 2. The time-to-live (TTL) field in the IP header is set to 1 in order to prevent forwarding of IGMP messages out of the local subnet.

RFC 1112 [9] also states that

> *"... every level 2 host must join the "all-hosts" group (address 224.0.0.1) on each network interface at initialization time and must remain a member for as long as the host is active.".*

This is necessary because membership queries are sent to the "all-hosts" group and every host on the system has to receive the queries.

Since the first version of IGMP the protocol has evolved into new versions with new features described in the following three sections.

2.6.1 IGMP version 1

The IGMPv1 protocol definition [9] does not mention a procedure of how a Querier is elected (IGMPv2 has a procedure for this; see section 2.6.2) and relies on a IP Multicast Routing Protocol like PIM [39] or the Distance Vector Multicast Routing Protocol (DVMRP) [20] to elect a Designated Router (DR). The DR is the designated multicast router that forwards all multicast traffic for the subnet. In IGMPv1 this DR is associated with the role of the IGMP-Querier for the local subnet by default. It is **not** important for the DR to know how many hosts are interested in a particular multicast group on the subnet it has only to know that there is at least **one** recipient for a particular multicast group (see "Report Suppression Mechanism" described later in this section).

The DR receives all Membership Reports and therefore knows for which multicast groups it has to forward traffic onto the local subnet. It is important to note that (as described earlier) the DR does **not** know **how many** hosts on the subnet want to receive the multicast group it is only important to know that at least **one** host wants to receive this group.

One of the main problems of IGMPv1 is that it provides no mechanism to directly inform a DR that a host wants to stop the reception of a specific

multicast group. The DR only knows about the fact that all hosts have left a specific multicast group if it receives no answer from any host after it have sent a Membership Query onto the subnet. This causes a very high latency of leaving host groups (leave-latency). IGMPv2 (see section 2.6.2) introduces a special IGMP Leave-Group message that solves this problem.

According to Williamson [13] IGMP version 1 (IGMPv1) is still widely used in today's IP stacks. E.g. Windows95 uses IGMPv1 unless it is upgraded to a later version of Microsoft's Winsock Dynamic Load Library (DLL). Also many UNIX implementations still use IGMPv1 but patches for IGMP version 2 (see section 2.6.2) exist, or the latest UNIX version has to be installed.

2.6.2 IGMP version 2

IGMP version 2 (IGMPv2) is defined in RFC 2236 [21] and is designed to be backward compatible with IGMPv1 [9].

New features introduced by IGMPv2 are:

- The protocol provides two different Query messages. "**General-Queries**" (for backward compatibility with IGMPv1) and "**Group-Specific Queries**"
- A Query message contains a "**Maximum Response Time**" field. It is used to tell hosts about the maximum of time they have to respond to a Query. This is used to tune the leave latency which was an inherent problem in IGMPv1
- IGMPv2 hosts send "**Leave-Group messages**" when they leave a particular multicast group. This prevents high leave latencies as in IGMPv1 which causes unnecessary multicast traffic on the subnet
- The **Querier-election process** is now handled in IGMPv2 and does not rely on the multicast routing protocol that is used
- All IGMPv2 routers join the "all-routers" multicast group 224.0.0.2 (ALL-ROUTERS.MCAST.net)

- All IGMP messages are encapsulated in IP datagrams, have an IP protocol number of 2, a TTL of 1 **and** have the “IP Router Alert option” set

General-Queries are used to ask all host on the subnet for a complete report about their multicast group memberships. These messages are used as a way to insure backward compatibility with IGMPv1. In a General-Query the Group Address field contains the IP 0.0.0.0 whereas in a Group-Specific Query the Group Address field is set to the Group Address for which the Query is sent. Group-Specific Queries are sent by a router to ask all hosts on a subnet if there is still at least one receiver for one particular group on the subnet. If no host replies to this Group-Specific Query then the router stops forwarding multicast traffic onto the subnet which prevents using unnecessary bandwidth on the subnet.

A summary of changes between IGMPv2 and IGMPv1 can be found in the Appendix of RFC 2236 [21].

2.6.3 IGMP version 3

The latest IGMP protocol is version 3 (IGMPv3) [22] and provides a new IGMP message format to signal the DR which multicast group (G) should be received from **which source (S)**. The functionality defining the group **and** the source for where a host wants to receive multicast traffic is very important for SSM to work properly (see section 2.4.2).

The IGMPv3 spec defines the functionality as follows:

> *“IGMP is the protocol used by IPv4 systems to report their IP multicast group memberships to neighboring multicast routers. Version 3 of IGMP adds support for "source filtering", that is, the ability for a system to report interest in receiving packets *only* from specific source addresses, or from *all but* specific source addresses, sent to a particular multicast address. That information may be used by multicast routing protocols to avoid delivering multicast packets from specific sources to networks where there are no interested receivers.”*

A IGMPv3 implementation (Kernel patch) is currently only available for Linux [23] or FreeBSD [24].

Because there is currently no IGMPv3 stack implementation available (e.g. in any of the current Windows or Unix IP stacks) Cisco provides two interim solutions for hosts to request SSM multicast groups and to signal this to the DR. These Cisco **proprietary** protocols are

- Internet Group Management Protocol version 3 lite (IGMP v3lite) and
- URL Rendezvous Directory (URD)

Details about these protocols can be found on the Cisco SSM website [25].

2.7 Addressing

2.7.1 Assigned Multicast Address Space

The Internet Assigned Number Authority (IANA) [11] assigned a specific IP address space [26] for the use with IP Multicast, the old class D address space that is defined by a binary prefix of '1110' and which is equivalent to a classless interdomain routing (CIDR) prefix of 224/4.

		Octet 1	Octet 2	Octet 3	Octet 4
First IP	224.0.0.0	**1110**0000	00000000	00000000	00000000
Last IP	239.255.255.255	**1110**1111	11111111	11111111	11111111
Netmask	240.0.0.0	**1111**0000	00000000	00000000	00000000

The netmask has to preserve the prefix of '1110' and therefore the first octet is set to '11110000' (decimal: 240).

2.7.2 Address Scoping

Multicast uses several concepts described in the following sections to restrict the delivery of packets out of a certain scope like a subnet or a administratively defined multicast region.

2.7.2.1 Link-Local Addresses

IANA defined multicast addresses with a link-local scope. The address range for link-local multicast addresses is **224.0.0/24**. The distribution of multicast traffic with a link-local scope is limited to the **local network segment** regardless of the Time to Live (TTL) value in the IP header. The following table represents an excerpt of addresses defined in RFC 1700 [26]:

Dest. IP	Function
224.0.0.0	Base Address (Reserved)
224.0.0.1	**All Systems (Hosts) on this subnet**
224.0.0.2	**All Routers on this subnet**
224.0.0.3	Unassigned
224.0.0.4	DVMRP Routers
...	...
224.0.0.12	DHCP Server / Relay Agent
224.0.0.13	**All-PIM-Routers**
...	...

Note: RFC 1700 [26] explicitly states that

> *"... Multicast routers should not forward any multicast datagram with destination addresses in this range, regardless of its TTL."*

2.7.2.2 TTL Scoping

The Time To Live (TTL) field in the IP packet header is traditionally used to limit the lifetime of a datagram but in the context of multicast is also used to restrict multicast traffic to a certain region or site.

Each time an IP packet is forwarded by a router the TTL field in the IP header is decrement by one. This mechanism also applies to multicast and is accomplished by setting the TTL threshold of a routers multicast interface to a certain value depicted in the following table.

TTL Scope	TTL threshold	assigned Address range	Description
Node	0		The datagram is restricted to the local host and will not be sent onto any network interface
Link-Local (LAN)	1	224.0.0.0 - 224.0.0.255	datagram will be restricted to the hosts subnet and will not be forwarded by any router attached to that subnet
Department (Site)	< 32	239.255.0.0 - 239.255.255.255	Restricted to a department within an organization
Organization (Region)	< 64	239.192.0.0 - 239.195.255.255	Restricted to an organization
Global (World)	< 255	224.0.1.0 - 238.255.255.255	Not restricted. Used for global multicast sessions

2.7.2.3 Administratively Scoped Multicast Addresses

Administratively scoped addresses for IP Multicast are defined in RFC 2365 [27] which is similar to the scoping of unicast “private address space” (e.g. 192.168/16) defined in RFC 1918 [28].

The address space that can be used within a private multicast domain is:

239.0.0.0 - 239.255.255.255 (prefix: 239/8)

A multicast router must not forward traffic for these multicast groups out of its private multicast domain. Therefore a multicast BR has to apply filters on its multicast interfaces in order to prevent multicast traffic from entering or leaving the multicast domain or AS (see section 5.1). An example BR configuration to accomplish this can be found in the Appendix.

2.7.2.4 Administratively Scoped Boundaries

Administratively scoped boundaries is a mechanism that uses the “Administratively Scoped Multicast Addresses” defined in section 2.7.2.3 to prevent multicast traffic from leaving a certain region which is similar to TTL scoping (see section 2.7.2.2)

The difference is that the scope of multicast traffic in the case of “Administratively Scoped Boundaries” is dependent on the multicast destination address and not on the TTL field in the IP header.

By assigning a specific administratively scoped boundary on a router interface it is possible to restrict traffic from leaving and entering a specific region. The following example shows two nested administratively scoped boundaries.

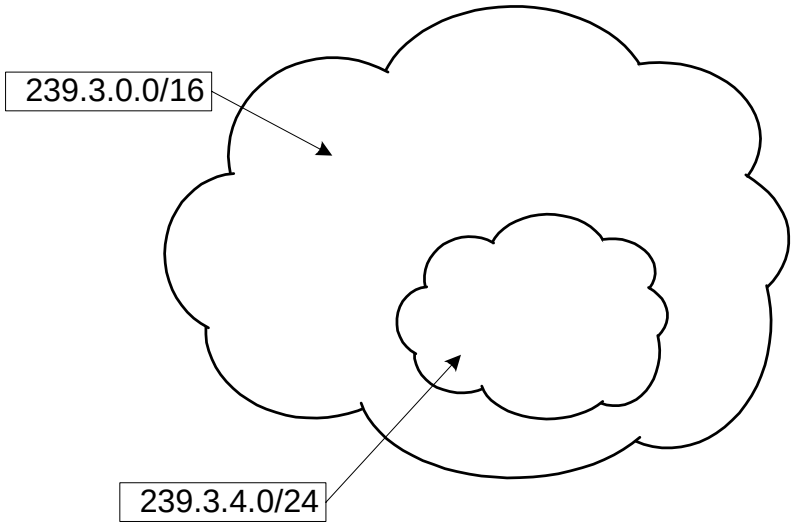

Figure 6: Example of administratively scoped boundaries

The outer boundary restricts multicast traffic in the range 239.3.0.0/16 from leaving or entering the region whereas the inner boundary restrict traffic from leaving or entering multicast traffic in the range 239.3.4.0/24.

Note: The address range 239.3.4.0/24 in the example above could be used more than once within the same outer administratively assigned boundary.

2.7.3 Multicast Group IP to MAC address mapping

In order to provide a full IP Multicast implementation Deering [9] also described “Extensions to an Ethernet Local Network Module”

> *“The Ethernet directly supports the sending of local multicast packets by allowing multicast addresses in the destination field of Ethernet packets. All that is needed to support the sending of multicast IP datagrams is a procedure for mapping IP host group addresses to Ethernet multicast addresses.*
>
> *An IP host group address is mapped to an Ethernet multicast address by placing the low-order 23-bits of the IP address into the low-order 23 bits of the Ethernet multicast address 01-00-5E-00-00-00 (hex). Because there are 28 significant bits in an IP host group address, more than one host group address may map to the same Ethernet multicast address.”*

The Ethernet address range used for multicast is defined to be in the range 00:00:5E:00:00:00 to 00:00:5E:7F:FF:FF but the 802.3 Ethernet standard of the Institute of Electrical and Electronics Engineers (IEEE) defined Bit 0 of Octet 0 of an Ethernet frame to have a special function. This Bit indicates if the frame is a broadcast or multicast frame depending on the destination MAC address as shown in the following figure:

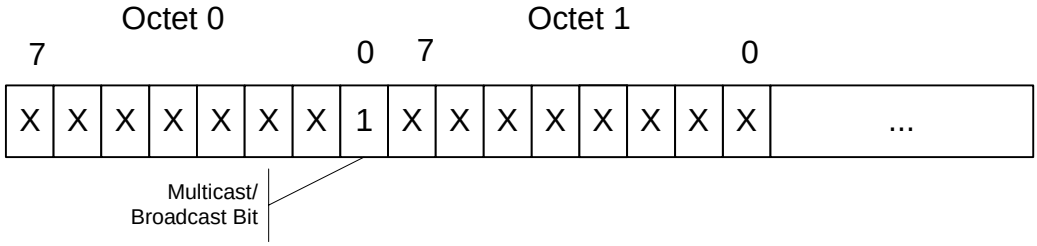

Figure 7: IEEE MAC frame header spec (first 2 octets)

- If Bit 0 of Octet 0 is set (1) and the destination MAC address is FF:FF:FF:FF:FF:FF (the MAC layer broadcast address) then the frame is destined to all hosts on a Ethernet segment
- If Bit 0 of Octet 0 is set (1) and the first three octets of the destination MAC address are 01:00:5E then the frame is destined to a group of multicast hosts on a Ethernet segment

The MAC address space available for IP multicast is in the range **01:00:5E:00:00:00 to 01:00:5E:7F:FF:FF** because of two reasons:

1) The first bit of the first octet has to be set to 1 for multicast/broadcast hence the address space starts with **01**
2) the first bit following the prefix 01:00:5E **must** be set to **0** hence the following octet is **7F** (01111111 → all remaining bits set to 1)

The definition of separate **multicast MAC frames** is very useful to limit the performance impact on multicast hosts. A hosts Network Interface Cards (NICs) can be "programmed" to filter MAC frames for specific destination MAC addresses. E.g. by default every MAC frame with a destination MAC of FF:FF:FF:FF:FF:FF (broadcast) is passed to the IP stack for further processing. Therefore, every time the NIC receives a frame with a destination MAC that should be received by the host it interrupts the currently running process for further processing of the received frame. If this is a broadcast frame then it is usually necessary to perform further processing of the frame. On the other hand if the frame is received by the host but not destined to it then (depending on the number of frames that erroneously have to be processed) this could have a severe performance impact on a host (CPU utilization is high).

Therefore, if a host wants to receive multicast traffic for a particular multicast group it must be able to "program" its NIC to receive only frames that the host has explicitly requested. These request are normally generated on the application layer. E.g. an application wants to receive multicast traffic for multicast group 224.2.2.2. In order to "program" the NIC to receive MAC frames for this group the multicast group address has to be mapped to a MAC address. Because a multicast application should not be concerned about this task this function is usually implemented by the NIC device driver and described in the following paragraph.

As Deering states [9], IP hosts group addresses are mapped to Ethernet multicast addresses by placing the low-order 23-bits of the IP address into the low-order 23 bits of the Ethernet multicast address 01:00:5E:00:00:00.

The problem is that there are 28 significant bits in an IP host group address hence there are 5 bits that are “lost” by the mapping process shown in Figure 8.

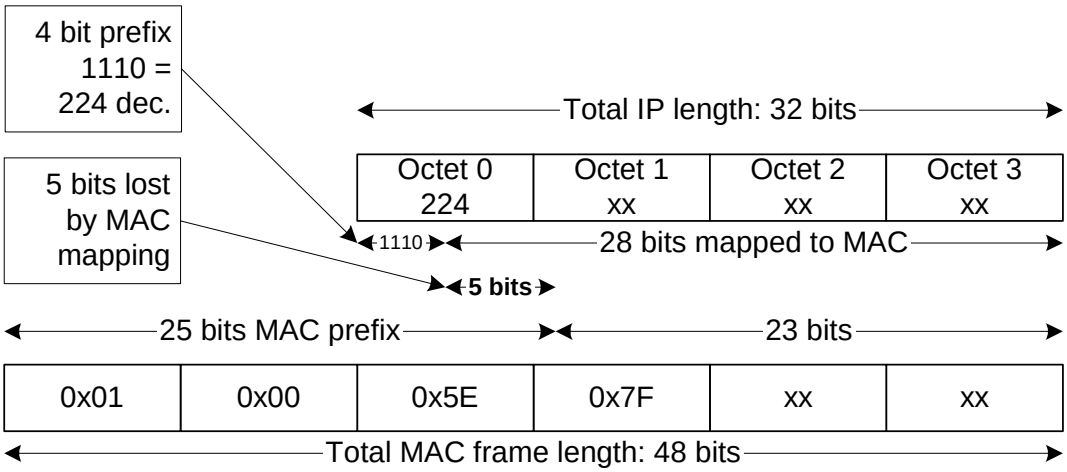

Figure 8: IP multicast group address to MAC address mapping

Because the 5 bits from the IP layer cannot be mapped into a destination MAC address 2^5 = 32 different IP group addresses map to the **same** MAC address. E.g. the IP group address 224.1.2.3 maps to the same MAC address as IP group address 229.1.2.3 as shown in the following example:

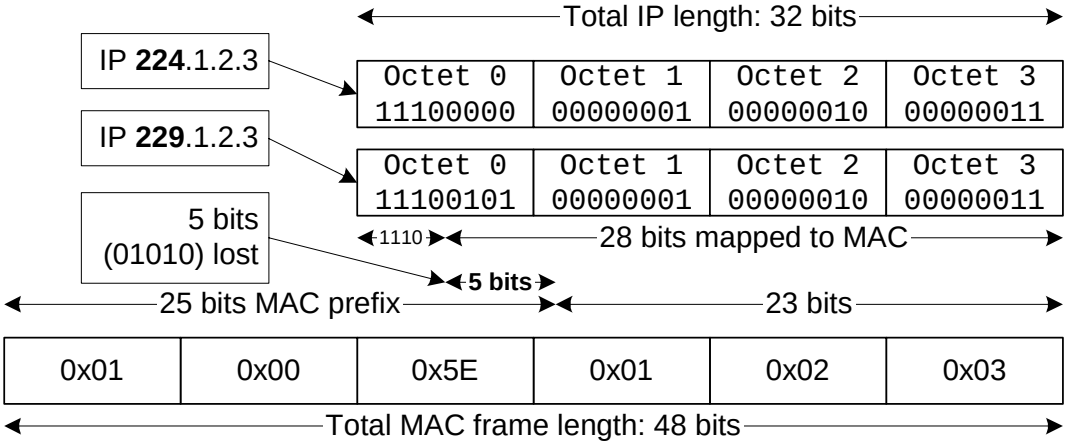

Figure 9: Ambiguous IP group address to MAC address mapping

This **MAC address ambiguity** could have negative effects on the multicast forwarding behavior and performance of a Layer 2 switch despite enabled multicast containment as described in section 6.3.

2.7.4 GLOP assigned IP to MAC address mapping

Since Deering described the method of Layer 2 multicast (see section 2.7.3) more and more hosts in the Internet started to use multicast. **Dynamic** multicast address allocation mechanism like the Session Directory Revised (SDR) [29] in conjunction with the Session Announcement Protocol (SAP) [30] / Session Description Protocol (SDP) [31] have been used traditionally. But more and more service or content provides needed **statically** allocated multicast addresses e.g. to broadcast live content like a radio station that also has its specific radio frequency.

Because a global multicast address allocation scheme in the Internet didn't exist Meyer and Lothberg wrote an Internet-Draft (now RFC 2770 [32]) that suggested to assign each AS it own static multicast address space based on their own Autonomous System Number (ASN). This multicast address space had to be in a range that is globally routed throughout the Internet. IANA therefore assigned the subnet **233/8** out of the class D address space for this "experiment" which was an addition to the allocation scheme defined in RFC 2365 [27].

RFC 2770 defines how a particular ASN (e.g. 5678) is mapped into the reserved IP multicast addresses range 233/8 (see Figure 10) which allows a single /24 network per AS:

1) Transform the ASN (dec) 5678 into (bin) 00010110 00101110 (left padded with 0s) the high order octet = (bin) 00010110 = (dec) **22** and the low order octet = 00101110 = (dec) **46**
2) Map the high order octet to the second octet of the reserved IP multicast address space
3) Map the low order octet to the third octet of the reserved IP multicast address space
4) The resulting multicast address space for ASN 5678 is **233.22.46.0/24**

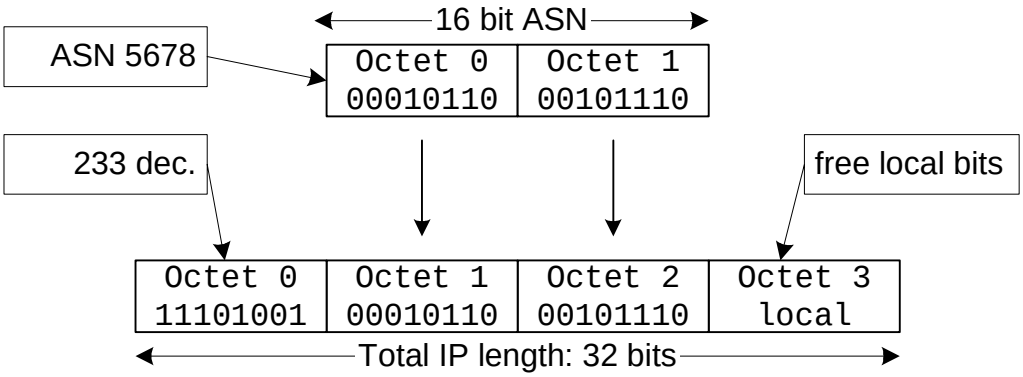

Figure 10: GLOP assigned ASN to multicast IP range mapping

The mapping mechanism of an IP multicast group address to a destination MAC address has already been described in section 2.7.3 and applies to the mapping of a GLOP assigned multicast address to a destination MAC address in the same way.

Based on the example of a GLOP assigned IP to MAC mapping by Eubanks [33] the above ASN 5678 would be mapped as follows:

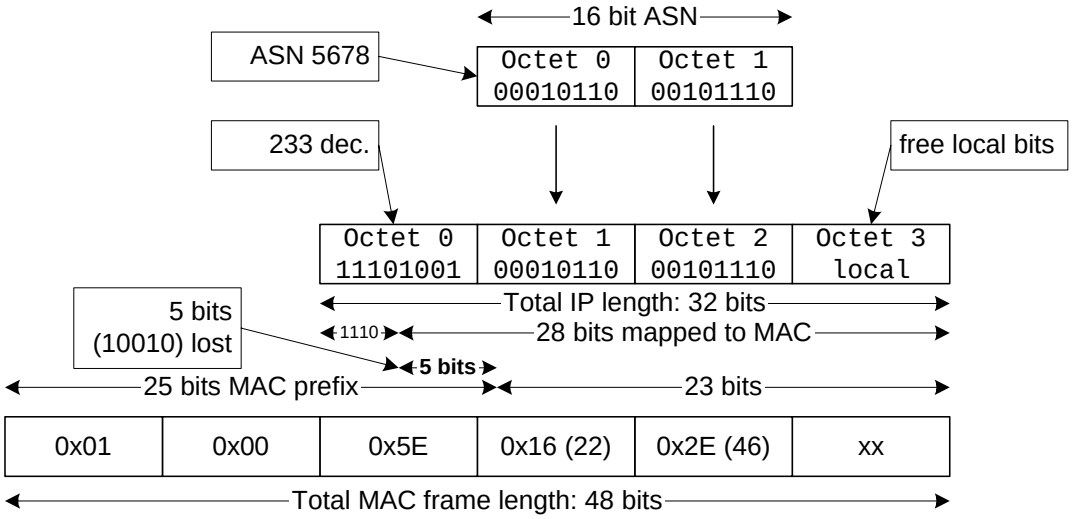

Figure 11: GLOP assigned multicast IP to MAC address mapping

Note: An online GLOP calculator is provide by the University of Oregon [34].

3 Multicast Distribution Trees

The main concept of multicast is to create and maintain multicast distribution trees. A **multicast distribution tree** is a loop free, tree based topology of interconnected routers that is created and maintained by multicast routing protocols (see chapter 4). All routers along the distribution tree replicate multicast packets as needed (towards downstream receivers).

Routing protocols like the Distance Vector Multicast Routing Protocol (DVMRP) [20] create a **flat** routing topology whereas other protocols like PIM (see section 5.2.2) create **hierarchical** routing topologies. In a flat routing topology all routers in a network have to store the whole multicast routing table of all interconnected routers in order to make forwarding decisions. In a hierarchical topology a router only knows information on how to forward traffic towards adjacent routers.

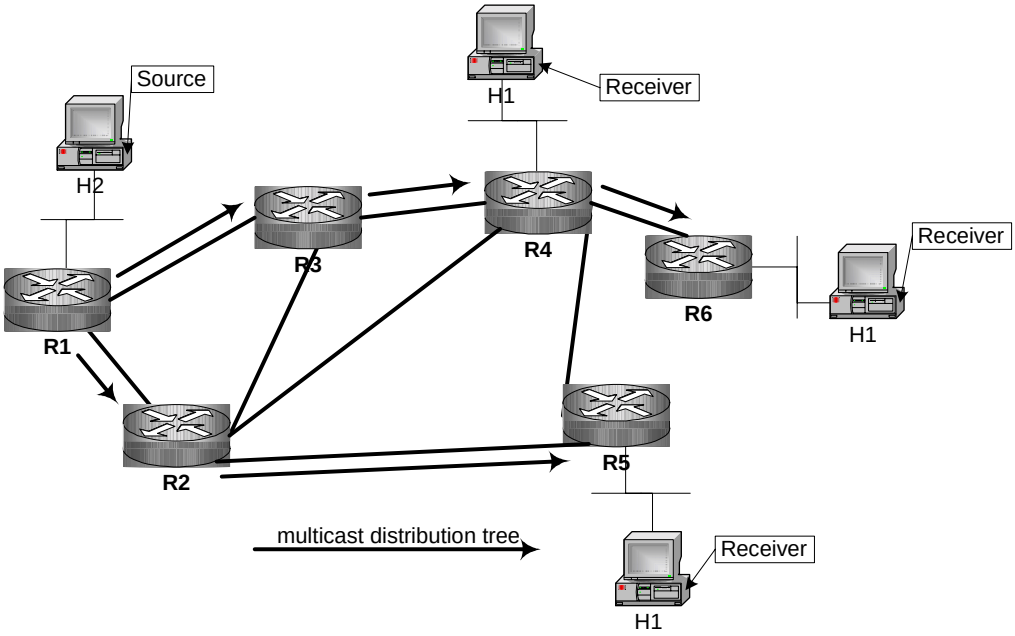

Figure 12: Example of a multicast distribution tree

There are two basic types of distribution trees that are support by current multicast routing protocols like PIM-SM (see section 4.3) described in the following two sections.

3.1 Shortest-Path Tree

A shortest-path tree (SPT) is the most efficient form of a multicast distribution tree because it uses a **single** spanning tree with the root of the tree at the source reaching all receivers throughout the network. This kind of distribution tree is also called “source tree” because of the fact that its root is at the **source** of the multicast traffic (the sender).

To create SPTs every router in the network has to keep **state information** about

- The source IP address (**S**) that is sending the traffic
- The multicast group address (**G**) it is sending to
- The incoming interface for a combined **(S,G) pair** entry
- The outgoing interface (**oif**) **list** for this particular entry

An (S,G) pair is the combination of the sources IP address and the multicast group address. E.g. if a source with IP address 141.28.2.1 is sending to a multicast group address 224.2.2.2 the (S,G) representation would be (141.28.2.1, 224.2.2.2).

It is very important to note that SPTs are **unidirectional** trees. Once the SPT is established traffic flows only **down** the SPT and never up the tree as depicted in Figure 13. Therefore, if another source (e.g. 141.28.2.20) wants to send to the same multicast group as in the example above (224.2.2.2) a separate SPT and hence (S,G) state is created in all routers/switches along the SPT.

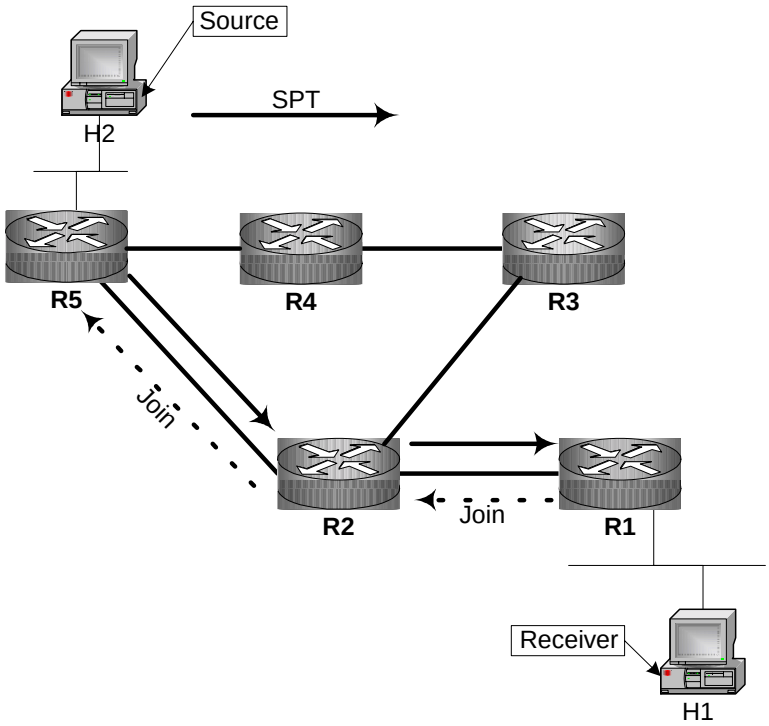

Figure 13: Example of a shortest path tree (SPT)

In the example above H1 wants to receive traffic from source H2. It therefore sends an IGMP report to request the reception of multicast group. The first hop router of H1 receives the IGMP report and send a PIM (S,G) Join message directly towards the source. Each router along the path towards the source repeats this process and creates (S,G) state entry for (141.28.2.1, 224.2.2.2). The PIM (S,G) Join message finally arrives at the sources DR (R5) and from that point on traffic flows down the SPT along R5, R2, R1 towards H1.

3.2 Shared Tree

A shared tree (ST) has its root in one particular point (sometimes more than one, see section 4.3) within the multicast network. This point (consisting of a specially configured router) is called a *rendezvous point* (RP) because senders and receivers meet at that point and “share” the root of the multicast distribution tree. Another common name for a shared-tree is core-based tree (CBT) (because they are routed at the core) or rendezvous point tree (RPT).

In a ST all multicast sources are sending their traffic to an RP and from there the multicast traffic is forwarded **down** the tree to all receivers. To represent that information as a forwarding state the notation (*,G) (pronounced “star comma G”) has been chosen to show that **every** multicast source (*) can send traffic to a specific multicast group (G) using a common shared tree.

Shared trees can either be unidirectional or bi-directional based on the routing protocol. In a unidirectional shared tree traffic can only flow **down** the tree to all receivers of the group whereas in a bi-directional shared tree traffic can flow **up and down** the multicast distribution tree.

Note: In the case of a unidirectional tree where the traffic can just flow **down** the tree (from the root to receivers) the traffic originated by a source has to get to the root somehow before it can flow down the shared tree. To accomplish this, the RP (root) creates an SPT to the source of the traffic in order to pull the traffic to the root. From there the traffic flows down the ST towards the receivers.

Figure 14 shows Host 2 with IP 141.28.2.1 sending to multicast group 224.2.2.2. H1 knows about this source from the RP (R3) and sends an IGMP report onto the local subnet. This report is receive by router R1 which creates a (*,G) entry for this group (*,224.2.2.2) and send another (*,G) shared-tree Join towards the RP. All routers along the path towards the **RP** perform the same task as R1 therefore creating a **shared-tree** between R1 and R3 represented by the (*,G) entry (*,224.2.2.2). Now traffic flows on the SPT from H2 to the RP and then on the ST down to the receiver H1.

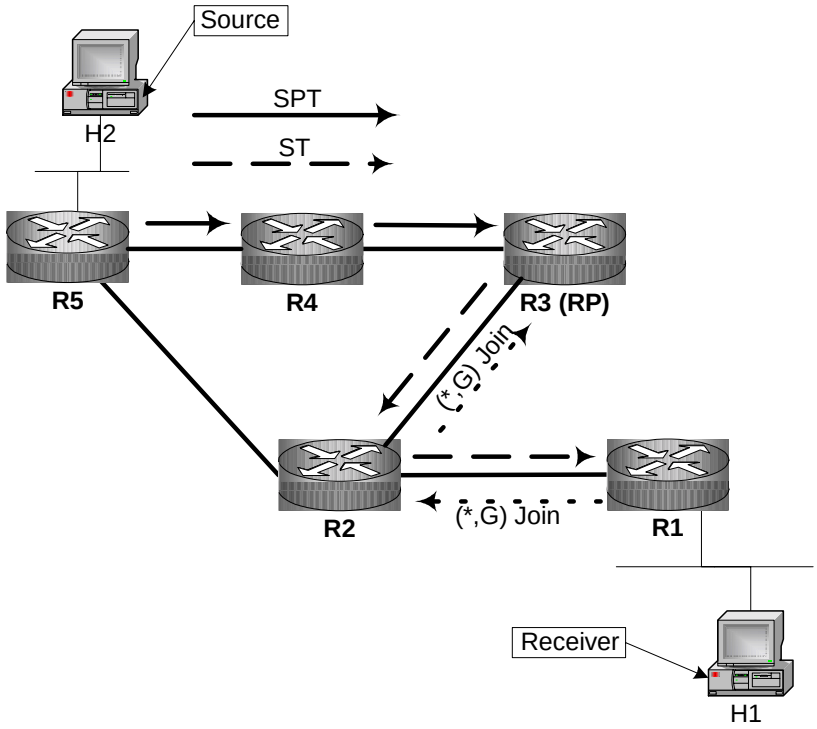

Figure 14: Example of a shared tree (ST)

4 Multicast Routing Protocols

IP Multicast routing protocols are used to

- create and maintain multicast distribution trees (see chapter 3) and
- forward IP multicast packets within these trees

The currently available multicast routing protocols can be divided into the following categories:

- Link-state protocols (e.g. DVMRP [20] / MOSPF [37])
- Dense Mode Protocols (see section 4.2) and
- Sparse Mode Protocols (see section 4.3)

These protocols are the basis of multicast in the intra- and interdomain as described in the following sections.

4.1 Reverse Path Forwarding

“Reverse Path Forwarding” (RPF), first described by Dalal [35] is a mechanism that is used by routers to decide whether or not to forward a packet that is received on a particular **incoming interface.** Multicast routing protocols like DVMRP [20] or PIM (see section 4.3) use this mechanism.

Every multicast packet that is received by a router has to pass an “RPF check” which is performed as follows:

If a packet is received on an interface that would be used to send the same packet back towards the source IP address in the IP header (where the packet originated) then it is on the “reverse path” and it will be forwarded to all interfaces in the outgoing interface (oif) list, otherwise it will be discarded.

The following example in Figure 15 shows the two cases for 1) RPF check fails (packet is dropped) or 2) RPF check succeeds (packet is forwarded):

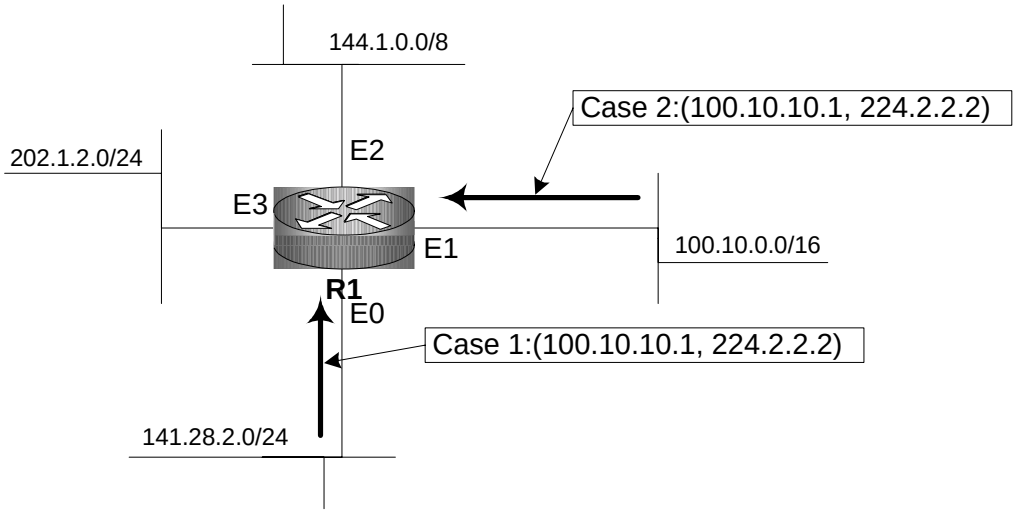

Figure 15: Reverse Path Forwarding example

Network	Interface
100.10.0.0/16	E1
141.28.2.0/24	E0
144.1.0.0/8	E2
202.1.2.0/24	E3

Multicast Routing table for R1

Multicast Group	Oif
224.2.2.2	E3, E0
224.2.2.3	E0
225.2.2.2	E0

Outgoing Interface (oif) list

Case 1: (RPF check fails → packet is dropped)

- Router R1 receives a multicast packet via interface **E0** with a source IP address of 100.10.10.1 and multicast group address 224.2.2.2
- The router uses the packets source IP address (100.10.10.1) in order to check if it was received on the reverse path back to the source (via the multicast routing table)

- The RPF check **fails** because the packet was **not** received on the interface that is on the reverse path back to the source as indicated by the multicast routing table. The packet was received on interface E0 that is on the reverse path back to network 141.28.2.0/24 and not 100.10.0.0/16 which is interface E1
- The packet is dropped

Case 2: (RPF check succeeds → packet is forward onto all interface in the oif-list)

- Router R1 receives a multicast packet via interface **E1** with a source IP address of 100.10.10.1 and multicast group address 224.2.2.2
- The router uses the packets source IP address (100.10.10.1) in order to check if it was received on the reverse path back to the source (via the multicast routing table)
- The RPF check **succeeds** because the packet was received on the interface that is on the reverse path back to the source as indicated by the multicast routing table
- The packet is forwarded **only** onto the ports indicated in the outgoing interface list (E3, E0) for the specified multicast group (224.2.2.2)

A PIM router uses the information in the unicast routing table to perform an RPF check (determine the incoming interface for a particular source). It does so by searching the routing table for the longest match of the source IP address in the multicast packet. If there are multiple entries for the same network in the routing table with equal cost paths then the interface with the highest next-hop IP address is chosen as the incoming interface.

According to Williamson [13] the following important rule applies:

> *"A router can have only ONE incoming interface for any entry in its multicast routing table"*

The following table shows the routing table of router R1.

Note: There are two route entries for network 100.10.0.0/16 with the same metric. The incoming interface on router R1 for a source in network 100.10.0.0/16 would be E0 because the interface with the highest next-hop IP address is used for the RPF check.

Network	Interface	Metric	Next hop
100.10.0.0/16	E1	4	141.2.2.1
100.10.0.0/16	E0	4	141.2.2.2 ←
144.1.0.0/8	E2	2	141.2.2.1
202.1.2.0/24	E3	5	141.2.2.3

Multicast Routing table of router R1 with associated metric

The following table shows a summary of Multicast Routing Options [36]:

Protocol	Unicast Protocol Requirements	Flooding Algorithm
PIM-dense mode [37]	Any	Reverse path flooding (RPF)
PIM-sparse mode [39]	Any	RPF
DVMRP [20]	Internal, RIP-like routing protocol	RPF
MOSPF [37]	Open Shortest Path First (OSPF)	Shortest-path first (SPF)

4.2 Protocol Independent Multicast – Dense Mode

Protocol Independent Multicast – Dense Mode [38] is (as the word "Dense" implies) based on the assumption that the members (receivers) of multicast traffic are "densely" populated within a given domain and therefore almost every host within a given domain wants to receive multicast traffic destined to a specific group.

"Dense Mode" protocols use a "**flood and prune**" mechanism in order to create and maintain multicast distribution trees and therefore belongs to the group of "flood and prune protocols".

Other dense-mode protocols like DVMRP and Multicast OSPF (MOSPF) are also designed for situations where multicast groups are widely represented and bandwidth is plentiful. With these schemes, data packets and/or membership report information may be sent out unnecessarily on interfaces that don't lead to multicast sources or interested receivers; additionally, routers store the associated state for these uninterested nodes, which is also unnecessary. This overhead is acceptable when most hosts are interested in the data and there is enough bandwidth to support the flow of control messages, but is otherwise inefficient.

"Flood-and-prune" protocols basically work as follows:

If a router receives multicast traffic it first **floods** it out of every router interface towards upstream routers. If an upstream router has no information about at least one receiver for this multicast group it sends a **prune** message back (out from the same interface where it received the multicast traffic) to the directly connected downstream router; otherwise the traffic is forwarded.

This behavior is the main disadvantage of "flood and prune" protocols like PIM-DM because they initially create unnecessary multicast traffic (and state information) until the upstream router prunes that branch of the multicast distribution tree back and stops the reception of multicast traffic.

4.3 Protocol Independent Multicast – Sparse Mode

Protocol Independent Multicast – Sparse Mode is defined in RFC 2362 [39]. The words "Protocol Independent" was used

> *"…because it is not dependent on any particular unicast routing protocol, and because it is designed to support sparse groups…".*

Multicast routing protocols like DVMRP [20] build their own routing tables in order to perform forwarding of multicast packets whereas all PIM protocols (PIM-SM/DM) are able to use any other underlying unicast routing protocol like OSPF [40], MBGP [41] (see section 5.2.1), RIP [42] or static routes injected in these routing tables.

PIM-SM belongs to the category of "Explicit Join Protocols" because they use a mechanism where routers have to **explicitly join** a multicast group by sending a join group message to the upstream router in order to receive multicast traffic from a specific group.

The PIM-SM protocol functionality can be summarized as follows [43]:

> *"Currently, PIM-SM is the de facto standard multicast routing protocol. It is designed to perform efficiently in WANs, where multicast groups are sparsely distributed. It maintains the traditional IP multicast service model of receiver-initiated membership and supports both shared and shortest-path trees. PIM-SM is not dependent on a specific unicast routing protocol. Because it is a router-to-router protocol, all routers in the network must be upgraded to support PIM-SM version 2."*

Note: "Sparsely distributed" does **not** mean that there are just a few multicast group members (receivers) in the network it just means that the group members are sparsely distributed throughout the network.

PIM-SM was especially designed to meet the following criteria:

- to be routing protocol independent (see begin of this section)
- to maintain the traditional ASM service model (see section 2.4.1)
  → receiver-initiated multicast group membership
  → Receivers signal their first hop routers (DRs) the group for which they want to receive multicast traffic
  → source just start sending without signaling to the DR
- the protocol must support STs **and** SPTs concurrently (see below)
  → STs use a RP as the root of the ST
  → SPTs directly interconnect sources and receivers with a separate SPT for each source
- the ability to adapt to changing network conditions and the dynamics of group membership and multicast source changes
- hosts do not have to be upgraded to participate in a PIM-SM network but all intermediate routers (between source and receiver)

Because PIM-SM is an "explicit join protocol" it works on the assumption that **no** host wants to receive multicast traffic unless it is **explicitly requested** (by PIM Join messages) or the reception is **explicitly stopped** (by PIM Prune messages).

PIM Join/Prune messages (message format see section 4.3.5) are used to signal the request to join or leave a ST (with the root of the tree at the RP) or an SPT (with the root of the tree at the source). These messages are sent to the "ALL-PIM-ROUTERS" address (224.0.0.13) and travel hop-by-hop towards the source in order to create PIM state in the routers along the path. This state information of all routers represents the whole ST/SPT multicast distribution tree topology in a PIM domain.

RFC 2362 [39] defines the term of a "**PIM domain**" as

> *"… a contiguous set of routers that all implement PIM and are configured to operate within a common boundary defined by PIM Multicast Border Routers (PMBRs). PMBRs connect each PIM domain to the rest of the internet."*

In this document the term "domain" is used as a synonym for "PIM domain".

The following sections describe the PIM-SM operation in the **intradomain** (PIM-SM operation in the interdomain is described in section 5.2.2). All examples are based on the current PIM-SM Version 2 protocol implementation [39]. Protocol implementation details have been excluded because for the scope of this document it is just important to have a general understanding of the PIM-SM distribution tree creation and state maintenance.

4.3.1 Rendezvous Point Election and Function

PIM-SM supports STs and SPTs. SPTs are routed at the source whereas STs are routed at a single common root (a specifically configured router) within the domain. This common root has to be defined before any ST can be established and is called rendezvous point (RP) (see section 3.2).

There are two basic mechanisms to define which router in a network should have the function of the RP. The first is to **statically configure** the IP of the RP in all routers in a domain in order to send ST Join messages (PIM (*,G) Joins) towards the RP. From a network management standpoint it is obvious that this is not a very elegant solution. The PIM-SMv2 protocol spec [39] provides a method called the "Bootstrap Router Mechanism" whereas Cisco implemented a mechanism called "Auto-RP" [13]. Both of these methods **dynamically assign** multicast group to RP mappings to all routers in a domain.

RPs are also used within a domain to discover sources because DRs in the ASM service model register sources via PIM register messages (see section 4.3.4).

Traditionally sources have been discovered using a multicast application called Session Directory Revised (SDR) [29] which uses the Session Announcement Protocol (SAP) [30] / Session Description Protocol (SDP) [31] to provide a list of active multicast sources and was also used to announce multicast sessions and therefore "reserve" a multicast host group address for a specific session. These source discovery applications and protocols are still used.

Note: The SSM service model does not support shared-trees and therefore an RP in a domain does not know about active SSM source within its domain. In the SSM service model sources are discovered mostly via websites.

4.3.2 Shared Tree Join Procedure

Figure 16 shows a basic example of a receiver joining a PIM-SM ST (shared-tree).

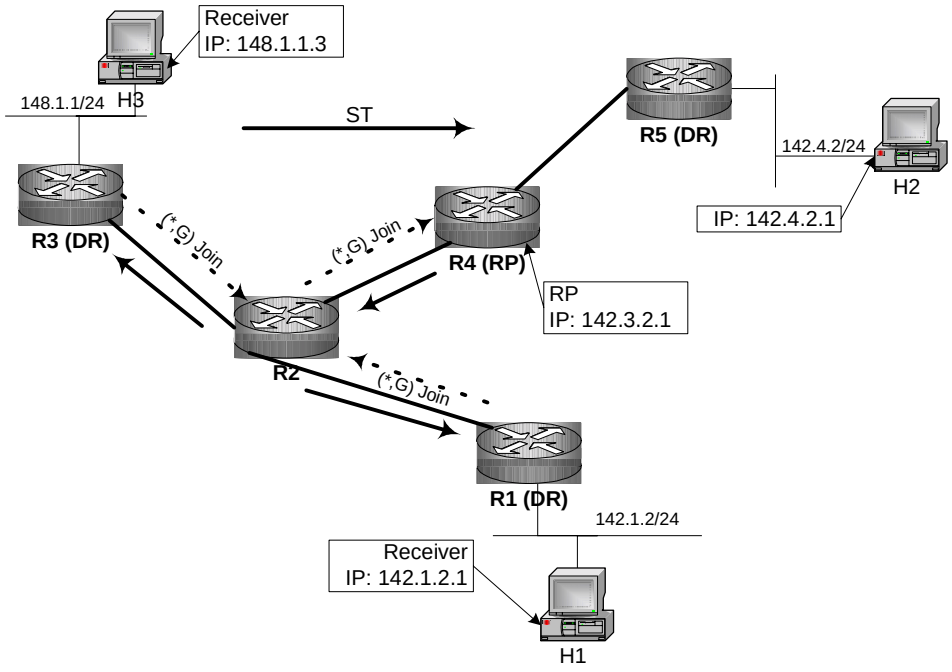

Figure 16: PIM shared-tree (ST) join procedure

R3 is the DR of subnet 148.1.1/24 (see chapter 2.6.1). H3 wants to receive multicast traffic for group 224.2.2.2 therefore it multicasts an IGMP membership report (destination IP: 224.2.2.2) on the subnet which is received by R3.

R3 creates a (*,G) entry for multicast group 224.2.2.2 → (*, 224.2.2.2) and puts the downstream interface (towards the receiver) in the outgoing interface (oif) list of the (*,G) entry. Then R3 sends a **PIM (*,G) Join** message upstream **towards** the **RP** (R4) which is in this case R2.

R2 in turn performs the same task as R3 and sends a (*,G) upstream towards the RP. Finally the (*,G) join arrives at the RP which also creates a (*,G) entry for group 224.2.2.2 and adds the link to R2 in the oif list.

At this point multicast traffic for group 224.2.2.2 can now flow down the ST towards H1.

If another host wants to join the ST like e.g. H1, R1 receives the IGMP report, creates a (*,G) and adds the interface towards H1 to it. Then R1 sends a (*,G) join to R2 (towards the RP). R2 already has a (*,G) entry for group 224.2.2.2 and simply adds the interface towards R3 to the oif list of the (*,G) entry for group 224.2.2.2. All subsequent traffic will flow from the RP down the ST towards H3 and H1.

Assume that H3 decides to stop receiving traffic for group 224.2.2.2. It sends an IGMP leave message which is received by R3. If this is the last host that leaves the group (checked by an IGMP group specific query), R3 removes the downstream interface from its oif list of the corresponding (*,G) entry and sends a **PIM (*,G) Prune** message to R2 (towards the RP) in order to inform its upstream routers that it has to stop forwarding traffic. If R2 removes the last entry from the oif list of the (*,G) entry it sends a PIM (*,G) Prune message towards the RP to prune itself off the ST.

4.3.3 Shortest-Path Tree Join Procedure

Figure 17 shows a basic example of a receiver joining a PIM-SM SPT (shortest-path tree) based on the example shown in Figure 16.

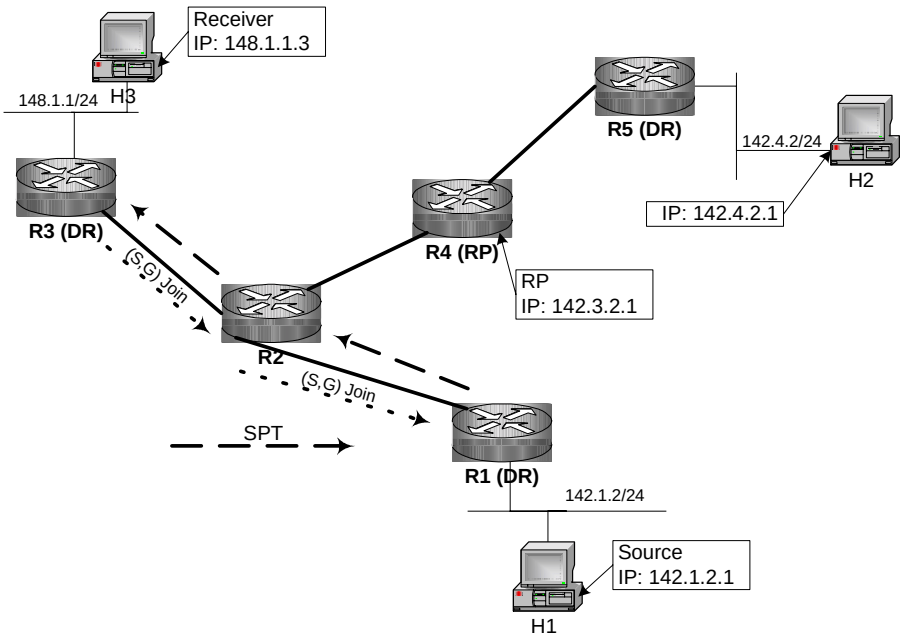

Figure 17: PIM shortest-path tree (SPT) join procedure

R1, R2, R3 and the RP (R4) currently belong to the ST for group 224.2.2.2. PIM provides a mechanism to switch from an ST to an SPT in order to optimize the distribution tree and therefore improve multicast forwarding. This mechanism is called “SPT switchover” and is performed by the receivers first-hop-router if the amount of received multicast traffic exceeds a certain threshold. Cisco routers by default perform the SPT switchover as soon as the first multicast packet from the source arrives at the receivers first-hop-router (DR).

Assume that R3 decides to join the SPT towards the source H1. R3 first creates an (S,G) entry → (142.1.2.1, 224.2.2.2) and adds the downstream interface to the oif list of that (S,G) entry. Then R3 sends a **PIM (S,G) Join** message **towards** the **source** H1. R3 determines the interface that should be used to send the (S,G) Join by checking which interface would be the RPF interface (see section 4.1) towards the source (based on the source address (S)) and sends the (S,G) Join.

When the join arrives at R2 it creates an (S,G) entry and adds the interface where the join was received to the oif list of the (S,G) entry. R2 in turn also performs an RPF interface calculation based on the source address (S) and sends an (S,G) Join towards this next hop (R1).

R1 receives the join, and adds the link to R2 to the already existing (S,G) entry. The (S,G) entry already existed because it was originally created as soon as the source started to send multicast traffic to group 224.2.2.2. The SPT is established and traffic now flows down the SPT from H1 to H3.

As in the example in section 4.3.2 we assume that H3 decides to stop receiving traffic for group 224.2.2.2 and that there are no other receivers for this (S,G) pair in the domain. H3 therefore sends an IGMP leave message which is received by R3. If this is the last host that leaves the group (checked by an IGMP group specific query), R3 removes the downstream interface from its oif list of the corresponding (S,G) entry and sends a **PIM (S,G) Prune** message to R2 in order to inform its upstream routers that they have to stop transmitting traffic. The (S,G) prune message is propagated down the SPT towards the source where it finally arrives and causes R1 to remove the oif list entry for this (S,G) pair.

4.3.4 PIM Register Messages

PIM-SM uses unidirectional STs where traffic can only flow **down** the ST towards receivers as described in section 3.2. According to the ASM service model sources can start sending traffic without signaling this information to any router in the network. The problem is how multicast traffic gets to the RP in the first place so that the RP can forward it down the ST to the receivers. PIM uses a special type of message called "register message" in order to [13]:

- Notify the RP that Source (S) is actively sending to group G
- Deliver the initial multicast packet(s) sent by the source (S) (each encapsulated inside of a single PIM register message) to the RP for delivery down the ST

Figure 18 shows how this mechanism is implemented.

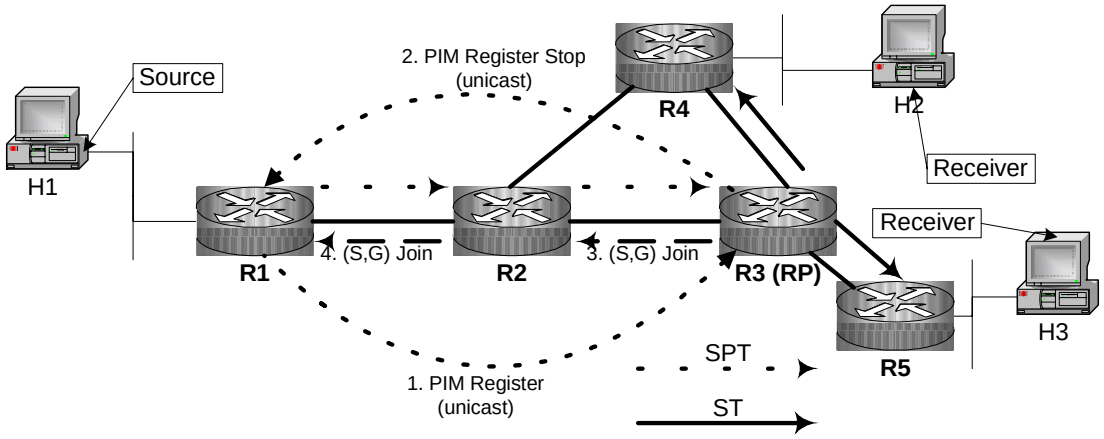

Figure 18: Source registration using PIM register messages

H1 is sending multicast traffic to group 224.2.2.2. The first hop DR of the subnet (R1) is encapsulating the multicast packets in PIM register messages and sends them as unicast packets to the RP. The RP receives the packets and decapsulates them. It now sees that these packets are destined to group 224.2.2.2. The RP checks if it has (*,G) forwarding state for this group. If there is no (*,G) state (no receivers for that group) the packets are discarded and a PIM register stop message is unicast back to R1 to stop sending unnecessary traffic.

On the other hand, if a (*,G) forwarding state for this group exists the RP starts sending the de-encapsulated multicast packets down the ST towards the receivers. Additionally the RP sends an (S,G) Join towards the source in order to create the SPT and to pull the multicast traffic natively down to the RP. As soon as the first multicast packets arrive via the SPT at the RP the RP unicasts a PIM register stop message to the DR (R1) because the traffic now flows over the SPT to the RP.

Note: The process of source registration via PIM register messages at the RP has an impact on the MSDP SA generation in a MSDP enabled router. The PIM register messages in the RP triggers MSDP SA messages to be sent to all MSDP peers (see section 5.2.3).

4.3.5 PIM Join/Prune Message Format

The PIM-SM spec RFC 2362 [39] describes PIM Join/Prune messages as follows:

> *"Join/Prune messages are sent to join or prune a branch off of the multicast distribution tree. A single message contains both a join and prune list, either one of which may be null. Each list contains a set of source addresses, indicating the source-specific trees or shared tree that the router wants to join or prune."*

All PIM control messages are (apart from Register and Register-Stop messages which are unicast) sent with an IP protocol number of **103** and multicast hop-by-hop to the "ALL-PIM-ROUTERS" group **224.0.0.13** (see section 2.7.2.1).

PIM-SM Join/Prune messages have the following format:

Bits 0 - 3	Bits 4 - 7	Bits 8 - 15	Bits 9 - 31
PIM Version	Type	Reserved	Checksum
Upstream Neighbor Address (Encoded-Unicast format)			
Reserved		Number of Groups	Holdtime
Multicast Group Address 1 (Encoded-Group format)			
Number of Joined Sources			Number of Pruned Sources
Joined Source Address 1 (Encoded-Source format)			
. . .			
Joined Source Address n (Encoded-Source format)			
Pruned Source Address 1 (Encoded-Source format)			
. . .			
Pruned Source Address n (Encoded-Source format)			
. . .			
Multicast Group Address m (Encoded-Group format)			
Number of Joined Sources			Number of Pruned Sources
Joined Source Address 1 (Encoded-Source format)			
. . .			
Joined Source Address n (Encoded-Source format)			
Pruned Source Address 1 (Encoded-Source format)			
. . .			
Pruned Source Address n (Encoded-Source format)			

Figure 19: PIM Join/Prune message format

The following table emphasizes on the most important fields in regard to this document. The full description (especially the format of “encoded” field types) can be found in RFC 2362 [39] chapter 4 “Packet Formats”:

Field	**Description**	**Possible Values**
PIM Version	PIM Version	2 = PIM Version 2
Type	PIM message type	e.g. 0=Hello / 3=Join/Prune
Encoded Upstream Neighbor Address	Encoded Address of the Upstream Neighbor where the message is sent to	e.g. 144.2.2.1
Holdtime	The amount of time a receiver must keep the Join/Prune state alive, in seconds. If the Holdtime is set to `0xffff', the receiver of this message never times out the oif (useful for ISDN lines) if the Holdtime is set to `0', the information is timed out immediately	3.5 * [Join/Prune-Period] Default: 210 seconds (see table below)
Number of Groups	The number of multicast group **sets** contained in the message	variable
Encoded-Multicast group address	The encoded multicast group address for which this messages is sent	e.g. 224.2.2.2
Number of Joined Sources	Number of join source addresses listed for a given group	variable
Join Source Address-1 … n	This list contains the sources that the sending router will forward multicast datagrams for if received on the interface this message is sent on	e.g. 166.2.2.1, 166.3.2.1, 166.3.2.2 …
Number of Pruned Sources	Number of prune source addresses listed for a group	variable

Prune Source Address-1 .. n	This list contains the sources that the sending router does not want to forward multicast datagrams for when received on the interface this message is sent on. If the Join/Prune message boundary exceeds the maximum packet size, then the join and prune lists for the same group must be included in the same packet	e.g. 168.2.2.1, 168.2.2.3, 169.2.1.2 …

Figure 19 shows that a PIM Join/Prune message can carry not just one Join or Prune request furthermore it can be a list of joins (**Number of Joined Sources**) and prunes (**Number of Pruned Sources**) within one message sent to an upstream router (**Upstream Neighbor Address**). This reduces the protocol overhead considerably. The processing of the list of joined/pruned sources is described in section 6.3.3.3.

4.3.6 Important Timers

The next table consists of an extract of important timers relevant to this document taken from the PIM-SM RFC 2362 [39]:

Timer	Description	Default value
[Join/Prune-Period]	This is the interval between sending Join/Prune messages.	60 seconds
[Join-Prune Holdtime]	This is the Holdtime specified in Join/Prune messages, and is used to time out oifs. This should be set to 3.5 * [Join/Prune-Period].	210 seconds
[Hello-Period]	Hello messages are sent periodically between PIM neighbors, every [Hello-Period] seconds. This informs routers what interfaces have PIM neighbors.	30 seconds
[Hello-Holdtime]	Hello messages are multicast using address 224.0.0.13 (ALL-PIM-ROUTERS group). The packet includes a Holdtime, set to [Hello-Holdtime], for neighbors to keep the Information valid.	105 seconds

Note: If e.g. a PIM neighbor router silently goes away “old” (*,G) or (S,G) PIM-SM forwarding state could exhaust a routers memory resources. Therefore PIM associates a default timeout value of [Join-Prune Holdtime] (see above) with each (*,G) or (S,G) entry. If this timer expires the state entry is automatically removed.

To prevent the expiration of these timers, and therefore deletion of PIM state entries, every PIM router has to periodically refresh the state information in the (appropriate) upstream PIM neighbor. It does so by sending PIM (*,G) or (S,G) Joins every [Join/Prune-Period] seconds for all state entries that have a non-empty oif list. This procedure ensures that any state information representing any existing STs or SPTs are kept "alive".

4.4 Problem of Keeping State Information

Every router in a PIM domain has to keep state information in order to build multicast distribution trees. This uses (depending on the multicast routing protocol) a lot of router resources (memory, CPU).

The PIM-SM RFC 2362 [39] states that a routers has to keep **route entries** that may include

> *"... such fields as the source address, the group address, the incoming interface from which packets are accepted, the list of outgoing interfaces to which packets are sent, timers, flag bits, etc."*

This shows that one route entry uses quite considerable amount of router memory. Multicast forwarding state is created each time a SPT or ST is created in the network. STs tend to use less resources because they are not "source specific" compared to SPTs. Therefore routing protocols that create STs seam to be preferred over routing protocols that solely relay on SPTs. However, STs are (in most cases) inefficient compared to SPTs because they only create sub-optimal paths in the distribution tree because there are routed at the RP.

An obvious problem of multicast distribution trees is that stat has to be created no matter what routing protocol is used. Because all (S,G) state information has to be stored in the memory of a router or switch a "state explosion" within the whole network could occur if for example a Denial of Service Attack (DoS) (see chapter 7) is targeted to a multicast enabled network.

Williamson [13] notes that creation of (S,G) state along the SPT consumes more router resources but …

> *"… the overall amount of (S,G) information maintained by routers in a PIM-SM network that uses SPTs is generally much less than is necessary for dense mode protocols. The reason is that the Flood-and-Prune mechanism used by dense mode protocols results in all routers in the network maintaining (S,G) state entries in their multicast routing tables for all active sources. This is true even if there are no active receivers for the groups to which the sources are transmitting. By joining SPTs in PIM-SM, we gain the advantage of an optimal distribution tree without suffering from the overhead and inefficiencies associated with other dense mode protocols such as PIM-DM, DVMRP, and MOSPF."*

The reasons why Explicit-Join-Protocols like PIM-SM should be preferred over "Flood-and-Prune" protocols are therefore clearly

- Less state information in routers
- Creation of optimal multicast distribution trees

5 Multicast Internet Exchange Point

5.1 General Topology

A MIX interconnects PIM Multicast Border-Routers (PMBRs) of Internet Service Providers (ISPs), Application Service Providers (ASPs), Content Providers or other network service providers each having their own Autonomous System Number (ASN) via a central Layer 2 switch fabric (**core switch**).

An Autonomous System (AS) is defined as

> *"… a connected group of one or more IP prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy."* [44]

The following figure shows how domains would interconnect their PMBRs without a core switch (Note that in the context of this document the term "domain" is used as defined in section 4.3.):

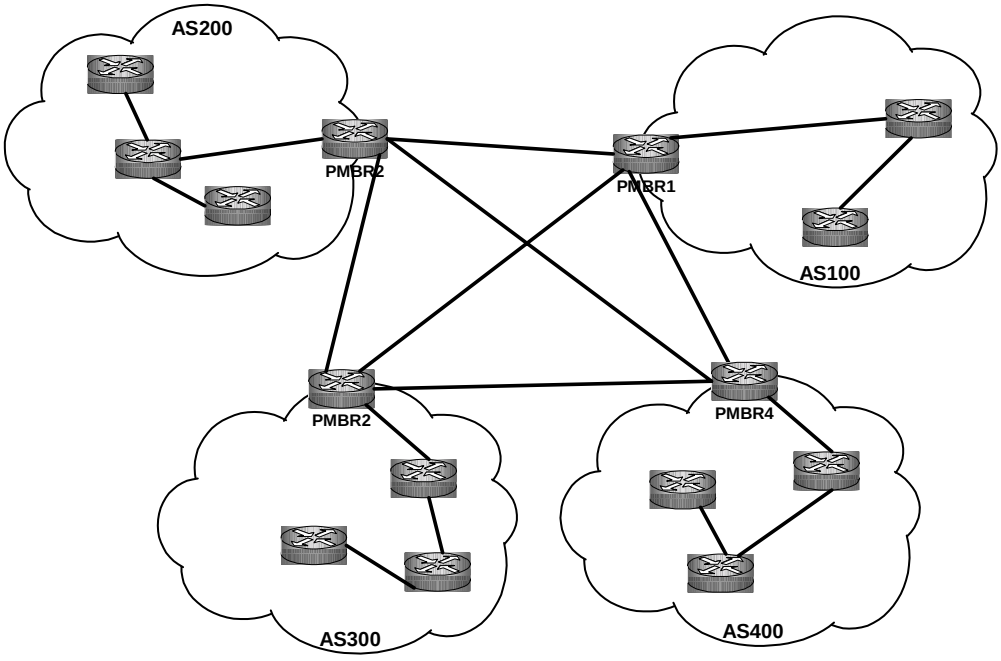

Figure 20: Directly interconnected domains

Figure 21 shows how domains would interconnect using a core switch at a MIX:

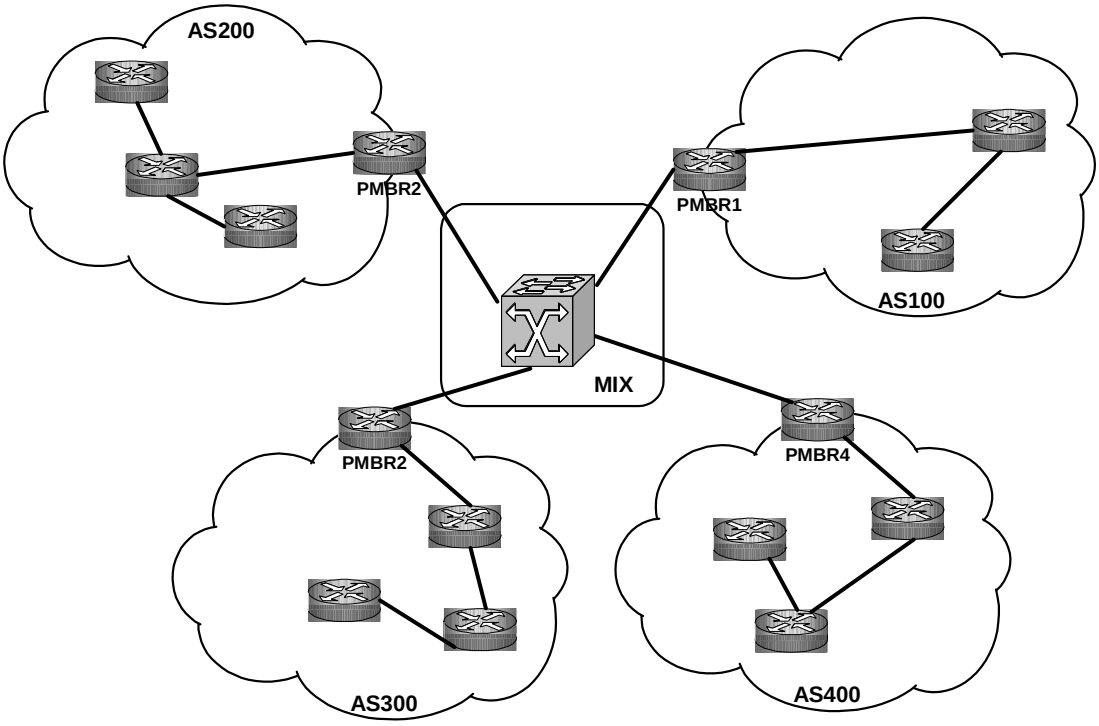

Figure 21: Domains interconnected via a core switch at a MIX

In the topology of directly interconnected PMBRs as seen in Figure 20 the amount of interconnections necessary to interconnect all domains in a full mesh is a quadratic function (**n * (n-1) / 2)**. E.g. if 5 domains want to be fully meshed together it would need (5*(5–1)/2) 10 direct interconnects (physical circuits). Additionally, every router in that mesh would have the burden of multicast packet replication.

In the case of a fully interconnect mesh via a core switch at a MIX the amount of interconnects needed would be **equal** to the number of connected domains and the burden of multicast packet replication (see section 3) is additionally left to the core switch.

This clearly shows that, if interdomain multicast traffic exchange is needed, there are significant benefits for all participants to interconnect their PMBRs at a MIX.

Note: Most of the currently existing MIXs or IXP provide a Layer 2 core switch infrastructure. Participants in the MIX are just allowed to connect Layer 3 devices (routers) to the exchange point and no Layer 1 or Layer 2 devices (hubs/Layer 2 switches).

5.2 Standard Protocols

According to La Master et al [45] all participants at a MIX have to agree upon the following prerequisites:

- the protocol to be used for multicast route exchange
  → MBGP (see section 5.2.1)
- the method for identifying active sources
  → MSDP (see section 5.2.3)
- the method for performing multicast forwarding
  → PIM-SM (see section 4.3)
- the physical medium for the multicast exchange
  → in this case an Ethernet switch fabric

The requirements stated above led to the protocols described in the following three sections and are the **de facto standard protocols** currently used to perform multicast traffic exchange at a MIX.

5.2.1 Multiprotocol Extensions for BGP-4

The "Multiprotocol Extensions for BGP-4" (MBGP) [41] (also known as BGP4+) defines an extension to the Border-Gateway Protocol Version 4 (BGP-4) [46]. In some documents MBGP is translated to "Multicast Border Gateway Protocol". This is somewhat misleading because MBGP is not only an extension to BGP-4 that is just able to carry routing information for IPv4, furthermore MBGP defines [41]

> *"… extensions to BGP-4 to enable it to carry routing information for multiple Network Layer protocols (e.g., IPv6, IPX, etc...). The extensions are backward compatible - a router that supports the extensions can interoperate with a router that doesn't support the extensions."*

So clearly MBGP is not just an extension to BGP-4 to carry multicast routing information it can be used to carry routing information for a variety of other network layer protocols. However, according to Williamson [13]

> *"… in multicast circles, the M of MBGP is understood to mean multicast and not multiprotocol."*

The objective of using MBGP at a MIX is to provide the necessary interdomain routing protocol information (between different domains) that is needed by PMBRs to perform **RPF checks** (see section 4.1). As already described in section 4.3 PIM is "protocol independent" and is therefore also able to use the BGP routing table for the RPF check.

The reason for using MBGP was that MIX participants (or ISPs that want to enable multicast within their domain using internal BGP (iBGP)) want to be able to establish **incongruent routes** for multicast and unicast traffic.

To accomplish these incongruent routes MBGP introduces two new Network Layer Reachability Information (NLRI) BGP attributes which are carried inside a BGP update messages to their respective BGP peers:

- MP_REACH_NLRI (Multiprotocol **Reachable** NLRI)
  "*… used to carry the set of reachable destinations together with the next hop information to be used for forwarding to these destinations"*
- MP_UNREACH_NRLI (Multiprotocol **Unreachable** NLRI)
  "*… used to carry the set of unreachable destinations.*"

MBGP is backward compatible to BGP-4 as stated in RFC 2858 [41]

> *"Both of these attributes are optional and non- transitive. This way a BGP speaker that doesn't support the multiprotocol capabilities will just ignore the information carried in these attributes, and will not pass it to other BGP speakers"*

These new attributes themselves consist of two fields that are used to identify the network layer protocol for which the Reachable- or Unreachable NLRI carries routes. These two fields are

- The Address Family Identifier (AFI)
  AFI = 1 → IPv4 (as defined by RFC 1700 [26])
- The Subsequent Address Family Identifier (SAFI)
  SAFI = 1 → NLRI information is used for unicast routing
  SAFI = 2 → NLRI is used for multicast routing
  SAFI = 3 → NLRI is used for unicast **and** multicast routing

The example below shows a configuration of incongruent routes between two domains.

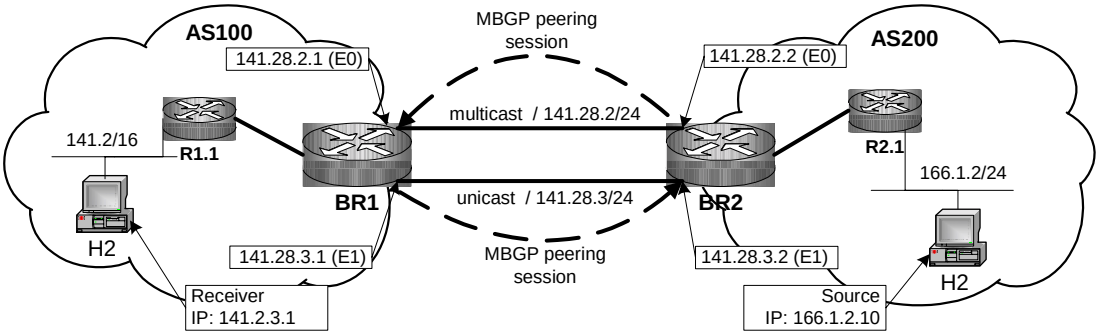

Figure 22: MBGP config with incongruent unicast / multicast routes

In the example above two PMBRs in two different domains are interconnected via two interfaces using the following IP addresses:

AS	BR	Interface	IP
100	BR1	E0	141.28.2.1/24
100	BR1	E1	141.28.3.1/24
200	BR2	E0	141.28.2.2/24
200	BR2	E1	141.28.3.2/24

The BRs of AS100 (R1) and AS200 (R2) have a BGP peering session established.

BR1 advertises that it wants to receive **multicast** traffic for one of its networks (prefix: 142.2/16) via interface E0. The BGP update message would be:

MP_REACH_NLRI:	142.2/16
AFI:	1
SAFI:	2 (NLRI is for **multicast** only)
AS_PATH:	100
NEXT_HOP:	**141.28.2.1**

BR1 also advertises that it wants to receive **unicast** traffic for network prefix 142.2/16 via interface E1:

MP_REACH_NLRI:	142.2/16
AFI:	1
SAFI:	1 (NLRI is for **unicast** only)
AS_PATH:	100
NEXT_HOP:	**141.28.3.1**

Note: The unicast route advertisement could have been sent with a NLRI attribute instead of a MP_REACH_NLRI attribute which represents the "old" BGP-4 attribute for unicast-only routes.

BR2 advertises that it wants (for whatever reason) to receive **unicast** and **multicast** traffic for network prefix 166.1.2/24 via interface (E0):

MP_REACH_NLRI:	166.1.2/24
AFI:	1
SAFI:	3 (NLRI is for **unicast** and **multicast**)
AS_PATH:	200
NEXT_HOP:	141.28.2.2

Using the example in Figure 22 where H2 is the sender of multicast traffic destined to multicast group 224.2.2.2 and H1 is a receiver. Both ASs are running PIM-SMv2 [39] as Interior Border Gateway Protocol (IBGP) and Exterior Border Gateway Protocol (EBGP). In order to create an SPT to pull the traffic from H2 in AS200 to H1 in AS100 BR1 has to send a PIM (S,G) Join towards the source in AS200. The BGP routing table in BR1 indicates that there is a route to network 166.1.2/24 via next hop 141.28.2.2 and therefore PIM should RPF to 141.28.2.2 for sources in this network. The subsequent RPF check succeeds and the PIM (S,G) Join is sent via BR1's interface **E0** because the MP_REACH_NLRI for network 166.1.2/24 indicates that IPv4 multicast/unicast traffic (AFI:1 / SAFI: 3) has a an AS_PATH of 200 and a next hop of 141.28.2.2.

Note: If BR1 would have to send **unicast** traffic towards AS 200 it would also use interface E0 because SAFI=3 in the MP_REACH_NLRI attribute indicates that the next hop 141.28.2.2 must be used for multicast **and unicast** traffic.

If a receiver in AS200 would join a multicast group in network 141.2/16 the subsequent PIM (S,G) Join would be sent via BR2's interface E0 and multicast traffic would flow via this link (according to BR2's BGP routing table).

Unicast traffic from AS200 towards AS100 would flow via BR2's interface E1 because its routing table indicates that MP_REACH_NLRI for IPv4 unicast traffic (AFI: 1 / SAFI: 1) has an AS_PATH of 100 and next hop 141.28.3.1.

5.2.2 Interdomain Multicasting using PIM-SM

Based on the description of the PIM-SM operation in section 4.3 this chapter describes the multicast distribution tree construction using PIM-SM Version 2 in the **interdomain**.

Three PMBRs are interconnected via a Layer 1 or Layer 2 network device shown in the figure below:

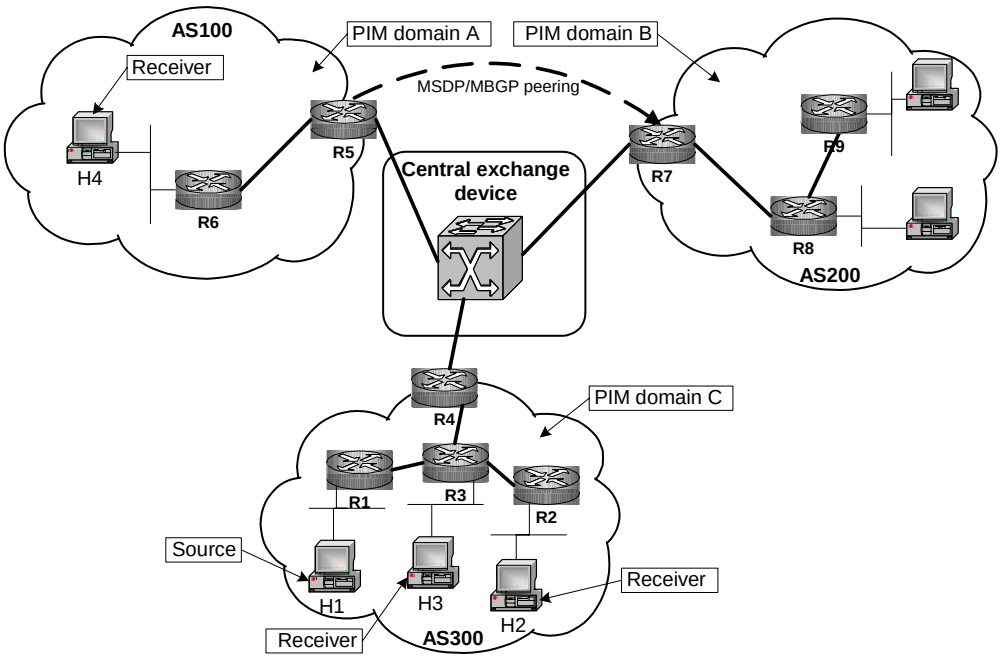

Figure 23: Example of interconnected PMBRs

This example shows two major problems associated with multicast traffic exchange in the interdomain:

1. How can receivers in PIM domain A (who only know about sources within their PIM domain from their RP) know about sources in domain B or vice versa?
   → This problem is solved by the use of MSDP described in section 5.2.3
2. **How can multicast distribution trees of PIM domain A be interconnected with distribution trees in PIM domain B or vice versa?**
   → PIM-SM **explicitly** uses **only SPTs** to interconnect PIM domains and therefore only sends **PIM (S,G) Joins/Prunes** and **no** (*,G) Joins/Prunes (in the case of STs) towards **sources** in other domains

The following steps explain how the problem stated under 2) is solved using PIM-SM as interdomain multicast routing protocol:

- Domain A, B and C (AS100, 200 and 300 respectively) are interconnected via a central exchange device. Each PMBR (R4, R5 and R7) has MSDP/MBGP peering and PIM-SM configured and each PMBR is also the RP for their domain.
- All receivers in domain C know about the source via their own RP (see section 4.3.1). The receivers in other domains know about this source via MSDP SA messages (see section 5.2.3).

- If the receiver H4 in domain A wants to receive traffic from the source in domain C its first hop routers sends an IGMP report onto the subnet which in turn is received by its DR (R6). R6 sends a (*,G) message towards the RP (R5) which creates an ST between the RP and R6. When the (*,G) arrives at the RP, it knows that this source is not within its own domain because it learned about that source via MSDP SA messages. Subsequently R5 sends a **PIM (S,G) Join message directly towards the source** (not to the RP in domain C) because it already knows the (S,G) information for that particular source. PIM-SM explicitly requires an RP in one domain to use (S,G) Join messages to create an interdomain SPT in order to interconnect domains.
- When the Join arrives at the DR of the source an SPT is established between R1 and R5 via the central exchange device. Traffic now initially flows down the SPT between R1 and R5 and then down the ST towards H4.

5.2.3 Multicast Source Discovery Protocol

Section 5.2.2 described the concept of a PIM-SM domain that uses RPs to provide information about active sources within the PIM-SM domain. Because each PIM-SM domain has its own RP and therefore only knows about sources in its own domain and nothing about sources in other PIM-SM domains the problem was how to interconnect these domains (STs) in the interdomain and **not** to be dependent on RPs in other domains.

This problem led to the Internet-Draft "Multicast Source Discovery Protocol (MSDP)" [47] which states the advantages of this protocol as follows:

- **No third-party resource dependencies on RP**
  PIM-SM domains can rely on their own RPs only
- **Receiver only domains**
  PIM domains with only receivers get data without globally advertising group membership

Figure 24 shows the general function of MSDP to exchange information about active sources between three different domains:

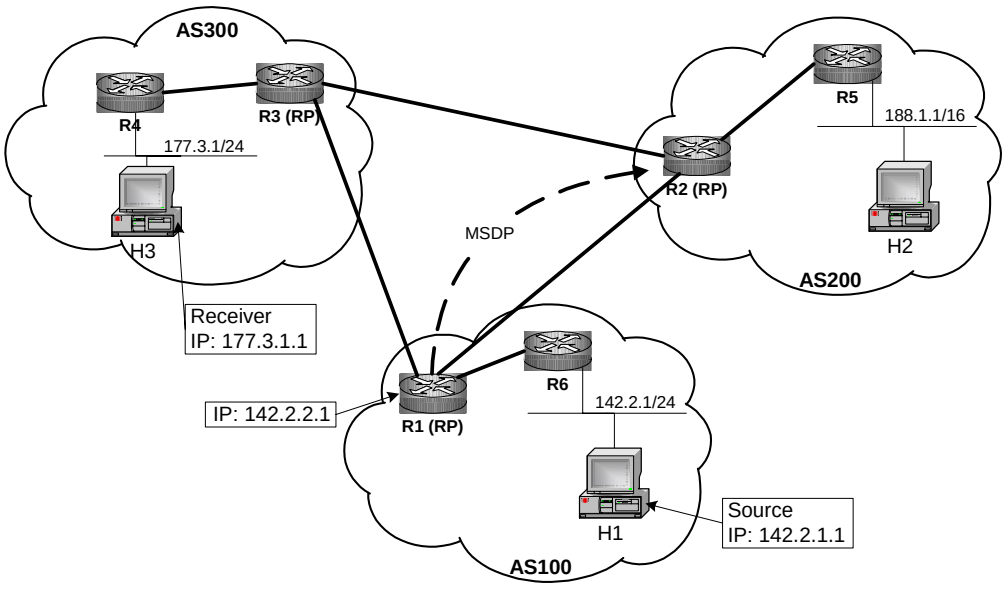

Figure 24: MSDP peering example

Each RP in a domain also has MSDP configured and maintains a MSDP peering session with the RPs in other domains (using a TCP/IP connection on port 639).

Host H1 (IP: 142.2.1.1) in AS100 starts sending multicast traffic to group 224.2.2.2. The first hop router (R6) encapsulates the multicast packets in PIM register messages and unicast them to the RP (R4) (see section 4.3.4). Therefore the RP knows that there is a source for group 224.2.2.2 within its domain. The RP periodically (default: 60 seconds) sends Source Active (SA) advertisements for each known source in its domain to MSDP peers in other domains containing the following information in the SA message:

- The source address of the data source (142.2.1.1)
- The group address the data source sends to (224.2.2.2)
- The IP address of the RP (142.2.2.1)

Each MSDP peer forwards these SA message based on the IP address of the originating RP away from this RP using a "peer-RPF flooding" mechanism described in [47]:

> *"The notion of peer-RPF flooding is with respect to forwarding SA messages. The Multicast RPF Routing Information Base (MRIB) is examined to determine which peer towards the originating RP of the SA message is selected. Such a peer is called an 'RPF peer'. … If the MSDP peer receives the SA from a non-RPF peer towards the originating RP, it will drop the message. Otherwise, it forwards the message to all its MSDP peers (except the one from which it received the SA message)."*

The SA message is therefore RPF flooded towards the RP in AS200 (R2) and the RP in AS300 (R3).

Host H3 in AS300 wants to receive multicast traffic for group 224.2.2.2. H3 knows about the existence of this source from its own RP (learned via SA messages from R1) therefore its first hop router first joins the ST towards its RP (R3) by sending a PIM (*,G) Join message upstream towards the RP (creating the ST). The RP (R3) in turn sends a **PIM (S,G) Join messages (142.2.1.1., 224.2.2.2)** towards the **source (142.2.1.1)** in AS100 effectively connecting the ST from the receiver to the RP (R3) together with an SPT from the RP (R3) to the source in AS100. After the PIM Join is received by the sources first hop router multicast traffic flows on the SPT from the source to R3 and then on the ST to the receiver H3.

If the amount of multicast traffic exceeds a certain threshold the first hop router of H3 can decide to create an SPT directly to the source (not via the RP (R3)). It does so by sending a PIM (S,G) Join message directly towards the source effectively creating an SPT that spans from the source directly to receiver H3. This procedure is also described in section 5.2.2.

The MSDP Internet-Draft also recommends the following router behavior:

> *"A MSDP speaker SHOULD cache SA messages. Caching allows pacing of MSDP messages as well as reducing join latency for new receivers of a group G at an originating RP which has existing MSDP (S,G) state. In addition, caching greatly aids in diagnosis and debugging of various problems."*

This recommendation can have a severe impact on the vulnerability of MSDP enabled routers because caching of SA information can put a rather heavy load on a router (see section 7.2)

A lot of discussion is going on in the multicast community if the use of MSDP will scale in the future because there are a few known problems associated with the use of MSDP:

- SA messages have to be propagated throughout all interconnected PIM domains which effectively creates a flat database of source information
- Routers normally cache SA messages. Therefore more router memory is needed to store SA information

- MSDP peers with a flawed configuration can cause MSDP SA loops that prevent SA messages from being distributed throughout all PIM domains (→ active sources cannot be “seen” in other domains)
- Because of its flooding behavior an the caching of SA information MSDP could be (was) used in Distributed Denial of Service (DDoS) attacks (see recommendation above and section 7.2)

Despite all the know problems MSDP is the de facto standard currently used to provide a method of interconnecting PIM-SM domains. The Internet Engineering Task Force (IETF) [48] is currently working on new protocols that could scale much better than MSDP (e.g. MADCAP/MASC).

Note: SSM (see section 2.4.2) only uses SPTs and SSM sources are discovered “out of band” (e.g. via a website) MSDP is not used in the SSM service model.

6 MIX Design Considerations

6.1 Border-Router-Only Environment

An interdomain traffic exchange point like a MIX consists of a core switch that interconnects **only interdomain BRs** (PMBRs) from various ASs (domains) (see section 5.1). This creates a special network environment in this document referred to as “BR-only” environment because only certain types of protocol traffic can be “seen” at the core switch which differs considerably from protocols that are used in the intradomain. For example IGMP membership reports are always sent with a TTL of 1 and will therefore never reach any router beyond its own subnet and are therefore bound to the intradomain. On the other hand protocols like BGP or PIM can be used in the interdomain or intradomain.

6.2 Switch Broadcast Behavior

Generally Layer 2 switches **broadcast** MAC frames with an **unknown** destination MAC address to **all** switch ports. Multicast MAC frames have a destination address that corresponds to the IP Multicast address (see section 2.7.3 / 2.7.4) and no physical MAC address of a device connected to the switch. A MAC frame forwarding decision however is based on the switches Forwarding Information Base (FIB) that consists of a table of MAC address/port correlations “learned” by the switch during the reception of frames. In order to populate the FIB the switch creates a new FIB entry whenever it receives a frame with a new MAC address and associates a timeout value with it in order to remove this entry from the FIB if the device on a specific port goes away.

For example a host directly connected to a switch sends a multicast MAC frame with destination multicast MAC address of 01:00:5E:02:02:02. The FIB does not contain this MAC address because it is no “physical” address of any device connected to the switch therefore the switch has to **broadcast** this frame **to all switch ports** instead of discarding it.

Because a switch reverts to a broadcast method this can have (depending on the actual hardware implementation) a serious impact on the switch performance as the following test in the Sandbox testbed showed.

The test of the BI8000 (MIX core switch) broadcast behavior consisted of the following steps:

1. Create a port-based Virtual LAN (VLAN) with two Gigabit Ethernet (GigE) ports (GigE 1/1 and GigE 1/2)
2. Enable multicast containment on the switch
3. Connect GigE 1/1 to the Ixia1600 traffic generator (port 2/1)
4. Connect GigE 1/2 to the Ixia1600 traffic generator (port 2/2)
5. Send 10,000,000 frames with multicast MAC address 01:00:5E:02:02:02 from the Ixia1600 port 2/1 to GigE 1/1 (which is effectively broadcast traffic because the switch has no FIB entry for this MAC address) with various frame sizes (64, 128, 256, 512, 1024, 1518 bytes) at various line-rates (10 – 100% of GigE line-rate in 10% increments)
6. Check how many multicast frames are received on the Ixia1600 port 2/2 from GigE 1/2
7. Repeat step 5 and 6 with all combinations of the frame sizes and line-rates stated in step 5.
8. The result are plotted in Figure 25

Note: This test might produce different results on different hardware and is dependent on the actual hardware architecture a particular vendor uses.

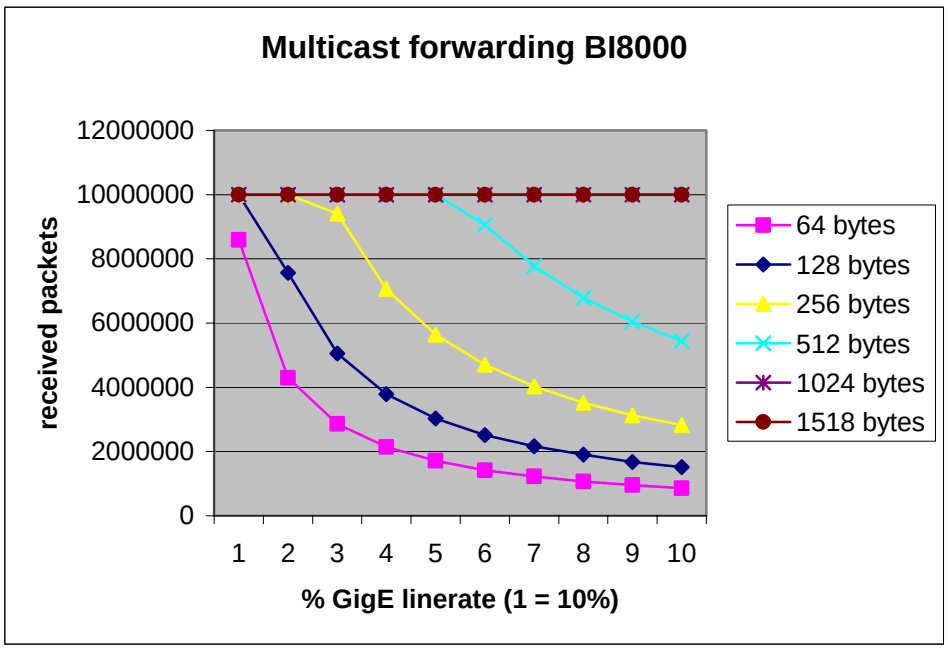

Figure 25: Broadcast behavior of a Foundry BigIron 8000 switch

The results in Figure 25 clearly show that the switch has to process each multicast frame via the CPU (slow-path). With a small frame size (64 bytes) at GigE line-rate ~90% of all multicast frames are dropped. Would the switch forward broadcast packets "in hardware" (fast-path) via an application-specific integrated circuit (ASIC) in conjunction with Content Addressable Memory (CAM) table the switch should be able to forward **all** frames (independent of frame size or transmission speed) to all necessary egress (output) ports. The following section addresses this problem and describes different multicast containment technologies that can be used to optimize multicast packet replication.

6.3 Multicast Containment Technologies

Most of today's switches are handling multicast traffic by default like broadcast traffic (see section 6.2) and flood MAC frames onto **all** switch ports (within a given VLAN) because of the lack of information on how to efficiently replicate specific multicast traffic onto ports where receivers for a specific multicast group traffic exist.

Multicast containment technologies try to prevent this situation by gathering information from Layer 2 or Layer 3 protocols in order to populate the FIB with forwarding information that can be used to forward frames using "fast-path" (ASIC).

Currently there are a few multicast containment protocols available which are specifically designed to work in a LAN environment (e.g. Cisco Group Management Protocol (CGMP) or IGMP-snooping [50]) but not in a BR-only network environment like a MIX (see section 6.1).

The following section describes three protocols that can be used for multicast containment. A combination of the first two protocols (RGPM and IGMP-snooping) led to the multicast containment proposal of "PIM-snooping" (see section 6.3.3).

6.3.1 Router-Port Group Management Protocol

In May 2000 Cisco Systems [7] first introduced a multicast containment protocol called Router-Port Group Management Protocol (RGMP) [8] with Internetwork Operating System (IOS) version 12.0(10)S. This protocol is especially designed to work in BR-only network environments.

Cisco describes RGMP as follows [8]:

> *"The Router-Port Group Management Protocol (RGMP) feature introduces a Cisco protocol that restricts IP multicast traffic in switched networks. RGMP is a Layer 2 protocol that enables a router to communicate to a switch (or a networking device that is functioning as a Layer 2 switch) the multicast group for which the router would like to receive or forward traffic. RGMP restricts multicast traffic at the ports of RGMP-enabled switches that lead to interfaces of RGMP-enabled routers. RGMP is designed for switched Ethernet backbone networks running Protocol Independent Multicast (PIM) sparse mode."*

RGMP is similar to the Cisco Group Management Protocol (CGMP) [50] as it uses control messages between Layer 2 and Layer 3 network devices in order to contain multicast traffic. However, CGMP and RGMP do not interoperate in the same, switched network unlike RGMP and IGMP-snooping.

Note: There also exists a informational Internet-Draft (now expired) [49].

RGMP / PIM-snooping Benefits

The benefits of using RGMP also apply to the proposed PIM-snooping multicast containment described in section 6.3.3. Cisco describes the benefits of using RGMP are follows [8]:

Increases Available Bandwidth

> *"By restricting unwanted multicast traffic in a switched network, RGMP increases the available bandwidth for all other multicast traffic in the network. Without RGMP, the sum of all multicast traffic sent into a switched network must be smaller than the slowest link on the slowest router can sustain. With RGMP, this restriction is limited. Multicast traffic will flood only on links between switches in the network, whereas routers will receive only the multicast traffic that they need."*

Note: This is even more important if the link between a core switch and a router consists of an "expensive" WAN circuit as in a MIX environment (see chapter 5).

Increases Scalability for Multicast Traffic

> *"In a switched Ethernet network where RGMP is not enabled and* n *routers (*n *being any number of routers) are connected to a single switch through individual 100-Mbps full-duplex connections, the theoretical maximum aggregate bandwidth for unicast traffic sent into the network is* n ** 100 Mbps and the theoretical maximum aggregate bandwidth for multicast traffic sent into the network is only 100 Mbps. When RGMP is enabled in this same network, the theoretical maximum aggregate bandwidth for multicast traffic sent into the network changes to* n ** 100 Mbps."*

Note: In a "best case" scenario (using RGMP containment) each connected router would send multicast traffic (for various multicast groups) only to **one** downstream router with receivers behind it. This is essentially the same as standard unicast forwarding. In a "worst case" scenario each router would send multicast traffic to various groups and **every** connected downstream router would have receivers behind it. The aggregated traffic in this case would still be 100 Mbps.

Increases Available Resources

> *"By eliminating unwanted multicast traffic in a switched network, routers need not devote processing resources to examining unwanted multicast packets."*

Note: Without multicast containment **all** routers connected to a MIX would have to process unwanted multicast traffic. This could have a very serious negative impact on the routing performance for unicast and multicast traffic of all connected PMBRs.

Limited Impact on Router and Switch Efficiency

> *"RGMP has limited impact on router and switch processing resources and does not require additional system memory. RGMP also does not introduce new timers or other control mechanisms in routers that might introduce new error conditions."*

Note: The impact of using RGMP on a router is not critical because it uses RGMP protocol messages but if containment technologies like IGMP-snooping (see section 6.3.2) or PIM-snooping (see section 6.3.3) are used on a switch the additional processing resources that have to be used to examine each received frame can be very high (depending on the hardware switch design)

6.3.2 IGMP-snooping

Internet Group Management Protocol (IGMP) snooping [13] is a **non-proprietary** Layer 2 protocol implemented on switches to restrict multicast traffic from switches to hosts in a LAN environment. The word "protocol" is somewhat misleading because IGMP-snooping does not introduce any kind of new IGMP protocol messages. It is a mechanism implemented in Layer 2 switches to intercept Layer 3 IGMP messages and use the information contained in these messages to maintain the switches CAM table in order to restrict multicast traffic from being flooded to all active (ports with active forwarding state) switch ports.

Another **proprietary** protocol used in LAN environments is the Cisco Group Management Protocol (CGMP) [50]. CGMP uses its own control messages to signal the reception of a particular multicast group to neighboring routers.

The problem with both multicast containment protocols is that they are designed for LAN environments and therefore they only restrict multicast traffic from switches to hosts and **not** from switches to routers like RGMP (see section 6.3.1).

This section describes only the basic concept of IGMP-snooping because it is similar to the newly proposed concept of PIM-snooping (see section 6.3.3).

The example in Figure 26 shows the basic IGMP-snooping mechanism.

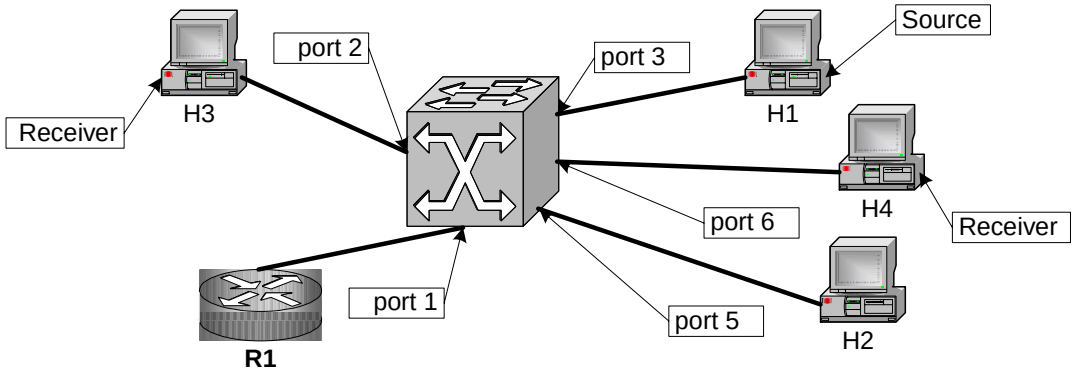

Figure 26: Basic IGMP snooping mechanism

Device	On switch port	Description
R1	1	The IGMP query router (DR for the subnet)
H1	3	Host sending to multicast group 224.2.2.2 (Multicast MAC: 01:00:5E:02:02:02)
H2	5	Not a receiver (passive host)
H3	2	Receiver of group 224.2.2.2
H4	6	Receiver of group 224.2.2.2

Because IGMP control messages are also send with the same multicast MAC address like normal multicast traffic they cannot be distinguished only on the basis of their MAC address. Therefore the switch has to analyze Layer 3 information in order to distinguish IGMP messages from multicast traffic. Therefore the switch initially creates a CAM entry that causes all Layer 3 IGMP control messages to be forward to the CPU (internal port 0) for further processing. Then the following steps are performed:

1. H1 is sending multicast traffic to group 224.2.2.2 which maps to the corresponding multicast MAC address 01:00:5E:02:02:02
2. The frames are received on switch port 3. Because there is no entry for that MAC address in the CAM table, the switch forwards the frames to ports where routers are connected (port 1). This behavior is based on how the application-specific integrated circuit (ASIC) is implemented (another method could be to initially flood all frames within a given VLAN)
3. H3 wants to receive this traffic and sends an unsolicited IGMP report for multicast group 224.2.2.2. This report is sent to multicast address 224.2.2.2 (01:00:5E:02:02:02) and received by the switch on port 2

4. The CAM table lookup shows that this is an IGMP control messages an is forwarded to the CPU
5. The CPU creates an entry for this MAC address in the CAM table and associates the port where the messages was receive with it.
   Now traffic flows from port 3 to port 2 and 1
6. H4 also wants to receive this traffic and sends an unsolicited IGMP report for multicast group 224.2.2.2. This report is sent to multicast address 224.2.2.2 (01:00:5E:02:02:02) and received by the switch on port 6
7. The CAM table lookup shows that this is an IGMP control messages an is forwarded to the switch CPU
8. The CPU examines the CAM table and (because the MAC address for this group already exists) adds the port where the report was received to the MAC entry in the CAM table. Then traffic flows from port 3 to port 2, 1 and 6

If a receiver wants to leave a multicast group (e.g. 224.2.2.2) it sends an IGMP leave messages (in IGMPv2), which is forwarded to the CPU. The CPU then performs a lookup in the CAM table and removes the port association from the MAC entry. If all port associations are removed from the MAC entry the whole CAM entry is removed. If a receiver goes silently away (IGMPv1 or the host is simply switched off) the switch has to wait for the next Membership query / response mechanism in order to detect ports with no receivers.

6.3.3 PIM-snooping

Based on the experience of the project at The London Internet Exchange [5] and the AMES MIX Internet-Draft [45] most of the current MIXs do not (can not) use any kind of multicast containment technology because it is simply not available. Therefore multicast traffic is flooded to all connected PMBRs. In order to prevent this situation there is currently (May 2001) only one **proprietary** solution available called RGMP as described in section 6.3.1.

This thesis **proposes a new**, **non-proprietary** approach called "**PIM-snooping**" which was implemented upon suggestion of Equinix [1] by Foundry Networks [51] on one of their high-end core switches (BigIron 8000 (BI8000) with switch code release >= 7.2).

The goal of PIM-snooping as well as IGMP-snooping or RGMP is to create multicast MAC entries in the Forwarding Information Base (FIB) of a switch in order to perform ASIC based, line-rate packet forwarding and avoid multicast packet flooding to all switch port. PIM-snooping utilizes PIM-SM Join/Prune messages (see section 4.3.5) to gather the required information.

Note: A PIM-snooping enable MIX core switch does **not** actively send any kind of messages it will only snoop (listen) for PIM Join/Prune messages and uses the information contained in these messages to maintain its Forwarding Information Base (FIB) and subsequently send multicast traffic only onto port where receivers have explicitly requested the reception of a particular multicast group.

The following example demonstrates how PIM-snooping is currently implemented on a Foundry BI8000 switch. The PMBR and host setup shown here is just a schematic example. For a more detailed and realistic example see section 6.3.3.4.

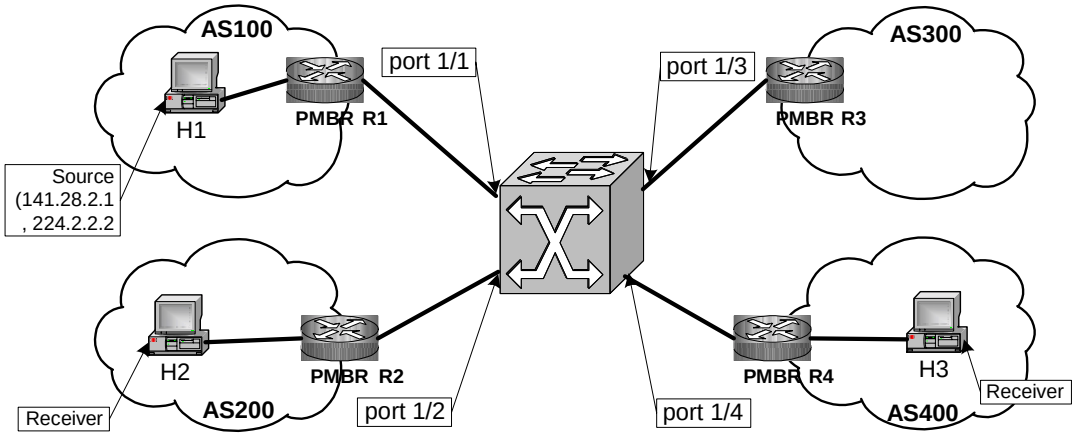

Figure 27: PIM-snooping operation

In the example above a core switch interconnects four PMBRs of four different domains. The following two prerequisites also apply to section 6.3.3.1 and 6.3.3.2:

PMBR prerequisites

In Figure 27 all PMBRs (R1, R2, R3 and R4)

- use one Ethernet Interface (10/100 or GigE) to connect to the core switch (see section 5.1)
- are PMBRs and are configured to be the RP for their domain (see section 4.3)
- are running MBGP, MSDP, PIM-SMv2 and are configured accordingly (see Appendix)
- have a full-mesh of MBGP peering sessions established between each other (see section 5.2.1)
- have a full-mesh of MSDP peering sessions established between each other (see section 5.2.3)

Core switch prerequisites

- the “PIM-snooping” multicast containment feature (described in this section) is **enabled**
- one port-based VLAN exists (e.g. 99). This VLAN will be called “**Multicast VLAN**”
- Ports 1/1, 1/2, 1/3 and 1/4 are members of this Multicast VLAN
- All multicast packets with a Link-local destination address (e.g. 224.0.0.13 → ALL-PIM-ROUTERS) (see section 2.7.2.1) have to be broadcast to all active ports within a given Multicast VLAN because multicast traffic in this range is administrative traffic.
- According to Foundry [52] the switch assumes that the group source and the Layer 2 switch are in different sub-nets and communicate through a router. The source must be in a different sub-net than the receiver because a PIM router sends PIM Join/Prune messages on behalf of a multicast group receiver **only** when the router and the source are in different sub-nets. When the receiver and source are in the same sub-net, they do not need the router in order to find another. They find one another directly within the subnet.

6.3.3.1 PIM Neighbor Discovery

In order to discover all PIM capable routers connected to ports that are a member in the Multicast VLAN the core switch snoops for PIM hello messages. RFC 2362 [39] defines the PIM hello mechanism as follows:

> *“Hello messages are sent periodically between PIM neighbors, every [Hello-Period] seconds. This informs routers what interfaces have PIM neighbors. Hello messages are multicast using address 224.0.0.13 (ALL-PIM-ROUTERS group). The packet includes a Holdtime, set to [Hello-Holdtime], for neighbors to keep the information valid. Hellos are sent on all types of communication links.”*

Note: For default timer values see section 4.3.6.

The core switch receives these PIM hello messages on ports where PIM-SM routers are connected. After snooping these messages the “PIM neighbor” information on the BI8000 switch using the command “show ip pim” for the example in Figure 27 is:

```
telnet@BI8000# sh ip pim
PIMSM snooping is enabled
VLAN ID 99
PIMSM Neighbor list:
        202.2.2.3 : 1/3   expire 95 s
        202.2.2.4 : 1/4   expire 105 s
        202.2.2.1 : 1/1   expire 85 s
        202.2.2.2 : 1/2   expire 75 s
```

As shown in the output above, the current PIM neighbor expiration timer of R3 (connected to port 1/3) is 95 seconds. This means that the last PIM hello message has been received 10 sec. ago. This counter will decrement until it reaches 65 sec. and will then be reset to 105 sec. by the next periodical PIM hello message. If no PIM hello message is received within 105 sec. the PIM neighbor entry will be deleted. Currently neighbor discovery is just used for informational purpose but this information can also be used in case the core switch is running out of CAM memory (see section 6.4). Note that it takes at least [Hello-Period] seconds to discover **all** PIM neighbors connected to the core switch.

6.3.3.2 PIM-snooping Operation

Based on the topology in Figure 27 we assume that the source of multicast traffic H1 (IP: 141.28.2.1) is sending to group 224.2.2.2 and that the receivers are H2 and H3. Note that there is **no** receiver in AS300.

Because the receivers behind R2 and R4 want to receive the multicast traffic their PMBRs will send a **PIM (S,G) Join message** (→ (141.28.2.1, 224.2.2.2) towards the source in AS100 (hence R1 will receive the PIM Join message) (for details, see section 6.3.3.4). The PIM-SM Join messages are sent with a destination multicast address of 224.0.0.13 (ALL-PIM-ROUTERS) and the IP protocol number in the IP header set to

103. This message traverses the core switch in order to reach the upstream PIM neighbor.

Creating CAM Table Entries

Then the core switch will perform the following tasks

1. The PIM (S,G) Join message (IP proto 103) is **received** on port 1/2 (from R2) and port 1/4 (from R4)
2. Each IP packet is examined for an IP protocol number of 103 in the IP header. If the packet had IP protocol number 103, the frame is broadcast to all active ports within the Multicast VLAN and then forwarded to the CPU (slow-path) for further processing (for details see section 6.3.3.3).
3. The switch checks on which **port** for which multicast **group** a PIM Join messages was received (Port: 1/2 Group: 224.2.2.2 and Port: 1/4 Group: 224.2.2.2)
4. It then checks if there is already an entry for the corresponding MAC address (224.2.2.2 maps to multicast MAC 01:00:5E:02:02:02) in the CAM table of the port where the PIM Join was received
5. If there was **no** entry, the switch creates an entry in the CAM table for multicast group 224.2.2.2 (MAC: 01:00:5E:02:02:02) and associates the port where the PIM Join was received with the MAC address
6. The following table shows the CAM table after the two PIM Joins have been processed

CAM table of the switch after step 6:

MAC address	Dest. Ports	Protocol type	Timeout (sec.)	camindex
01:00:5E:00:00:xx	0 (CPU)	PIM	n/a	n/a
01:00:5E:02:02:02	1/2	!PIM	210	2061
01:00:5E:02:02:02	1/4	!PIM	210	2062

The CAM table above is based on an example of IGMP-snooping from Williamson [13] (page 417) and is based on an assumption of what kind of information should be kept in the CAM table in order to perform ASIC based, line-rate multicast packet forwarding. Unfortunately Foundry Networks did not want to give details about their PIM-snooping implementation and CAM table structure to the public.

The entries in the table above are as follows:

- The first entry in the above CAM table is created by default after enabling multicast containment and instructs the switch to forward all link-local multicast addresses with an IP protocol number of 103 (PIM) in the IP header to the CPU (internal port 0) for further processing.
- The second entry was created by the reception of the PIM (S,G) Join/Prune received on port 1/2 from R2 and instructs the switch to **replicate** subsequently received multicast frames with MAC address 01:00:5E:02:02:02 onto port 1/2
- The third entry was created by the reception of the PIM (S,G) Join/Prune received on port 1/4 from R4 and instructs the switch to **replicate** subsequently received multicast frames with MAC address 01:00:5E:02:02:02 onto port 1/4
- The 2nd and 3rd MAC entry will timeout after 210 seconds if the timers are not reset by another PIM Join message for the same destination multicast MAC address (default 60 seconds). For default timeout values see section 4.3.6.

Removing CAM Table Entries

In order to remove the two “snooped” MAC entries for multicast MAC 01:00:5E:02:02:02 there are two possibilities:

- The entry expires (times out after 210 seconds) or
- The entry is removed by the reception of a PIM Prune message destined for the **same** multicast group address.

The problem at that point is that the PIM-SM protocol specification supports the creation of SPTs based on the **unique** combination of **S**ource **and G**roup IP addresses (see also section 2.4.2) but a switches CAM table structure only allows to store **one** destination MAC entry without regard to the source of the multicast packets.

Therefore if **one** router (e.g. R2 in Figure 27) would send a PIM Join message for **S**ource 141.28.2.1 and **G**roup 224.2.2.2 → (141.28.2.1, 224.2.2.2) this would result in **one** CAM table MAC entry for multicast group 224.2.2.2 (MAC: 01:00:5E:02:02:02) with an associated reception port of 1/2.

Assume that another PIM Join message containing a different **S**ource address (154.3.5.1) but the **same G**roup address (224.2.2.2) → (154.3.5.1, 224.2.2.2) is received from R2 on the **same** port (1/2). This would **not** result in a new CAM table entry because there already exists a MAC entry for Group 224.2.2.2 / port 1/2.

The following table shows the forwarding table entries after the reception of the two PIM Joins on port 1/2 and an additional PIM Join received on port 1/4 for (141.28.2.1, 224.2.2.2):

MAC address	Dest. Ports	Protocol type	Timeout (sec.)	source-list pointer
01:00:5E:00:00:xx	0 (CPU)	PIM	n/a	n/a
01:00:5E:02:02:02	1/2	!PIM	210	0010
01:00:5E:02:02:02	1/4	!PIM	210	0020

source-list pointer	Source IP address
0010	141.28.2.1
0010	154.3.5.1
0020	141.28.2.1

In the next step R2 wants to Prune the (S,G) entry for (**141.28.2.1**, 224.2.2.2) but the forwarding state for (**154.3.5.1**, 224.2.2.2) should be kept alive in the CAM table because this multicast source is still sending multicast traffic on this SPT. This obviously causes a problem because there is only one entry for group 224.2.2.2 in the CAM table that must be kept until the **last source** is pruned by R2.

The current PIM-snooping implementation solves this problem as follows:

- For each MAC/port entry in the CAM table, **a list** of sources (source-list) is stored outside of the CAM table e.g. in Random Access Memory (RAM). A reference (pointer) to the associated source-list entries is additionally stored in the CAM table (source-list pointer). Each time a PIM Prune message for a different **S**ource but the **same G**roup is received, only the source-list entry is deleted. The CAM table entry for this Group is **not** removed.
- When the **last entry** in the source-list either expires or is removed by a PIM Prune then the corresponding MAC/port entry in the CAM table is also removed.

After the above PIM Prune for (141.28.2.1, 224.2.2.2) was processed the CAM table would look like this:

MAC address	Dest. Ports	Protocol type	Timeout (sec.)	source-list pointer
01:00:5E:00:00:xx	0 (CPU)	PIM	n/a	n/a
01:00:5E:02:02:02	1/2	!PIM	210	0010
01:00:5E:02:02:02	1/4	!PIM	210	0020

source-list pointer	Source IP address
0010	154.3.5.1
0020	141.28.2.1

If the switch would receive another PIM Prune for (154.3.5.1, 224.2.2.2) the source-list entry 0010 → 154.3.5.1 **and** the CAM table entry for MAC address 01:00:5E:02:02:02, port 1/2 would be removed because it was the last entry in the source-list.

6.3.3.3 PIM Join/Prune Message Processing

The actual PIM Join/Prune message processing described in section 6.3.3.2 was simplified in order to emphasize on the PIM-snooping operation.

As the PIM Join/Prune message format (see section 4.3.5) implies, a Join/Prune message can not just be used to send **one** join message or **one** prune message to an upstream PIM neighbor, furthermore a Join/Prune message consists of a **set** of multicast group addresses which subsequently contain

- 0 to *n* joined source addresses and/or
- 0 to *n* pruned source addresses

Therefore the processing of Join/Prune messages in the switch is more complex because the switch has to process the whole list of Joins and Prunes and update the switches CAM table accordingly.

The following pseudo-code shows how the content of a PIM Join/Prune message (see section 4.3.5) could be processed (maintain CAM/source-list entries):

```
If [PIM version] == 2 AND [Type] == 3 AND [Checksum] == OK
    i = j = k = 1;
    rec_port = reception_port();
    ip_proto = 103;

    For each i = [Number of Groups] do
        If Group_not_exists_in_cam([Multicast Group Address i])
            source_list_pointer = get_free_source_list_pointer();

            Create_cam_group_entry([Multicast Group Address i],
                rec_port, ip_proto, source_list_pointer);
        else
            source_list_pointer =
            get_existing_source_list_pointer([Multicast Group Address
            i], rec_port, ip_proto);
        End if

        For each j = [Number of Joined Sources] do
            Create_source_list_entry(source_list_pointer, [Joined
            Source Address j]);
        Next j

        For each k = [Number of Pruned Sources] do
            Delete_source_list_entry(source_list_pointer, [Pruned
            Source Address k]);
        Next k

        If source_list_empty(source_list_pointer)
            Delete_cam_group_entry([Multicast Group Address i],
            rec_port, ip_proto);
        End if
    Next i
End if
```

6.3.3.4 PIM-snooping Operation at a MIX

This section describes the usage of PIM-snooping at a MIX by an example of four domains interconnected via a MIX core switch:

Figure 28 shows how the PIM-snooping mechanism works and is based on the following topology:

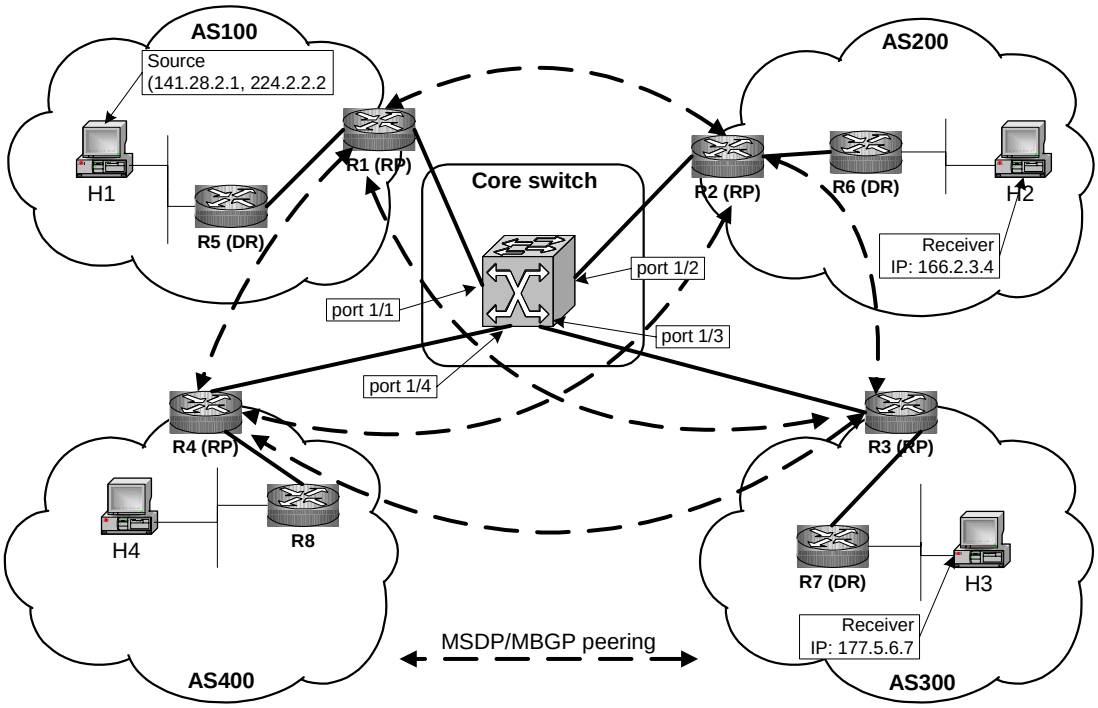

Figure 28: Example of PIM-snooping at a MIX

For this example the PMBR/Core switch prerequisites from section 6.3.3 and the Host prerequisites below apply.

Host prerequisites

The receivers H2 (AS200) and H3 (AS300) have already sent a IGMP membership report for group 224.2.2.2. Therefore a ST between R2 (RP) and R6 (DR) (R3, R7 respectively) has already been created and the RP "knows" that there are receivers for this traffic in its domain. As soon as traffic for group 224.2.2.2 arrives at the RPs it will flow down the ST to the receivers H2 and H3.

Note: H4 in AS400 has **not** sent any IGMP membership report for any multicast group (is passive).

The following paragraph describes the steps that happen before traffic flows between the source (H1) and the receivers (H2, H3):

1. H1 (IP: 141.28.2.1) in AS100 starts sending traffic to multicast group 224.2.2.2. According to the ASM service model (see section 2.4.1) there is no further signaling to any router necessary. The source does not have to join the group in order to be able to send traffic to the group
2. R5 encapsulates the first multicast packets in PIM register messages (see section 4.3.4) and unicast them to R1
3. R1 receives the PIM register messages for group 224.2.2.2 and immediately sends MSDP SA messages (see section 5.2.3) to all its MSDP peers (unicast) to signal the existence of a source 141.28.2.1 in AS100 sending to group 224.2.2.2
4. R1 sends an (S,G) join towards the source in order to pull the traffic natively to the RP. This creates a SPT from the source to the RP
5. As soon as the first multicast packets arrive via the SPT R1 sends PIM register stop messages to R5 in order to stop the unicast reception via PIM register messages
6. The MSDP SA messages are received by R2 and R3 (MSDP peers). Both RPs know that they have receivers for this traffic within their domain and therefore **immediately** send a **PIM-SM (S,G) Join message** that travels **hop-by-hop** towards the **source** H1 in AS100 → (S,G) = (141.28.2.1, 224.2.2.2) (see section 4.3.3). For a detailed explanation of the PIM Join/Prune message format see section 4.3.5
7. The PIM Join messages are traversing the core switch that uses PIM-snooping (see section 6.3.3.2) in order to update its CAM table according to the PIM Join messages it receives from R2 and R3

8. The CAM table / source-list now contains two entries for (141.28.2.1, 224.2.2.2) see below:

MAC address	Dest. Ports	Protocol type	Timeout (sec.)	source-list pointer
01:00:5E:00:00:xx	0 (CPU)	PIM	n/a	n/a
01:00:5E:02:02:02	1/2	!PIM	210	0010
01:00:5E:02:02:02	1/3	!PIM	210	0020

source-list pointer	Source IP address
0010	141.28.2.1
0020	141.28.2.1

9. R1 sends the received PIM Join to the next-hop in the direction of the source effectively creating the (141.28.2.1, 224.2.2.2) SPT that spans from R5 to R2/R3
10. When the PIM (S,G) Joins reach R1 (which already has an (S,G) entry for that source and is already receiving traffic via the SPT) it sends the multicast frames natively towards downstream receivers
11. When the multicast frames (destination MAC 01:00:5E:02:02:02) are received by the core switch it **already** has a CAM table entry for these destination MAC addresses (see table above) and **replicates** the received packets onto ports 1/2 and 1/3 and **no other ports** within the Multicast VLAN.
12. Now multicast traffic flows on the SPT from H1 via R5, R1, R2, R6 to H2 and via R5, R1, R3, R7 to H3. Note that the (S,G) state is only maintained once in each router along the SPT
13. If H2 leaves the multicast group R6 will send a PIM Prune message (141.28.2.1, 224.2.2.2) hop-by-hop towards the source

14. The core switch also receives the PIM Prune message and subsequently updates its CAM table and source-list entries. The source-list entry and (because it is the last entry in the source-list for that group) the CAM table entry for port 1/2 is deleted and traffic will **not** be replicated onto port 1/2 anymore

15. R3 still receives traffic from R1 because the CAM/port, source-list entry still exists

Note:

- In step 6, initial MSDP SA messages contain source multicast packets. These are immediately forwarded down the ST towards receivers. The MSDP Internet-Draft states [47]:

  "The RP may encapsulate multicast data from the source. An interested RP may decapsulate the packet, which SHOULD be forwarded as if a PIM register encapsulated packet was received. … Note that when doing data encapsulation, an implementation MUST bound the time during which packets are encapsulated. This allows for small bursts to be received before the multicast tree is built back toward the source's domain. For example, an implementation SHOULD encapsulate at least the first packet to provide service to bursty sources."

- R7 in AS400 did **not** have to send any PIM-SM Prune to any router because PIM-SM is an explicit **join** protocol and traffic only flows if it was explicitly requested by a PIM-SM Join message.

6.3.3.5 PIM-snooping Using Trunked Switch

In certain cases it could be required to deploy a MIX infrastructure that consists of multiple edge-switches interconnected (via trunks) to one core switch as shown in Figure 29. This topology was implemented within Equinix IBX™ centers as their unicast (multicast) exchange topology.

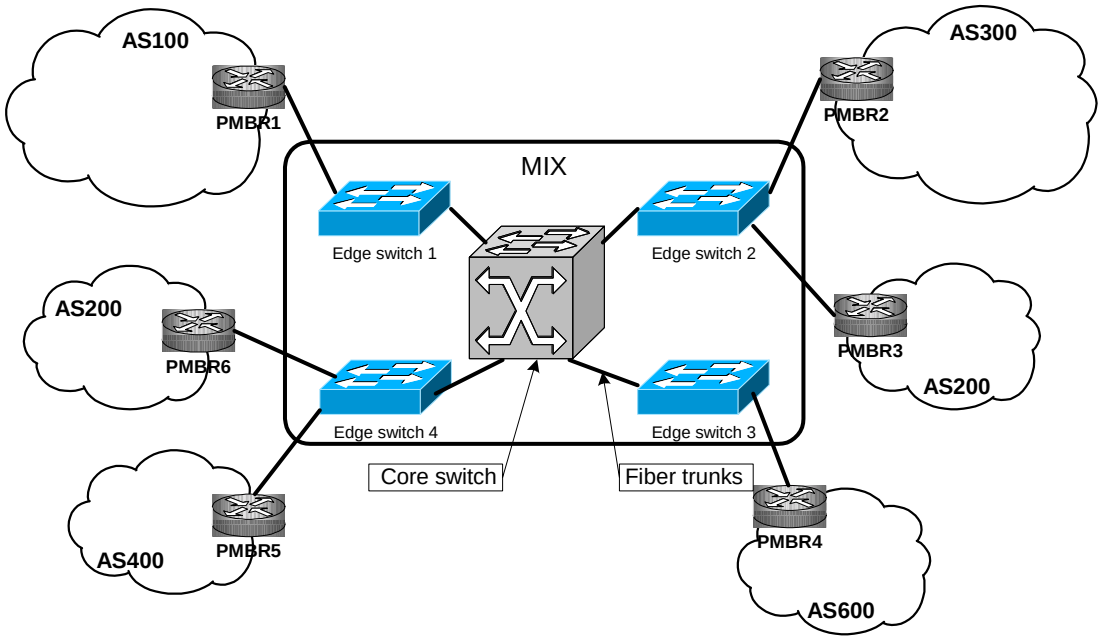

Figure 29: Planned Equinix MIX topology

In the topology above each edge-switch uses one trunk (consisting of 2 GigE interfaces) to connect to the core switch.

The scalability of a MIX depends not just on the scalability of the core switch but also on all switches in the whole MIX topology. Therefore in order to provide a scalable MIX all switches (including the edge-switches) have to support the PIM-snooping multicast containment feature. According to tests in the Foundry labs [52] PIM-snooping in a "Global Ethernet environment" test works exactly like in a environment that consists of just **one** core switch. This could not been verified in the Sandbox because we didn't have the necessary switches (at least three BI8000). From a theoretical standpoint there is no reason why PIM snooping should not work in a trunked Layer 2 MIX topology.

The only problem that could occur is that forwarding state information (CAM table entries) have to be redundantly stored in edge-switches that don't need this information. This assumption must also be verified in further tests.

6.4 Packet Replication Restrictions

The Layer 2 packet replication restrictions on a switch using PIM-snooping (see section 6.3.3) also apply to RGMP (see section 6.3.1):

- PIM-snooping does **not** restrict flooding of multicast traffic for link-local multicast addresses 224.0.0.0 – 224.0.0.255 (see section 2.7.2.1). Multicast traffic with IP multicast destination addresses in this range and MAC ambiguous link-local addresses (see section 2.7.3) in the same range will always be flooded to all ports within the Multicast VLAN.
- The current PIM-snooping implementation does **not** restrict multicast traffic based on a combination of multicast source **and** group address (S,G) it only uses the group address (G). This is due to the fact that
    - The hardware (ASIC) and CAM table design (MAC entries can only be 6 bytes wide)
    - The actual PIM-snooping implementation does not support it

    According to Foundry Networks, future switches will allow "wider" CAM table entries and could therefore make forwarding decisions based on a combination of source **and** group address (S,G). This would be especially beneficial for SSM based applications because SSM creates **only** SPTs and **no** STs and therefore only has (S,G) forwarding state.

- The core switch cannot restrict the replication of multicast traffic for ambiguous multicast MAC addresses (see section 2.7.3 and 2.7.4)
  E.g. a source sending to multicast group **224**.1.2.3 will conflict with a source sending to multicast group **229**.1.2.3 because both map to the same multicast MAC 01:00:5E:01:02:03.

  For example, a host H1 is sending traffic to (141.28.2.1, 224.1.2.3) received by the core switch on port 1/1. Another host (H2) is sending traffic to (155.33.2.11, 229.1.2.3) received on core switch port 1/2.

  The CAM table / source-list would be updated as follows:

MAC address	Dest. Ports	Protocol type	Timeout (sec.)	Source-list pointer
01:00:5E:00:00:xx	0 (CPU)	PIM	n/a	n/a
01:00:5E:01:02:03	1/1	!PIM	210	0010
01:00:5E:01:02:03	1/2	!PIM	210	0020

source-list pointer	Source IP address
0010	141.28.2.1
0020	155.33.2.11

  This causes traffic of both groups to interfere with each other. There is currently **no** general workaround available for this problem but the usage of GLOP addressing in 233/8 helps to prevent this situation (see section 8.3)

- Multicast containment using PIM-snooping is dependent on the amount of available CAM memory because every PIM Join message creates state information in the switch. If the CAM memory is exhausted the switch cannot process any further PIM Join requests unless they create **no** additional state information (like periodic PIM Join messages used to reset PIM state timers, see section 4.3.6)

  If CAM memory (for any reason) gets exhausted Foundry Networks suggested to revert to the following method in order to prevent a “core switch meltdown”:

  As long as CAM memory is exhausted, **all** multicast traffic is flooded to **all** ports within the Multicast VLAN where PIM neighbors have been detected (see section 6.3.3.1)

  Note: Additionally a Simple Network Management Protocol (SNMP) [53] trap should be generated to inform an administrator about this condition.

SSM Specific Restrictions

In the Source Specific Multicast (SSM) model the switch has to do essentially the same as in the ISM model. But there are is a very important difference:

Forwarding decisions in ISM are based **only** on the multicast group address but SSM forwarding decisions **must** be based on the source- and multicast group address.

For example a multicast source on a host with IP 141.1.1.1 which is sending to multicast group 224.2.2.2 is uniquely identified by its multicast group address in the ASM model.

In the SSM model a multicast source is uniquely identified by its source **and** multicast group address (141.1.1.1, 224.2.2.2).

This causes some problems if the central switch fabric has to use PIM Join/Prune messages to maintain it's FIB because it has to know if it is a "SSM Join/Prune" or a "ISM Join/Prune". To distinguish between these two types the switch should analyze the multicast group destination address in the PIM Join/Prune message:

If the multicast group address is in the IP range 232/8 (232.0.0.0 to 232.255.255.255) then it is considered to be a "SSM Join/Prune" message and the switch should use the source IP address and the multicast group address (S,G) as a unique identifier to maintain its FIB entry. Otherwise the switch forwards the frame as described in section 6.3.3.2)

7 Security Considerations

7.1 Impact of Denial of Service Attacks

Denial of Service (DoS) attacks or Distributed Denial of Service (DDoS) attacks can cause serious instability (or a complete melt down) of the MIX topology and/or the adjacent PMBRs or even intradomain multicast routers.

Holbrook [17] states that in SSM the channel subscription request creates (S,G) state (in ASM also (*,G) state) to record the subscription and that this normally causes subsequent processing in a neighboring PIM router. A host can therefore simply launch a DoS attack by requesting a large number of channel or group (see section 2.4.2) subscriptions. Holbrook also states that a denial of service can result if:

- *a large amount of traffic arrives when it was otherwise undesired, consuming network resources to deliver it and host resources to drop it.* (Annotation: WAN/LAN circuits get overloaded)

- *a large amount of source-specific multicast state is created in network routers, using router memory and CPU resources to store and process the state* (Annotation: this also applies to the ASM service model as described in section 2.4.1)

- *a large amount of control traffic is generated to manage the source-specific state, using router CPU and network bandwidth* (Annotation: this also applies to the ASM service model as described in section 2.4.1)

The following sections describe multicast DoS and DDoS attacks that have either already been targeted to various multicast enabled networks or could cause denial of service in the future.

7.2 RAMEN Worm

The RAMEN worm [54] causes the creation of huge amounts of MSDP SA messages and (S,G)/(*,G) state in multicast enabled networks and therefore causes MSDP enabled routers to be the target of the attack. The worm works as follows:

The worm runs on a host and starts to scan a portion of the multicast address space (~ 64,000 groups addresses) (see section 2.7) by sending multicast packets to each scanned multicast address. If the hosts DR is a PIM router it encapsulates these packets in PIM register messages (see section 4.3.4) and unicasts them to the RP. All PIM register messages (for each scanned multicast address) cause the RP to create MSDP SA messages that are flooded to all MSDP peers in other domains. These MSDP peers in turn also flood MSDP SA messages to their peers because MSDP SA messages are stored throughout the Internet. This chain reaction could break all MSDP peers in the whole MSDP mesh.

The results of a RAMEN worm DoS attack can be quite serious. Eubanks described the impact of a RAMEN worm attack that was launched on one of his routers as follows [55]:

> *"... We have gotten 15,000 SA's a minute. Dealing with these can melt down routers. (We had to reboot a Cisco 7204, for example, which apparently either filled up or fragmented its memory beyond usability.)*
>
> *I think it is fair to say that the question of rate limiting and other DOS filtering in PIM/SSM/MSDP multicast is getting serious attention now."*

The impact of MSDP SA creation on the core switch at the MIX is not critical because the core switch does not store any MSDP SA messages. In fact MSDP uses standard unicast peering sessions (see section 5.2.3) and therefore the amount of additional unicast traffic on the core switch should be negligible. On the other hand the impact on PMBRs (connected to the MIX) and intradomain PIM routers (RPs/MSDP peers) can (according to Eubanks) be very serious.

Note: The impact of PIM state generation by a DoS/DDoS attack in PMBRs that are peering at a MIX can melt down the PMBRs and the MIX (see section 7.4)

7.3 Spoofing Intraprotocol Messages

Another kind of attack exploits the possibility of **spoofing** all kinds of PIM **intraprotocol messages** (e.g. PIM hello, Join/Prune, Assert etc.). For example, a host could pretend to be a PIM neighbor by sending PIM hello messages on the subnet. Then the host could start to send PIM Join messages for an arbitrary number of multicast groups and sources. This would cause PIM (S,G) or (*,G) creation in neighboring PIM routers (comparable to RAMEN worm → see section 7.2) and in **all** routers in a domain or even other domains which could also affect a MIX that interconnects two domains/PMBRs because it also has to keep forwarding state information in its CAM tables (see section 7.4).

7.4 Critical MIX DDoS Attack

The following DDoS scenario could cause a denial of service at a MIX and precautions have to be taken that this kind of attack will not disrupt the MIX service.

This scenario uses the same domain configuration as shown in Figure 28 section 6.3.3.4:

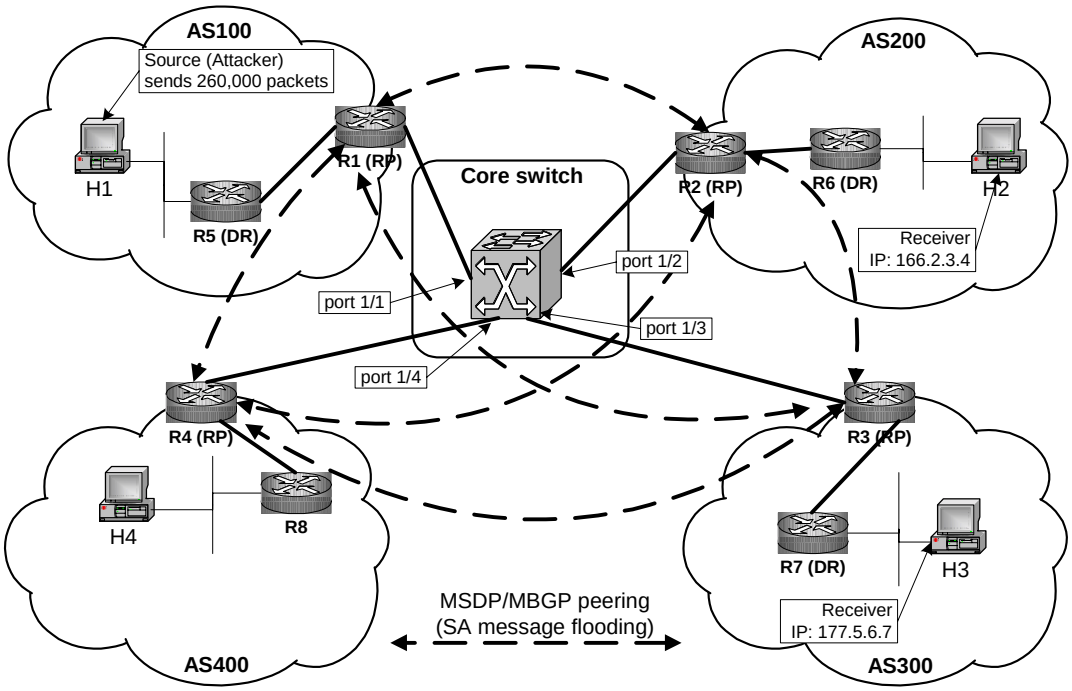

Figure 30: MIX critical DDoS attack scenario

Assume that the attacker uses H1 in AS100 as a multicast source and starts sending just **one** multicast packet to all groups in the multicast address range 224.2.0.0 to 224.5.255.255 (~260,000 groups).

The attacker coordinates the reception of **all** these groups with a receiver in a **different** domain (in the example by H2 in AS200).

The following happens:

1. H1 starts to send multicast packets. This causes MSDP SA messages to be flooded to MSDP peers that could immediately melt down the originating MSDP router and all peers. So far this has **no** serious effects on the MIX core switch yet (see section 7.2)
2. H2 sends IGMP membership reports for the same groups H1 is sending multicast traffic. This first creates a ST between the DR of H2 (R6) and the RP (R2) and afterwards a SPT from the source to the receiver
3. The RP “knows” (by way of MSDP SA messages) that there is a source for these groups in AS100 and sends PIM (S,G) Join messages towards the source in AS100
4. These PIM Joins traverse the “PIM-snooping” enabled MIX core switch that updates its CAM table with the multicast group MAC address and the source-list with the IP of the source
5. If the core switch receives thousands of PIM Joins and therefore has to store thousands of multicast MAC addresses in its CAM table this could **break the core switch** or could at least negatively affect the performance

7.5 Denial of Service Prevention

The **stability** and therefore the **availability** of the MIX is the most important criteria for all MIX participants. Any kind of attack launched specifically targeted to the MIX directly affects all participants. Therefore it is crucial to protect the MIX from these kinds of attacks.

The following features could help to prevent inter- or intradomain denial of service:

Router features

- *A router SHOULD verify that the source of a subscription request is a valid address for the interface on which it was received. Failure to do so would exacerbate a spoofed-source address attack.* [17]
- *The total rate at which all hosts on any one interface are allowed to initiate subscriptions (to limit the damage caused by forged source-address attacks)* [17]
- *The total number of subscriptions that can be initiated from any single interface or host.* [17]
- Configure router interfaces to **explicitly not** accept any other PIM neighbors (PIM hello messages) and subsequently not accept any spoofed PIM (S,G) or (*,G) Join messages
- Limit the **number** of joins per time period that can be received from a specific source and/or receiver for
    - a specific (S,G) pair
    - a (S) and/or (G) address range
    - Globally
- A router configured as RP/MSDP peer should provide MSDP rate limits

Rate limiting of protocol traffic could prevent attacks but on the other hand Holbrook [17] notes that

> *"Any decision by an implementor to artificially limit the rate or number of subscriptions should be taken carefully, however, as future applications may use large numbers of channels. Tight limits on the rate or number of channel subscriptions would inhibit the deployment of such applications."*

Therefore it is not always advisable to rate limit any kind of administrative protocol traffic.

Note: General strategies to protect (routers) against DDoS attacks can be found on the Cisco website [56]

8 MIX Implementation

This chapter describes requirements and recommendations that must/should be adhered to provide a reliable and scaleable MIX service.

8.1 Requirements

8.1.1 Core Switch

- The multicast containment PIM-snooping **must** be enabled in order restrict multicast traffic flooding and provide a scaleable MIX (see RGMP/ PIM-snooping benefits in section 6.3.1)
- All multicast packets with a destination IP in the range 224.0.0/24 (administrative multicast traffic) **must be flooded** to all active switch ports within the Multicast VLAN. Restriction of flooding these packets would break protocol requirements like e.g. PIM neighbor discovery.

8.1.2 PMBRs

- The interface facing the MIX must be configured for PIM-SM **only** in order to prevent flooding of multicast traffic onto the MIX.

  The participant has to obtain an IP address for the interface on the MIX from the MIX administrator. This is usually one IP out of a /24 network that is reserved for all PMBRs connected to the MIX

- A PMBR connected to the MIX has to be configured for
    - PIM-SM version 2 on the interface towards the MIX
    - Restrict administratively scoped traffic from entering/leaving the domain
    - BGP distances are all the same (to preserve the PIM assert mechanism)

  For an example Cisco router configuration see Appendix.

8.2 Recommendations

8.2.1 Core Switch

- In case the PIM-snooping feature breaks (e.g. because of a bug in the PIM-snooping code) flooding of multicast traffic onto **all** switch ports could occur. To prevent this situation a port based Multicast VLAN should be created and only ports that connect PMBRs should be members of this VLAN. This prevents multicast traffic to interfere with unicast traffic exchanged on the same core switch.
- It should be tested if a separate VLAN for MBGP peering and PIM-SM control messages exchange could be created to keep actual multicast traffic and control traffic separate to guarantee that control messages will always flow timely.
- A thoroughly tested and stable switch code version has to be installed

- The size of the CAM table has to be large enough to hold all multicast MAC/port entries to perform fast-path multicast forwarding. Additionally enough memory to hold all source associations have to be provided.

  In case of running out of CAM table memory the switch could revert to flood multicast traffic to all ports where **PIM neighbors** where detected as long as CAM table memory is being available again.

- There following features should be available to ensure core switch operation and to inform the MIX administrator:
  - set a maximum number of G entries in the CAM table (per port, globally)
  - set a maximum number of Source entries (per port, globally)
  - set a maximum amount of CAM memory to be used (in % of total) before revert to flooding of multicast traffic
  - set a maximum multicast traffic limit in bits/second (bps) → per port, per trunk egress or ingress based)
  - If one these limits is exceeded an SNMP trap should be sent.

  Williamson [13] notes a very important fact about rate-limiting of broadcast/multicast traffic:

  "As it turns out, rate-limiting is a really bad idea as the arbitrary dropping of certain types of broadcast frames can result in network instability that in some cases can be bad enough to melt down the network. For example, bridge protocol data units (BPDUs) are multicast to the special All Bridges multicast MAC address. If enough of these BPDUs are discarded, the network can suffer instabilities as the spanning tree algorithm constantly attempts to converge. As the spanning tree algorithm in the switch tries to converge, more BPDUs may be lost, leading to a network meltdown. … (My personal motto: 'Just say no to MAC Layer broadcast/multicast rate limiting.')"

- Multicast traffic is best-effort delivery by default. A simple Quality of Service (QoS) switch feature could provide a premium multicast service to MIX participants. The idea is to create a hash on the least significant octet of a multicast destination IP address (if GLOP addressing is used this octet can be chosen by the owner of the specific ASN), apply modulo 8 (if a switch has 8 priority queues) and put the frame into the resulting priority queue for forwarding e.g.
  224.2.2.**12** mod 8 → priority queue 4 or
  224.2.2.**78** mod 8 → priority queue 6.
- The current PIM-snooping implementation on Foundry switches only forwards traffic based on destination multicast MAC address without regard to the source address. This should be changed to source and group based forwarding to support SPTs and therefore SSM (S,G) based forwarding specifically (this is already planned for future releases of Foundry "PIM-snooping" switch code)
- In order to use load balancing over trunked switch ports it is necessary to use an adequate load balancing algorithm for multicast traffic. Multicast traffic patters are mainly flows identified by their source and group address. In order to preserve their packet order (which is crucial for stream oriented traffic patterns) within each flow it is recommended to use a trunk algorithm that uses one physical link of a trunk per multicast flow. Therefore the load balancing algorithm should consist of a hash function on the source and destination IP address. This would guarantee that flows with the same source and destination IP address always use the same physical link.

8.2.2 PMBRs

PMBR "multi-homing"

In order to get a higher resilience the participants at a MIX have sometimes the possibility to connect their BRs to two physically different switches (multi-homing). E.g. members of the LINX [4] have the choice of connecting their BRs to the unicast exchange via the primary (core) switch and/or, with a separate router interface, to a secondary switch (from a different switch vendor) for enhanced resilience. It should be testes if, and how PMBRs can be multi-homed to one switch (on two different slots) or on two separate switches.

MIX PMBR configuration example

A basic PMBR configuration based on the following setup can be found in the Appendix.

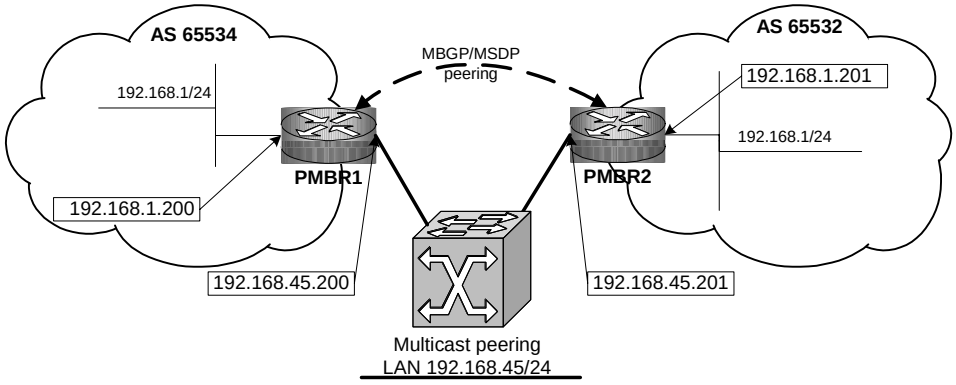

Figure 31: MIX PMBR configuration example

Additional information on how to configure multicast on Cisco routers can be found on the Cisco website [57].

8.2.3 Management

PIM MIB

In order to manage the MIX core switch it should provide a Management Information Base (MIB) specifically for PIM-snooping related information like

- Current status of multicast forwarding based on CAM table entries (destination MAC address/reception port/source address list)
- Current PIM neighbor list
- Statistics about ingress/egress amount of non-administrative multicast traffic
- Statistics about ingress/egress amount of administrative multicast traffic (forwarded to the CPU)
- All configured rate-limits or other threshold values

The PIM Protocol Independent Multicast MIB for IPv4 [58] could be used as a template for MIB objects that are needed to retrieve the information stated above.

Some PIM MIB entries should be available (read-only) to all MIX participants in order to diagnose multicast forwarding problems. This information could be provided on a website which requires the user to login in order to restrict access to the PIM MIB entries (this already worked in a test setup with the Foundry BI8000).

8.3 General notes

Impact of Multicast MAC address ambiguity

Additionally the use of Class D addresses in the following ranges must be avoided because if these addresses map to multicast MAC addresses they result in link-local ambiguous multicast MAC addresses which are always flooded within a given Multicast VLAN.

- 224.0.0.x to 239.0.0.x
  (e.g. 224.0.0.x, 225.0.0.x, 226.0.0.x, …)
- 224.128.0.x to 239.128.0.x
  (e.g. 224.128.0.x, 225.128.0.x, 226.128.0.x, …)

where x is in the range from 0 to 255.

Using GLOP Multicast Address Allocation at a MIX

There is currently no explicit MIX address allocation scheme available that would be useful to prevent multicast MAC ambiguity to occur on the MIX. However, if the GLOP Addressing scheme [32] is used by all MIX participants this (currently) eliminates MAC ambiguity. The reason for that is that currently only 15 bits out of the 16 available bits in the ASN are globally used (ASN <= 32767). Therefore the high order bit of the first octet is **always** 0 and does not get lost in the mapping process. This situation only applies as long as no ASNs > 32767 are assigned and SSM sources would use a similar global addressing scheme.

9 Conclusions

Tests of the PIM-snooping multicast containment feature implemented by Foundry on a BI8000 switch showed that it will provide the necessary functionality that is needed at a MIX. It could not be tested (in our labs) that PIM-snooping will work and scale if used on a single core switch or even in a trunked Layer 2 MIX topology because we did not have the necessary amount of switches (at least three BI8000), routers and multicast sources/receivers. However, tests conducted by Foundry in their labs showed that a network topology that consisted of three interconnected BI8000 switches configured with one Multicast VLAN and PIM-snooping enabled works. If PIM-snooping is able to scale in a “real” MIX topology with e.g. 50 PMBRs connected to the MIX exchanging “real” multicast traffic could not be tested but must be tested before a MIX should be deployed in a production environment.

The current implementation of the PIM-snooping code and architecture of CAM tables could cause serious problems in case ambiguous multicast MAC addresses are used by MIX participants. This problem could circumvent the PIM-snooping multicast containment which again would cause the switch to forward multicast traffic onto ports where receivers for a “MAC ambiguous” multicast address exist but who don’t need to receive the traffic. The use of GLOP addressing in 233/8 by all MIX participants could eliminate this problem as long as ASNs <= 32767 are used but this can just be considered as an interim solution for the multicast MAC ambiguity problem. A long term solutions would be to base forwarding decisions not on destination MAC addresses but on the source and/or destination IP addresses depending on the multicast service model that is used.

The Reliability of the MIX is a very important aspect. Tests of the current PIM-snooping code showed that it still contains a lot of bugs which first have to be eliminated before the PIM-snooping feature can be used in a production environment. Another serious impact on the reliability of a core switch can be that it has to keep forwarding state information which uses

switch memory resources. If these resources get exhausted the switch could stop forwarding traffic. It should further be investigated if this could be solved by applying specific threshold values or rate-limiting features to stop the exhaustion of resources before the core switch breaks. This is especially important in order to protect the core switch from DoS/DDoS attacks that could either be launched directly to the switch or to MIX participants.

The management of the MIX topology with currently available MIB information is not optimal. The MIX core switch should provide a PIM MIB that contains all information about multicast related features. This information should be partly provided to the MIX participants for them to diagnose multicast problems in the network.

In order to further improve the scalability and reliability of new and existing MIXs throughout the Internet it should be considered to write an Internet-Draft about multicast containment at a MIX using PIM-snooping in order to lead this new technology towards a possible standard.

10 Appendix

Example MIX PMBR configuration:

```
Config of test-router1 (PMBR1)
(internal IP: 192.168.1.200 / MIX IP: 192.168.45.200)
!
version 12.1
!
hostname test-router1
ip subnet-zero
no ip finger
# enable multicast routing
ip multicast-routing
interface Loopback0
 ip address 10.10.10.10 255.255.255.255
 ip pim sparse-dense-mode
 ip sap listen
!
# interface on the internal network
interface FastEthernet0/0
 description Connection to internal network
 ip address 192.168.1.200 255.255.255.0
 ip igmp query-interval 10
 ip igmp version 2
 ip igmp query-max-response-time 3
 ip sap listen
 duplex full
!
# interface on the MIX
interface FastEthernet3/0
 ip address 192.168.45.200 255.255.255.0
 ip pim bsr-border
 ip pim sparse-mode
 ip multicast boundary 1
 duplex half
!
autonomous-system 65534
!
router bgp 65534
 no synchronization
 bgp log-neighbor-changes
 network 192.168.45.0
 network 192.168.59.0
 neighbor 192.168.1.12 remote-as 65533
 neighbor 192.168.1.12 send-community
 neighbor 192.168.1.12 distribute-list filter-all-routes in
 neighbor 192.168.1.12 filter-list 1 out
 neighbor 192.168.45.201 remote-as 65532
!
# configuration of MBGP peering neighbors and default
# bgp distance in order to preserve the PIM assert process
# Note: this is the new format of bgp neighbor definition
 address-family ipv4 multicast
 neighbor 192.168.45.201 activate
 distance bgp 20 80 80
# announced networks
```

```
 network 141.28.0.0
 network 141.29.0.0
 network 141.30.0.0
 network 141.31.0.0
 network 192.168.1.0
 network 192.168.45.0
 exit-address-family
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.1.254
ip route 141.28.0.0 255.255.0.0 Loopback0
ip route 141.29.0.0 255.255.0.0 Loopback0
ip route 141.30.0.0 255.255.0.0 Loopback0
ip route 141.31.0.0 255.255.0.0 Null0
ip route 192.168.59.0 255.255.255.0 Null0
ip route 207.20.85.171 255.255.255.255 192.168.1.254
# configure RP announcements and discovery on Loopback0
ip pim send-rp-announce Loopback0 scope 16
ip pim send-rp-discovery scope 16
# configure the MSDP peering session
ip msdp peer 192.168.45.201 connect-source FastEthernet3/0
ip msdp cache-sa-state
!
!
# configure the a basic boundary list for
# Cisco Auto-RP announcements (.39 and .40)
# and administratively scoped addresses (239/8)
# multicast packets with these destination address should
# not leave our domain
ip access-list standard filter-all-routes deny any
access-list 1 deny   224.0.1.39
access-list 1 deny   224.0.1.40
access-list 1 deny   239.0.0.0 0.255.255.255
access-list 1 permit any
# restrict other administrative intradomain multicast traffic
# from leaving our domain
access-list 111 deny   ip any host 224.0.2.2
access-list 111 deny   ip any host 224.0.1.2
access-list 111 deny   ip any host 224.0.1.3
access-list 111 deny   ip any host 224.0.1.22
access-list 111 deny   ip any host 224.0.1.24
access-list 111 deny   ip any host 224.0.1.35
access-list 111 deny   ip any host 224.0.1.39
access-list 111 deny   ip any host 224.0.1.60
access-list 111 deny   ip any host 224.0.1.40
access-list 111 deny   ip any 239.0.0.0 0.255.255.255
access-list 111 deny   ip 10.0.0.0 0.255.255.255 any
access-list 111 deny   ip 127.0.0.0 0.255.255.255 any
access-list 111 permit ip any any
!
end
```

Config of test-router2 (PMBR2)
(internal IP: 192.168.1.201 / MIX IP: 192.168.45.201)

```
!
version 12.0
!
hostname test-router2
ip subnet-zero
ip multicast-routing
!
interface Loopback0
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 ip sdr listen
!
interface FastEthernet1/0
 description Connection to MIX
 ip address 192.168.45.201 255.255.255.0
 no ip directed-broadcast
 ip pim bsr-border
 ip pim sparse-mode
 ip multicast boundary 1
!
autonomous-system 65532
!
router bgp 65532
 neighbor 192.168.1.12 remote-as 65533
 neighbor 192.168.45.200 remote-as 65534 nlri multicast
!
ip classless
ip route 192.168.61.0 255.255.255.0 Null0
ip msdp peer 192.168.45.200 remote-as 65534
ip msdp description 192.168.45.200 MSDP peer with AS65534
ip msdp sa-filter in 192.168.45.200 list 111
ip msdp sa-filter out 192.168.45.200 list 111
ip msdp cache-sa-state
!
access-list 1 deny   224.0.1.39
access-list 1 deny   224.0.1.40
access-list 1 deny   239.0.0.0 0.255.255.255
access-list 1 permit any
access-list 111 deny   ip any host 224.0.2.2
access-list 111 deny   ip any host 224.0.1.2
access-list 111 deny   ip any host 224.0.1.3
access-list 111 deny   ip any host 224.0.1.22
access-list 111 deny   ip any host 224.0.1.24
access-list 111 deny   ip any host 224.0.1.35
access-list 111 deny   ip any host 224.0.1.39
access-list 111 deny   ip any host 224.0.1.40
access-list 111 deny   ip any host 224.0.1.60
access-list 111 deny   ip any 239.0.0.0 0.255.255.255
access-list 111 deny   ip 10.0.0.0 0.255.255.255 any
access-list 111 deny   ip 127.0.0.0 0.255.255.255 any
access-list 111 permit ip any any

end
```

11 Glossary

AFI	Address Family Identifier
AS	Autonomous System
ASIC	Application-Specific Integrated Circuit
ASM	Any-Source Multicast
ASN	Autonomous System Number
ASP	Application Service Provider
BGP	Border Gateway Protocol
BGP-4	Border Gateway Protocol version 4
BGP4+	see MBGP
BPDU	Bridge Protocol Data Unit
BR	Border Router
CAM	Content Addressable Memory
CGMP	Cisco Group Management Protocol
CIDR	Classless Inter-Domain Routing
CPU	Central Processing Unit
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DLL	Dynamic Link Library
DoS	Denial of Service
DR	Designated Router
DVMRP	Distance Vector Multicast Routing Protocol
EBGP	Exterior Border Gateway Protocol
FIB	Forwarding Information Base
GigE	Gigabit-Ethernet
GLOP	no acronym → see [32]

IANA	Internet Assigned Numbers Authority
IBGP	Interior Border Gateway Protocol
IBX	Internet Business Exchange™
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IGMPv1	Internet Group Management Protocol version 1
IGMPv2	Internet Group Management Protocol version 2
IGMPv3	Internet Group Management Protocol version 3
IGP	Interior Gateway Protocol
IGRP	Interior Gateway Routing Protocol
IOS	Internet Operating System
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IPX	Inter Packet Exchange
ISM	Internet Standard Multicast
ISP	Internet Service Provider
LAN	Local Area Network
LINX	London Internet Exchange
MAC	Media Access Control
MBGP	Multiprotocol extensions for BGP-4
MIB	Management Information Base
MIX	Multicast Internet Exchange
MOSPF	Multicast Extensions to Open Shortest Path First (OSPF)
MRIB	Multicast Routing Information Base

MSDP	Multicast Source Discovery Protocol
NIC	Network Interface Card
NLRI	Network Layer Reachability Information
OSPF	Open Shortest Path First
PIM	Protocol Independent Multicast
PIM-DM	Protocol Independent Multicast – dense mode
PIM-SM	Protocol Independent Multicast – sparse mode
PIM-SSM	Protocol Independent Multicast – Source Specific multicast
PMBR	Protocol Independent Multicast Border Router
QoS	Quality of Service
RFC	Request for Comments
RGMP	Router-Port Group Management Protocol
RIP	Routing Information Protocol
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
SA	Source Active
SAFI	Subsequent Address Family Identifier
SAP	Session Announcement Protocol
SDP	Session Description Protocol
SDR	Session Directory Revised
SNMP	Simple Network Management Protocol
SPT	Shortest-Path Tree
SSM	Source-Specific Multicast
ST	Shared Tree
TCP	Transmission Control Protocol

TCP/IP	Transmission Control Protocol/Internet Protocol
TTL	Time-To-Live
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network
WAN	Wide Area Network

12 Bibliography

[1] Equinix Inc., Mountain View, USA
http://www.equinix.com

[2] University of Applied Sciences (Fachhochschule), Furtwangen, Germany
http://www.fh-furtwangen.de

[3] Equinix Inc., Mountain View, USA – internal website
http://nemo.corp.equinix.com/research/sandbox/EquinixSandboxMission.doc

[4] The London Internet Exchange, Peterborough, UK
http://www.linx.net

[5] “Deploying a Multicast Internet Exchange (MIX) at the London Internet Exchange Ltd (LINX) – technical documentation”, Besch, G., Busch, T., Osswald, R., Weinmann, T., The London Internet Exchange Ltd., February 2000

[6] De Montfort University, Leicester, UK
http://www.dmu.ac.uk

[7] Cisco Systems, San Jose, CA, USA
http://www.cisco.com

[8] “Router-Port Group Management Protocol”, Cisco, January 2001
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s10/dtrgmp.htm

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

[10] “How IP Multicast Works”, Johnson, V., Johnson, M., IP Multicast Initiative (IPMI), October 1997
http://www.ipmulticast.com/community/whitepapers/howipmcworks.html

[11] Internet Assigned Numbers Authority
http://www.iana.org

[12] “User Datagram Protocol”, Postel, J., RFC 768, August 1980

[13] “Developing IP Multicast Networks, Volume I”, Williamson, B., Cisco Press, 2000, ISBN: 1-57870-077-9

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

[15] “RTP Profile for Audio and Video Conferences with Minimal Control”, Schulzrinne, H., RFC 1890, January 1996

[16] “Transmission Control Protocol”, Postel, J., RFC 761, January 1980

[17] “Source-Specific Multicast for IP”, Holbrook, H., Cain, B., draft-holbrook-ssm-arch-02.txt, March 2001

[18] “Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)”, Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., draft-ietf-pim-sm-v2-new-02.txt, March 2001

[19] “Astrogram”, NASA Ames Research Center Employee Newspaper, Moffett Field, CA, USA, June 1999 http://amesnews.arc.nasa.gov/astrogram/astrostories/060799/Internet.html

[20] ”Distance Vector Multicast Routing Protocol”, Waitzman, D., Partridge, C., Deering, S., RFC 1075, November 1988

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

[22] “Internet Group Management Protocol, Version 3”, Cain, B., Deering, S., Fenner, B., Kouvelas, I., Thyagarajan, A., Work in progress, draft-ietf-idmr-igmp-v3-07.txt, March 2001

[23] “Sprint Labs IGMPv3 Multicast Implementation for Linux”, Sprint Labs, http://www.sprintlabs.com/Department/IP-Interworking/multicast/linux-igmpv3/

[24] “Host Implementation of IGMPv3 on FreeBSD”, Wilbert de Graaf, http://home.hetnet.nl/~wilbertdg/igmpv3.html

[25] “Interdomain Multicast Solutions Using SSM”, Cisco System, http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/mcst_p1/mcst_p2.htm

[26] “Assigned Numbers”, Reynolds, J., Postel, J., RFC 1700, October 1994 (http://www.iana.org/assignments/multicast-addresses)

[27] “Administratively Scoped IP Multicast”, Meyer, D., RFC 2365, July 1998

[28] “Address Allocation for Private Internets”, Rekhter, Y, Moskowitz, B., Karrenberg, D., de Groot, G. J., Lear, E., RFC 1918, February 1996

[29] “Session Directory”, University College London, UK http://www-mice.cs.ucl.ac.uk/multimedia/software/sdr/

[30] “Session Announcement Protocol”, Handley, M., Perkins, C., Whelan, E., RFC 2974, October 2000

[31] “SDP: Session Description Protocol”, Handley, M., Jacobson, V., RFC 2327, April 1998

[32] “GLOP Addressing in 233/8”, Meyer, D., Lothberg, P., RFC 2770, February 2000

[33] “Multicast FAQ File”, Multicast Technologies, VA, USA http://www.multicasttech.com/faq/multicast_faq.html

[34] “Glop Calculator for Group Range 233/8”, University of Oregon, http://gigapop.uoregon.edu/glop/

[35] “Broadcast Protocols in Packet Switched Computer Networks (Digital Systems Laboratory, Dept. of Electrical Engineering”, Dalal, Y., Stanford University, 1977

[36] “Internet Protocol (IP) Multicast”, Cisco, June 1999 http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ipmulti.htm

[37] “Multicast Extensions to OSPF”, Moy, J., RFC 1584, March 1994

[38] "Protocol Independent Multicast Version 2 – Dense Mode Specification”, Deering, S., Estrin, D., Farinacci, D., Jacobson, V., Helmy, A., Meyer, D., Wei, L., draft-ietf-pim-v2-dm-03.txt, June 1999

[39] “Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification”, Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., Liu, C., Sharma, P., Wei, L., RFC 2362, June 1998

[40] “OSPF Version 2”, Moy, J., RFC 2328, April 1998

[41] “Multiprotocol Extensions for BGP-4”, Bates, T., Rekhter, Y., Chandra, R., Katz, D., RFC 2858, June 2000

[42] “RIP Version 2”, Malkin, G., RFC 2453, November 1998

[43] “PIM-SM Multicast Routing Protocol”, White Paper, Microsoft Corporation, http://www.microsoft.com/windows2000/library/unzippeddocs/pimsm2.doc

[44] “Guidelines for creation, selection, and registration of an Autonomous System (AS)”, Hawkinson, J., Bates, T., RFC 1930, March 1996

[45] “Multicast-Friendly Internet Exchange (MIX)”, LaMaster, H., Shultz, S., Meylor, J., Meyer, D., draft-ietf-mboned-mix-01.txt, June 1999 http://community.roxen.com/developers/idocs/drafts/draft-ietf-mboned-mix-01.html

[46] "A Border Gateway Protocol 4 (BGP-4)", Rekhter, Y., Li T., RFC 1771, March 1995

[47] “Multicast Source Discovery Protocol (MSDP)”, Meyer, D., Fenner, B., draft-ietf-msdp-spec-10.txt, May 2001

[48] The Internet Engineering Task Force (IETF)
http://www.ietf.org

[49] “Router-port Group Management Protocol”, Eckert, T., November 2000
http://cph.telstra.net/ietf/old-ids/draft-wu-rgmp-00.txt

[50] “Multicast in a Campus Network: CGMP and IGMP Snooping”, Cisco,
http://www.cisco.com/warp/public/473/22.html

[51] Foundry Networks, Santa Clara, CA, USA
http://www.foundrynetworks.com

[52] “Release Notes for IronWare SoftwareRelease 07.2.02”, Foundry Networks, Santa Clara, CA, USA
http://www.foundrynet.com/cgi-bin/support.cgi?A=download&B=1EQU3466&C=releasenotes&D=RelNotes07202.pdf

[53] "Simple Network Management Protocol", Case, J., Fedor, M., Schoffstall, M., and J. Davin, RFC 1157, STD 15, MIT Laboratory for Computer Science, May 1990

[54] “Ramen Internet Worm Analysis”, Max Vision’s Whitehats,
http://www.whitehats.com/library/worms/ramen/

[55] “RAMEN worm”, email from: Marshall Eubanks <tme@21rst-century.com> To: nanog@merit.edu, Fri, 19 Jan 2001 09:43:18

[56] “Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks”, Cisco Systems,
http://www.cisco.com/warp/public/707/newsflash.html

[57] “Multicast Quick-Start Configuration Guide”, Cisco,
http://www.cisco.com/warp/public/105/48.html

[58] “Protocol Independent Multicast MIB for IPv4”, Farinacci, D., Fenner, B., Thaler, D., RFC 2934, October 2000