IP Multicast Configuration Guide, Cisco IOS XE 17.x

First Published: 2022-03-11

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Audience and Scope

This document is designed for the person who is responsible for configuring your Cisco Enterprise router. This document is intended primarily for the following audiences:

- Customers with technical networking background and experience.
- System administrators familiar with the fundamentals of router-based internetworking but who might not be familiar with Cisco IOS software.
- System administrators who are responsible for installing and configuring internetworking equipment, and who are familiar with Cisco IOS software.

Feature Compatibility

For more information about the Cisco IOS XE software, including features available on your device as described in the configuration guides, see the respective router documentation set.

To verify support for specific features, use the Cisco Feature Navigator tool. This tool enables you to determine the Cisco IOS XE software images that support a specific software release, feature set, or a platform.

Document Conventions

This documentation uses the following conventions:

Convention	Description
^ or **Ctrl**	The ^ and **Ctrl** symbols represent the Control key. For example, the key combination ^**D** or **Ctrl-D** means hold down the **Control** key while you press the **D** key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

The command syntax descriptions use the following conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter exactly as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example, see the following table.

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`bold screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. Exclamation points are also displayed by the Cisco IOS XE software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials that may not be contained in this manual.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

Troubleshooting

For the most up-to-date, detailed troubleshooting information, see the Cisco TAC website at https://www.cisco.com/en/US/support/index.html.

Go to **Products by Category** and choose your product from the list, or enter the name of your product. Look under **Troubleshoot and Alerts** to find information for the issue that you are experiencing.

PART I

IP Multicast Overview

CHAPTER 1

IP Multicast Technology Overview

IP multicast is a bandwidth-conserving technology that reduces traffic by delivering a single stream of information simultaneously to potentially thousands of businesses and homes. Applications that take advantage of multicast include video conferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news.

This module contains a technical overview of IP multicast. IP multicast is an efficient way to use network resources, especially for bandwidth-intensive services such as audio and video. Before beginning to configure IP multicast, it is important that you understand the information presented in this module.

Information About IP Multicast Technology

Role of IP Multicast in Information Delivery

IP multicast is a bandwidth-conserving technology that reduces traffic by delivering a single stream of information simultaneously to potentially thousands of businesses and homes. Applications that take advantage of multicast include video conferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news.

IP multicast routing enables a host (source) to send packets to a group of hosts (receivers) anywhere within the IP network by using a special form of IP address called the IP multicast group address. The sending host inserts the multicast group address into the IP destination address field of the packet and IP multicast routers and multilayer switches forward incoming IP multicast packets out all interfaces that lead to the members of the multicast group. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message.

Multicast Group Transmission Scheme

IP communication consists of hosts that act as senders and receivers of traffic as shown in the first figure. Senders are called sources. Traditional IP communication is accomplished by a single host source sending packets to another single host (unicast transmission) or to all hosts (broadcast transmission). IP multicast provides a third scheme, allowing a host to send packets to a subset of all hosts (multicast transmission). This subset of receiving hosts is called a multicast group. The hosts that belong to a multicast group are called group members.

Multicast is based on this group concept. A multicast group is an arbitrary number of receivers that join a group in order to receive a particular data stream. This multicast group has no physical or geographical boundaries--the hosts can be located anywhere on the Internet or on any private internetwork. Hosts that are interested in receiving data from a source to a particular group must join that group. Joining a group is accomplished by a host receiver by way of the Internet Group Management Protocol (IGMP).

In a multicast environment, any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group can receive packets sent to that group. Multicast packets are delivered to a group using best-effort reliability, just like IP unicast packets.

In the next figure, the receivers (the designated multicast group) are interested in receiving the video data stream from the source. The receivers indicate their interest by sending an IGMP host report to the routers in the network. The routers are then responsible for delivering the data from the source to the receivers. The routers use Protocol Independent Multicast (PIM) to dynamically create a multicast distribution tree. The video data stream will then be delivered only to the network segments that are in the path between the source and the receivers.

IP Multicast Routing Protocols

The software supports the following protocols to implement IP multicast routing:

- IGMP is used between hosts on a LAN and the routers on that LAN to track the multicast groups of which hosts are members.
- Protocol Independent Multicast (PIM) is used between routers so that they can track which multicast packets to forward to each other and to their directly connected LANs.

This figure shows where these protocols operate within the IP multicast environment.

IP Multicast Group Addressing

A multicast group is identified by its multicast group address. Multicast packets are delivered to that multicast group address. Unlike unicast addresses that uniquely identify a single host, multicast IP addresses do not identify a particular host. To receive the data sent to a multicast address, a host must join the group that address identifies. The data is sent to the multicast address and received by all the hosts that have joined the group indicating that they wish to receive traffic sent to that group. The multicast group address is assigned to a group at the source. Network administrators who assign multicast group addresses must make sure the addresses conform to the multicast address range assignments reserved by the Internet Assigned Numbers Authority (IANA).

IP Class D Addresses

IP multicast addresses have been assigned to the IPv4 Class D address space by IANA. The high-order four bits of a Class D address are 1110. Therefore, host group addresses can be in the range 224.0.0.0 to 239.255.255.255. A multicast address is chosen at the source (sender) for the receivers in a multicast group.

Note The Class D address range is used only for the group address or destination address of IP multicast traffic. The source address for multicast datagrams is always the unicast source address.

IP Multicast Address Scoping

The multicast address range is subdivided to provide predictable behavior for various address ranges and for address reuse within smaller domains. The table provides a summary of the multicast address ranges. A brief summary description of each range follows.

Table 1: Multicast Address Range Assignments

Name	Range	Description
Reserved Link-Local Addresses	224.0.0.0 to 224.0.0.255	Reserved for use by network protocols on a local network segment.
Globally Scoped Addresses	224.0.1.0 to 238.255.255.255	Reserved to send multicast data between organizations and across the Internet.
Source Specific Multicast	232.0.0.0 to 232.255.255.255	Reserved for use with the SSM datagram delivery model where data is forwarded only to receivers that have explicitly joined the group.
GLOP Addresses	233.0.0.0 to 233.255.255.255	Reserved for statically defined addresses by organizations that already have an assigned autonomous system (AS) domain number.
Limited Scope Address	239.0.0.0 to 239.255.255.255	Reserved as administratively or limited scope addresses for use in private multicast domains.

Reserved Link-Local Addresses

The IANA has reserved the range 224.0.0.0 to 224.0.0.255 for use by network protocols on a local network segment. Packets with an address in this range are local in scope and are not forwarded by IP routers. Packets with link local destination addresses are typically sent with a time-to-live (TTL) value of 1 and are not forwarded by a router.

Within this range, reserved link-local addresses provide network protocol functions for which they are reserved. Network protocols use these addresses for automatic router discovery and to communicate important routing information. For example, Open Shortest Path First (OSPF) uses the IP addresses 224.0.0.5 and 224.0.0.6 to exchange link-state information.

IANA assigns single multicast address requests for network protocols or network applications out of the 224.0.1.xxx address range. Multicast routers forward these multicast addresses.

Globally Scoped Addresses

Addresses in the range 224.0.1.0 to 238.255.255.255 are called globally scoped addresses. These addresses are used to send multicast data between organizations across the Internet. Some of these addresses have been reserved by IANA for use by multicast applications. For example, the IP address 224.0.1.1 is reserved for Network Time Protocol (NTP).

Source Specific Multicast Addresses

Addresses in the range 232.0.0.0/8 are reserved for Source Specific Multicast (SSM) by IANA. In Cisco IOS software, you can use the **ip pim ssm**command to configure SSM for arbitrary IP multicast addresses also. SSM is an extension of Protocol Independent Multicast (PIM) that allows for an efficient data delivery mechanism in one-to-many communications. SSM is described in the IP Multicast Delivery Modes, on page 4 section.

GLOP Addresses

GLOP addressing (as proposed by RFC 2770, GLOP Addressing in 233/8) proposes that the 233.0.0.0/8 range be reserved for statically defined addresses by organizations that already have an AS number reserved. This practice is called GLOP addressing. The AS number of the domain is embedded into the second and third octets of the 233.0.0.0/8 address range. For example, AS 62010 is written in hexadecimal format as F23A. Separating the two octets F2 and 3A results in 242 and 58 in decimal format. These values result in a subnet of 233.242.58.0/24 that would be globally reserved for AS 62010 to use.

Limited Scope Addresses

The range 239.0.0.0 to 239.255.255.255 is reserved as administratively or limited scoped addresses for use in private multicast domains. These addresses are constrained to a local group or organization. Companies, universities, and other organizations can use limited scope addresses to have local multicast applications that will not be forwarded outside their domain. Routers typically are configured with filters to prevent multicast traffic in this address range from flowing outside an autonomous system (AS) or any user-defined domain. Within an AS or domain, the limited scope address range can be further subdivided so that local multicast boundaries can be defined.

Note Network administrators may use multicast addresses in this range, inside a domain, without conflicting with others elsewhere in the Internet.

Layer 2 Multicast Addresses

Historically, network interface cards (NICs) on a LAN segment could receive only packets destined for their burned-in MAC address or the broadcast MAC address. In IP multicast, several hosts need to be able to receive a single data stream with a common destination MAC address. Some means had to be devised so that multiple hosts could receive the same packet and still be able to differentiate between several multicast groups. One method to accomplish this is to map IP multicast Class D addresses directly to a MAC address. Using this method, NICs can receive packets destined to many different MAC address.

Cisco Group Management Protocol (CGMP) is used on routers connected to Catalyst switches to perform tasks similar to those performed by IGMP. CGMP is necessary for those Catalyst switches that cannot distinguish between IP multicast data packets and IGMP report messages, both of which are addressed to the same group address at the MAC level.

IP Multicast Delivery Modes

IP multicast delivery modes differ only for the receiver hosts, not for the source hosts. A source host sends IP multicast packets with its own IP address as the IP source address of the packet and a group address as the IP destination address of the packet.

Any Source Multicast

For the Any Source Multicast (ASM) delivery mode, an IP multicast receiver host can use any version of IGMP to join a multicast group. This group is notated as G in the routing table state notation. By joining this group, the receiver host is indicating that it wants to receive IP multicast traffic sent by any source to group G. The network will deliver IP multicast packets from any source host with the destination address G to all receiver hosts in the network that have joined group G.

ASM requires group address allocation within the network. At any given time, an ASM group should only be used by a single application. When two applications use the same ASM group simultaneously, receiver hosts of both applications will receive traffic from both application sources. This may result in unexpected excess traffic in the network. This situation may cause congestion of network links and malfunction of the application receiver hosts.

Source Specific Multicast

Source Specific Multicast (SSM) is a datagram delivery model that best supports one-to-many applications, also known as broadcast applications. SSM is a core network technology for the Cisco implementation of IP multicast targeted for audio and video broadcast application environments.

For the SSM delivery mode, an IP multicast receiver host must use IGMP Version 3 (IGMPv3) to subscribe to channel (S,G). By subscribing to this channel, the receiver host is indicating that it wants to receive IP multicast traffic sent by source host S to group G. The network will deliver IP multicast packets from source host S to group G to all hosts in the network that have subscribed to the channel (S, G).

SSM does not require group address allocation within the network, only within each source host. Different applications running on the same source host must use different SSM groups. Different applications running on different source hosts can arbitrarily reuse SSM group addresses without causing any excess traffic on the network.

Protocol Independent Multicast

The Protocol Independent Multicast (PIM) protocol maintains the current IP multicast service mode of receiver-initiated membership. PIM is not dependent on a specific unicast routing protocol; it is IP routing protocol independent and can leverage whichever unicast routing protocols are used to populate the unicast routing table, including Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), and static routes. PIM uses unicast routing information to perform the multicast forwarding function.

Although PIM is called a multicast routing protocol, it actually uses the unicast routing table to perform the reverse path forwarding (RPF) check function instead of building up a completely independent multicast routing table. Unlike other routing protocols, PIM does not send and receive routing updates between routers.

PIM can operate in dense mode or sparse mode. The router can also handle both sparse groups and dense groups at the same time. The mode determines how the router populates its multicast routing table and how the router forwards multicast packets it receives from its directly connected LANs.

For information about PIM forwarding (interface) modes, see the following sections:

PIM Dense Mode

PIM dense mode (PIM-DM) uses a push model to flood multicast traffic to every corner of the network. This push model is a method for delivering data to the receivers without the receivers requesting the data. This method is efficient in certain deployments in which there are active receivers on every subnet in the network.

In dense mode, a router assumes that all other routers want to forward multicast packets for a group. If a router receives a multicast packet and has no directly connected members or PIM neighbors present, a prune message is sent back to the source. Subsequent multicast packets are not flooded to this router on this pruned branch. PIM builds source-based multicast distribution trees.

PIM-DM initially floods multicast traffic throughout the network. Routers that have no downstream neighbors prune back the unwanted traffic. This process repeats every 3 minutes.

Routers accumulate state information by receiving data streams through the flood and prune mechanism. These data streams contain the source and group information so that downstream routers can build up their multicast forwarding table. PIM-DM supports only source trees--that is, (S,G) entries--and cannot be used to build a shared distribution tree.

Note Dense mode is not often used and its use is not recommended. For this reason it is not specified in the configuration tasks in related modules.

PIM Sparse Mode

PIM sparse mode (PIM-SM) uses a pull model to deliver multicast traffic. Only network segments with active receivers that have explicitly requested the data will receive the traffic.

Unlike dense mode interfaces, sparse mode interfaces are added to the multicast routing table only when periodic Join messages are received from downstream routers, or when a directly connected member is on the interface. When forwarding from a LAN, sparse mode operation occurs if an RP is known for the group. If so, the packets are encapsulated and sent toward the RP. When no RP is known, the packet is flooded in a dense mode fashion. If the multicast traffic from a specific source is sufficient, the first hop router of the receiver may send Join messages toward the source to build a source-based distribution tree.

PIM-SM distributes information about active sources by forwarding data packets on the shared tree. Because PIM-SM uses shared trees (at least, initially), it requires the use of a rendezvous point (RP). The RP must be administratively configured in the network. See the Rendezvous Points, on page 9 section for more information.

In sparse mode, a router assumes that other routers do not want to forward multicast packets for a group, unless there is an explicit request for the traffic. When hosts join a multicast group, the directly connected routers send PIM Join messages toward the RP. The RP keeps track of multicast groups. Hosts that send multicast packets are registered with the RP by the first hop router of that host. The RP then sends Join messages toward the source. At this point, packets are forwarded on a shared distribution tree. If the multicast traffic from a specific source is sufficient, the first hop router of the host may send Join messages toward the source to build a source-based distribution tree.

Sources register with the RP and then data is forwarded down the shared tree to the receivers. The edge routers learn about a particular source when they receive data packets on the shared tree from that source through the RP. The edge router then sends PIM (S,G) Join messages toward that source. Each router along the reverse path compares the unicast routing metric of the RP address to the metric of the source address. If the metric for the source address is better, it will forward a PIM (S,G) Join message toward the source. If the metric for the RP is the same or better, then the PIM (S,G) Join message will be sent in the same direction as the RP. In this case, the shared tree and the source tree would be considered congruent.

If the shared tree is not an optimal path between the source and the receiver, the routers dynamically create a source tree and stop traffic from flowing down the shared tree. This behavior is the default behavior in software. Network administrators can force traffic to stay on the shared tree by using the **ip pim spt-threshold infinity** command.

PIM-SM scales well to a network of any size, including those with WAN links. The explicit join mechanism prevents unwanted traffic from flooding the WAN links.

Sparse-Dense Mode

If you configure either sparse mode or dense mode on an interface, then sparseness or denseness is applied to the interface as a whole. However, some environments might require PIM to run in a single region in sparse mode for some groups and in dense mode for other groups.

An alternative to enabling only dense mode or only sparse mode is to enable sparse-dense mode. In this case, the interface is treated as dense mode if the group is in dense mode; the interface is treated in sparse mode if the group is in sparse mode. You must have an RP if the interface is in sparse-dense mode and you want to treat the group as a sparse group.

If you configure sparse-dense mode, the idea of sparseness or denseness is applied to the groups for which the router is a member.

Another benefit of sparse-dense mode is that Auto-RP information can be distributed in a dense mode; yet, multicast groups for user groups can be used in a sparse mode manner. Therefore there is no need to configure a default RP at the leaf routers.

When an interface is treated in dense mode, it is populated in the outgoing interface list of a multicast routing table when either of the following conditions is true:

- Members or DVMRP neighbors are on the interface.
- There are PIM neighbors and the group has not been pruned.

When an interface is treated in sparse mode, it is populated in the outgoing interface list of a multicast routing table when either of the following conditions is true:

- Members or DVMRP neighbors are on the interface.
- An explicit Join message has been received by a PIM neighbor on the interface.

Bidirectional PIM

Bidirectional PIM (bidir-PIM) is an enhancement of the PIM protocol that was designed for efficient many-to-many communications within an individual PIM domain. Multicast groups in bidirectional mode can scale to an arbitrary number of sources with only a minimal amount of additional overhead.

The shared trees that are created in PIM sparse mode are unidirectional. This means that a source tree must be created to bring the data stream to the RP (the root of the shared tree) and then it can be forwarded down the branches to the receivers. Source data cannot flow up the shared tree toward the RP--this would be considered a bidirectional shared tree.

In bidirectional mode, traffic is routed only along a bidirectional shared tree that is rooted at the RP for the group. In bidir-PIM, the IP address of the RP acts as the key to having all routers establish a loop-free spanning tree topology rooted in that IP address. This IP address need not be a router address, but can be any unassigned IP address on a network that is reachable throughout the PIM domain.

Bidir-PIM is derived from the mechanisms of PIM sparse mode (PIM-SM) and shares many of the shared tree operations. Bidir-PIM also has unconditional forwarding of source traffic toward the RP upstream on the shared tree, but no registering process for sources as in PIM-SM. These modifications are necessary and sufficient to allow forwarding of traffic in all routers solely based on the (*, G) multicast routing entries. This feature eliminates any source-specific state and allows scaling capability to an arbitrary number of sources.

Multicast Group Modes

In PIM, packet traffic for a multicast group is routed according to the rules of the mode configured for that multicast group. The Cisco implementation of PIM supports four modes for a multicast group:

- PIM Bidirectional mode
- PIM Sparse mode
- PIM Dense mode
- PIM Source Specific Multicast (SSM) mode

A router can simultaneously support all four modes or any combination of them for different multicast groups.

Bidirectional Mode

In bidirectional mode, traffic is routed only along a bidirectional shared tree that is rooted at the rendezvous point (RP) for the group. In bidir-PIM, the IP address of the RP acts as the key to having all routers establish a loop-free spanning tree topology rooted in that IP address. This IP address need not be a router, but can be any unassigned IP address on a network that is reachable throughout the PIM domain. This technique is the preferred configuration method for establishing a redundant RP configuration for bidir-PIM.

Membership to a bidirectional group is signalled via explicit Join messages. Traffic from sources is unconditionally sent up the shared tree toward the RP and passed down the tree toward the receivers on each branch of the tree.

Sparse Mode

Sparse mode operation centers around a single unidirectional shared tree whose root node is called the rendezvous point (RP). Sources must register with the RP to get their multicast traffic to flow down the shared tree by way of the RP. This registration process actually triggers a shortest path tree (SPT) Join by the RP toward the source when there are active receivers for the group in the network.

A sparse mode group uses the explicit join model of interaction. Receiver hosts join a group at a rendezvous point (RP). Different groups can have different RPs.

Multicast traffic packets flow down the shared tree to only those receivers that have explicitly asked to receive the traffic.

Dense Mode

Dense mode operates using the broadcast (flood) and prune model.

In populating the multicast routing table, dense mode interfaces are always added to the table. Multicast traffic is forwarded out all interfaces in the outgoing interface list to all receivers. Interfaces are removed from the outgoing interface list in a process called pruning. In dense mode, interfaces are pruned for various reasons including that there are no directly connected receivers.

A pruned interface can be reestablished, that is, grafted back so that restarting the flow of multicast traffic can be accomplished with minimal delay.

Rendezvous Points

A rendezvous point (RP) is a role that a device performs when operating in Protocol Independent Multicast (PIM) Sparse Mode (SM). An RP is required only in networks running PIM SM. In the PIM-SM model, only network segments with active receivers that have explicitly requested multicast data will be forwarded the traffic.

This method of delivering multicast data is in contrast to PIM Dense Mode (PIM DM). In PIM DM, multicast traffic is initially flooded to all segments of the network. Routers that have no downstream neighbors or directly connected receivers prune back the unwanted traffic.

An RP acts as the meeting place for sources and receivers of multicast data. In a PIM-SM network, sources must send their traffic to the RP. This traffic is then forwarded to receivers down a shared distribution tree. By default, when the first hop device of the receiver learns about the source, it will send a Join message directly to the source, creating a source-based distribution tree from the source to the receiver. This source tree does not include the RP unless the RP is located within the shortest path between the source and receiver.

In most cases, the placement of the RP in the network is not a complex decision. By default, the RP is needed only to start new sessions with sources and receivers. Consequently, the RP experiences little overhead from traffic flow or processing. In PIM version 2, the RP performs less processing than in PIM version 1 because sources must only periodically register with the RP to create state.

Auto-RP

In the first version of PIM-SM, all leaf routers (routers directly connected to sources or receivers) were required to be manually configured with the IP address of the RP. This type of configuration is also known as static RP configuration. Configuring static RPs is relatively easy in a small network, but it can be laborious in a large, complex network.

Following the introduction of PIM-SM version 1, Cisco implemented a version of PIM-SM with the Auto-RP feature. Auto-RP automates the distribution of group-to-RP mappings in a PIM network. Auto-RP has the following benefits:

- Configuring the use of multiple RPs within a network to serve different groups is easy.
- Auto-RP allows load splitting among different RPs and arrangement of RPs according to the location of group participants.
- Auto-RP avoids inconsistent, manual RP configurations that can cause connectivity problems.

Multiple RPs can be used to serve different group ranges or serve as backups to each other. For Auto-RP to work, a router must be designated as an RP-mapping agent, which receives the RP-announcement messages from the RPs and arbitrates conflicts. The RP-mapping agent then sends the consistent group-to-RP mappings to all other routers. Thus, all routers automatically discover which RP to use for the groups they support.

Note If you configure PIM in sparse mode or sparse-dense mode and do not configure Auto-RP, you must statically configure an RP.

Note If router interfaces are configured in sparse mode, Auto-RP can still be used if all routers are configured with a static RP address for the Auto-RP groups.

To make Auto-RP work, a router must be designated as an RP mapping agent, which receives the RP announcement messages from the RPs and arbitrates conflicts. The RP mapping agent then sends the consistent group-to-RP mappings to all other routers by dense mode flooding. Thus, all routers automatically discover which RP to use for the groups they support. The Internet Assigned Numbers Authority (IANA) has assigned two group addresses, 224.0.1.39 and 224.0.1.40, for Auto-RP. One advantage of Auto-RP is that any change to the RP designation must be configured only on the routers that are RPs and not on the leaf routers. Another advantage of Auto-RP is that it offers the ability to scope the RP address within a domain. Scoping can be achieved by defining the time-to-live (TTL) value allowed for the Auto-RP advertisements.

Each method for configuring an RP has its own strengths, weaknesses, and level of complexity. In conventional IP multicast network scenarios, we recommend using Auto-RP to configure RPs because it is easy to configure, well-tested, and stable. The alternative ways to configure an RP are static RP, Auto-RP, and bootstrap router.

Sparse-Dense Mode for Auto-RP

A prerequisite of Auto-RP is that all interfaces must be configured in sparse-dense mode using the **ip pim sparse-dense-mode** interface configuration command. An interface configured in sparse-dense mode is treated in either sparse mode or dense mode of operation, depending on which mode the multicast group operates. If a multicast group has a known RP, the interface is treated in sparse mode. If a group has no known RP, by default the interface is treated in dense mode and data will be flooded over this interface. (You can prevent dense-mode fallback; see the module “Configuring Basic IP Multicast.”)

To successfully implement Auto-RP and prevent any groups other than 224.0.1.39 and 224.0.1.40 from operating in dense mode, we recommend configuring a “sink RP” (also known as “RP of last resort”). A sink RP is a statically configured RP that may or may not actually exist in the network. Configuring a sink RP does not interfere with Auto-RP operation because, by default, Auto-RP messages supersede static RP configurations. We recommend configuring a sink RP for all possible multicast groups in your network, because it is possible for an unknown or unexpected source to become active. If no RP is configured to limit source registration, the group may revert to dense mode operation and be flooded with data.

Bootstrap Router

Another RP selection model called bootstrap router (BSR) was introduced after Auto-RP in PIM-SM version 2. BSR performs similarly to Auto-RP in that it uses candidate routers for the RP function and for relaying the RP information for a group. RP information is distributed through BSR messages, which are carried within PIM messages. PIM messages are link-local multicast messages that travel from PIM router to PIM router. Because of this single hop method of disseminating RP information, TTL scoping cannot be used with BSR. A BSR performs similarly as an RP, except that it does not run the risk of reverting to dense mode operation, and it does not offer the ability to scope within a domain.

Multicast Source Discovery Protocol

In the PIM sparse mode model, multicast sources and receivers must register with their local rendezvous point (RP). Actually, the router closest to a source or a receiver registers with the RP, but the key point to note is that the RP “knows” about all the sources and receivers for any particular group. RPs in other domains have no way of knowing about sources that are located in other domains. Multicast Source Discovery Protocol (MSDP) is an elegant way to solve this problem.

MSDP is a mechanism that allows RPs to share information about active sources. RPs know about the receivers in their local domain. When RPs in remote domains hear about the active sources, they can pass on that information to their local receivers. Multicast data can then be forwarded between the domains. A useful feature of MSDP is that it allows each domain to maintain an independent RP that does not rely on other domains, but it does enable RPs to forward traffic between domains. PIM-SM is used to forward the traffic between the multicast domains.

The RP in each domain establishes an MSDP peering session using a TCP connection with the RPs in other domains or with border routers leading to the other domains. When the RP learns about a new multicast source within its own domain (through the normal PIM register mechanism), the RP encapsulates the first data packet in a Source-Active (SA) message and sends the SA to all MSDP peers. Each receiving peer uses a modified Reverse Path Forwarding (RPF) check to forward the SA, until the SA reaches every MSDP router in the interconnected networks--theoretically the entire multicast internet. If the receiving MSDP peer is an RP, and the RP has a (*, G) entry for the group in the SA (there is an interested receiver), the RP creates (S,G) state for the source and joins to the shortest path tree for the source. The encapsulated data is decapsulated and forwarded down the shared tree of that RP. When the last hop router (the router closest to the receiver) receives the multicast packet, it may join the shortest path tree to the source. The MSDP speaker periodically sends SAs that include all sources within the domain of the RP.

MSDP was developed for peering between Internet service providers (ISPs). ISPs did not want to rely on an RP maintained by a competing ISP to provide service to their customers. MSDP allows each ISP to have its own local RP and still forward and receive multicast traffic to the Internet.

Anycast RP

Anycast RP is a useful application of MSDP. Originally developed for interdomain multicast applications, MSDP used for Anycast RP is an intradomain feature that provides redundancy and load-sharing capabilities. Enterprise customers typically use Anycast RP for configuring a Protocol Independent Multicast sparse mode (PIM-SM) network to meet fault tolerance requirements within a single multicast domain.

In Anycast RP, two or more RPs are configured with the same IP address on loopback interfaces. The Anycast RP loopback address should be configured with a 32-bit mask, making it a host address. All the downstream routers should be configured to "know" that the Anycast RP loopback address is the IP address of their local RP. IP routing automatically will select the topologically closest RP for each source and receiver. Assuming that the sources are evenly spaced around the network, an equal number of sources will register with each RP. That is, the process of registering the sources will be shared equally by all the RPs in the network.

Because a source may register with one RP and receivers may join to a different RP, a method is needed for the RPs to exchange information about active sources. This information exchange is done with MSDP.

In Anycast RP, all the RPs are configured to be MSDP peers of each other. When a source registers with one RP, an SA message will be sent to the other RPs informing them that there is an active source for a particular multicast group. The result is that each RP will know about the active sources in the area of the other RPs. If any of the RPs were to fail, IP routing would converge and one of the RPs would become the active RP in more than one area. New sources would register with the backup RP. Receivers would join toward the new RP and connectivity would be maintained.

Note The RP is normally needed only to start new sessions with sources and receivers. The RP facilitates the shared tree so that sources and receivers can directly establish a multicast data flow. If a multicast data flow is already directly established between a source and the receiver, then an RP failure will not affect that session. Anycast RP ensures that new sessions with sources and receivers can begin at any time.

Multicast Forwarding

Forwarding of multicast traffic is accomplished by multicast-capable routers. These routers create distribution trees that control the path that IP multicast traffic takes through the network in order to deliver traffic to all receivers.

Multicast traffic flows from the source to the multicast group over a distribution tree that connects all of the sources to all of the receivers in the group. This tree may be shared by all sources (a shared tree) or a separate distribution tree can be built for each source (a source tree). The shared tree may be one-way or bidirectional.

Before describing the structure of source and shared trees, it is helpful to explain the notations that are used in multicast routing tables. These notations include the following:

- (S,G) = (unicast source for the multicast group G, multicast group G)
- (*,G) = (any source for the multicast group G, multicast group G)

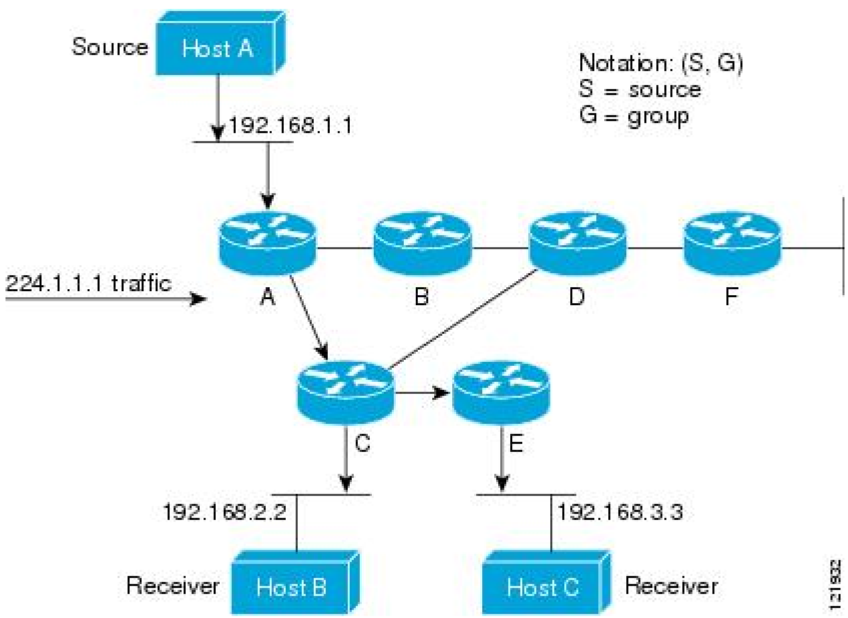
The notation of (S,G), pronounced “S comma G,” enumerates a shortest path tree where S is the IP address of the source and G is the multicast group address.

Shared trees are (*,G) and the source trees are (S,G) and always routed at the sources.

Multicast Distribution Source Tree

The simplest form of a multicast distribution tree is a source tree. A source tree has its root at the source host and has branches forming a spanning tree through the network to the receivers. Because this tree uses the shortest path through the network, it is also referred to as a shortest path tree (SPT).

The figure shows an example of an SPT for group 224.1.1.1 rooted at the source, Host A, and connecting two receivers, Hosts B and C.



Using standard notation, the SPT for the example shown in the figure would be (192.168.1.1, 224.1.1.1).

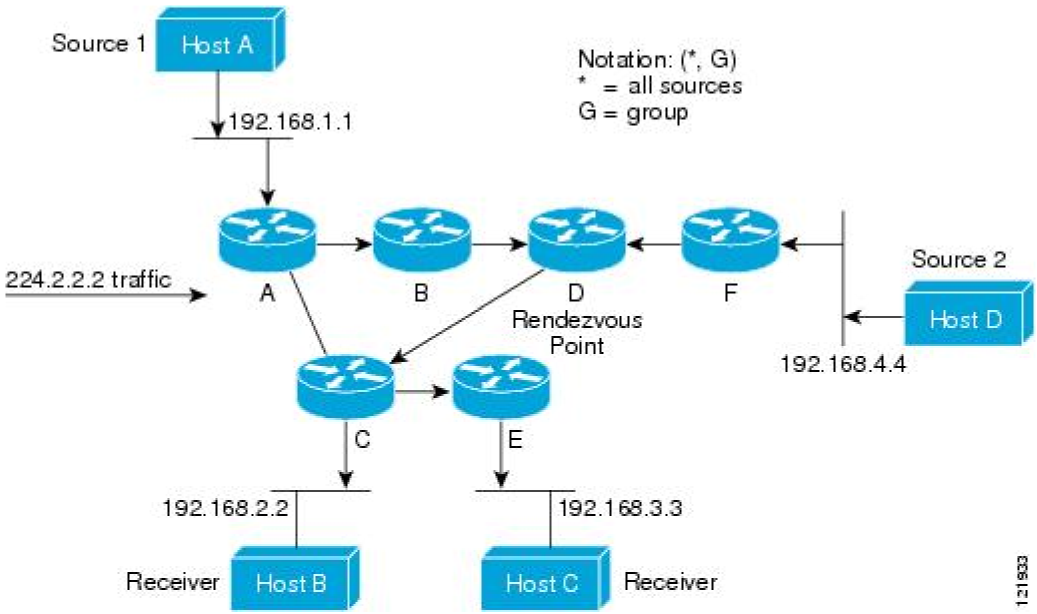
The (S,G) notation implies that a separate SPT exists for each individual source sending to each group--which is correct.

Multicast Distribution Shared Tree

Unlike source trees that have their root at the source, shared trees use a single common root placed at some chosen point in the network. This shared root is called a rendezvous point (RP).

The following figure shows a shared tree for the group 224.2.2.2 with the root located at Router D. This shared tree is unidirectional. Source traffic is sent towards the RP on a source tree. The traffic is then forwarded down the shared tree from the RP to reach all of the receivers (unless the receiver is located between the source and the RP, in which case it will be serviced directly).

Figure 1: Shared Tree



In this example, multicast traffic from the sources, Hosts A and D, travels to the root (Router D) and then down the shared tree to the two receivers, Hosts B and C. Because all sources in the multicast group use a common shared tree, a wildcard notation written as (*, G), pronounced "star comma G", represents the tree. In this case, * means all sources, and G represents the multicast group. Therefore, the shared tree shown in the figure would be written as (*, 224.2.2.2).

Both source trees and shared trees are loop-free. Messages are replicated only where the tree branches. Members of multicast groups can join or leave at any time; therefore the distribution trees must be dynamically updated. When all the active receivers on a particular branch stop requesting the traffic for a particular multicast group, the routers prune that branch from the distribution tree and stop forwarding traffic down that branch. If one receiver on that branch becomes active and requests the multicast traffic, the router will dynamically modify the distribution tree and start forwarding traffic again.

Source Tree Advantage

Source trees have the advantage of creating the optimal path between the source and the receivers. This advantage guarantees the minimum amount of network latency for forwarding multicast traffic. However, this optimization comes at a cost. The routers must maintain path information for each source. In a network that has thousands of sources and thousands of groups, this overhead can quickly become a resource issue on the routers. Memory consumption from the size of the multicast routing table is a factor that network designers must take into consideration.

Shared Tree Advantage

Shared trees have the advantage of requiring the minimum amount of state in each router. This advantage lowers the overall memory requirements for a network that only allows shared trees. The disadvantage of shared trees is that under certain circumstances the paths between the source and receivers might not be the optimal paths, which might introduce some latency in packet delivery. For example, in the figure above the shortest path between Host A (source 1) and Host B (a receiver) would be Router A and Router C. Because we are using Router D as the root for a shared tree, the traffic must traverse Routers A, B, D and then C. Network designers must carefully consider the placement of the rendezvous point (RP) when implementing a shared tree-only environment.

In unicast routing, traffic is routed through the network along a single path from the source to the destination host. A unicast router does not consider the source address; it considers only the destination address and how to forward the traffic toward that destination. The router scans through its routing table for the destination address and then forwards a single copy of the unicast packet out the correct interface in the direction of the destination.

In multicast forwarding, the source is sending traffic to an arbitrary group of hosts that are represented by a multicast group address. The multicast router must determine which direction is the upstream direction (toward the source) and which one is the downstream direction (or directions) toward the receivers. If there are multiple downstream paths, the router replicates the packet and forwards it down the appropriate downstream paths (best unicast route metric)--which is not necessarily all paths. Forwarding multicast traffic away from the source, rather than to the receiver, is called Reverse Path Forwarding (RPF). RPF is described in the following section.

Reverse Path Forwarding

In unicast routing, traffic is routed through the network along a single path from the source to the destination host. A unicast router does not consider the source address; it considers only the destination address and how to forward the traffic toward that destination. The router scans through its routing table for the destination network and then forwards a single copy of the unicast packet out the correct interface in the direction of the destination.

In multicast forwarding, the source is sending traffic to an arbitrary group of hosts that are represented by a multicast group address. The multicast router must determine which direction is the upstream direction (toward the source) and which one is the downstream direction (or directions) toward the receivers. If there are multiple downstream paths, the router replicates the packet and forwards it down the appropriate downstream paths (best unicast route metric)--which is not necessarily all paths. Forwarding multicast traffic away from the source, rather than to the receiver, is called Reverse Path Forwarding (RPF). RPF is an algorithm used for forwarding multicast datagrams.

Protocol Independent Multicast (PIM) uses the unicast routing information to create a distribution tree along the reverse path from the receivers towards the source. The multicast routers then forward packets along the distribution tree from the source to the receivers. RPF is a key concept in multicast forwarding. It enables routers to correctly forward multicast traffic down the distribution tree. RPF makes use of the existing unicast routing table to determine the upstream and downstream neighbors. A router will forward a multicast packet only if it is received on the upstream interface. This RPF check helps to guarantee that the distribution tree will be loop-free.

RPF Check

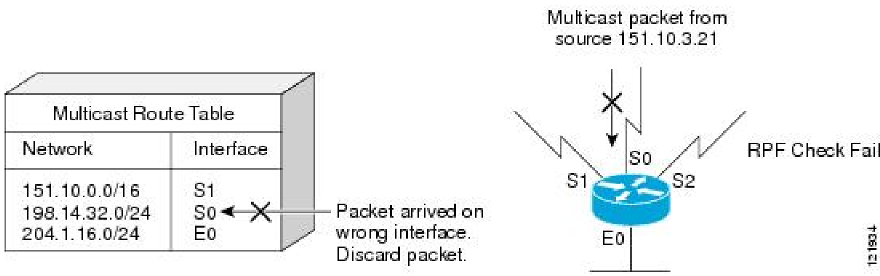
When a multicast packet arrives at a router, the router performs an RPF check on the packet. If the RPF check succeeds, the packet is forwarded. Otherwise, it is dropped.

For traffic flowing down a source tree, the RPF check procedure works as follows:

1. The router looks up the source address in the unicast routing table to determine if the packet has arrived on the interface that is on the reverse path back to the source.
2. If the packet has arrived on the interface leading back to the source, the RPF check succeeds and the packet is forwarded out the interfaces present in the outgoing interface list of a multicast routing table entry.
3. If the RPF check in Step 2 fails, the packet is dropped.

The figure shows an example of an unsuccessful RPF check.

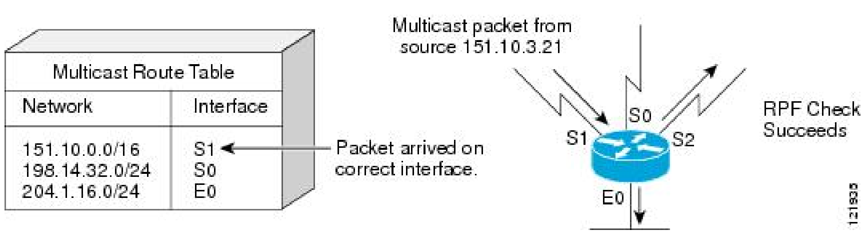
Figure 2: RPF Check Fails



As the figure illustrates, a multicast packet from source 151.10.3.21 is received on serial interface 0 (S0). A check of the unicast route table shows that S1 is the interface this router would use to forward unicast data to 151.10.3.21. Because the packet has arrived on interface S0, the packet is discarded.

The figure shows an example of a successful RPF check.

Figure 3: RPF Check Succeeds



In this example, the multicast packet has arrived on interface S1. The router refers to the unicast routing table and finds that S1 is the correct interface. The RPF check passes, and the packet is forwarded.

PIM Dense Mode Fallback

If you use IP multicast in mission-critical networks, you should avoid the use of PIM-DM (dense mode).

Dense mode fallback describes the event of the PIM mode changing (falling back) from sparse mode (which requires an RP) to dense mode (which does not use an RP). Dense mode fallback occurs when RP information is lost.

If all interfaces are configured with the **ip pim sparse-mode** command, there is no dense mode fallback because dense mode groups cannot be created over interfaces configured for sparse mode.

Note To configure multicast over DMVPN (dynamic multipoint virtual private network), ensure that you configure PIM sparse mode using **ip pim sparse-mode**command. Configuring PIM dense mode over a DMVPN tunnel interface is not supported on IOS-XE.

Cause and Effect of Dense Mode Fallback

PIM determines whether a multicast group operates in PIM-DM or PIM-SM mode based solely on the existence of RP information in the group-to-RP mapping cache. If Auto-RP is configured or a bootstrap router (BSR) is used to distribute RP information, there is a risk that RP information can be lost if all RPs, Auto-RP, or the BSR for a group fails due to network congestion. This failure can lead to the network either partially or fully falling back into PIM-DM.

If a network falls back into PIM-DM and AutoRP or BSR is being used, dense mode flooding will occur. Routers that lose RP information will fallback into dense mode and any new states that must be created for the failed group will be created in dense mode.

Effects of Preventing Dense Mode Fallback

Prior to the introduction of PIM-DM fallback prevention, all multicast groups without a group-to-RP mapping would be treated as dense mode.

With the introduction of PIM-DM fallback prevention, the PIM-DM fallback behavior has been changed to prevent dense mode flooding. By default, if all of the interfaces are configured to operate in PIM sparse mode (using the **ip pim sparse-mode** command), there is no need to configure the **no ip pim dm-fallback** command (that is, the PIM-DM fallback behavior is enabled by default). If any interfaces are not configured using the **ip pim sparse-mode**command (for example, using the **ip pim sparse-dense-mode** command), then the PIM-DM fallback behavior can be explicit disabled using the **no ip pim dm-fallback**command.

When the **no ip pim dm-fallback** command is configured or when **ip pim sparse-mode** is configured on all interfaces, any existing groups running in sparse mode will continue to operate in sparse mode but will use an RP address set to 0.0.0.0. Multicast entries with an RP address set to 0.0.0.0 will exhibit the following behavior:

- Existing (S, G) states will be maintained.
- No PIM Join or Prune messages for (*, G) or (S, G, RPbit) are sent.
- Received (*, G) or (S, G, RPbit) Joins or Prune messages are ignored.
- No registers are sent and traffic at the first hop is dropped.
- Received registers are answered with register stop.
- Asserts are unchanged.
- The (*, G) outgoing interface list (olist) is maintained only for the Internet Group Management Protocol (IGMP) state.
- Multicast Source Discovery Protocol (MSDP) source active (SA) messages for RP 0.0.0.0 groups are still accepted and forwarded.

Guidelines for Choosing a PIM Mode

Before beginning the configuration process, you must decide which PIM mode needs to be used. This determination is based on the applications you intend to support on your network.

Basic guidelines include the following:

- In general, if the application is one-to-many or many-to-many in nature, then PIM-SM can be used successfully.
- For optimal one-to-many application performance, SSM is appropriate but requires IGMP version 3 support.
- For optimal many-to-many application performance, bidirectional PIM is appropriate but hardware support is limited to Cisco devices and the Catalyst 6000 series switches with Sup720.

Where to Go Next

- To configure basic IP multicast, see the “ Configuring Basic IP Multicast ” module.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	*Cisco IOS IP Multicast Command Reference*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
CISCO-PIM-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2934	*Protocol Independent Multicast MIB for IPv4*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IP Multicast Technology Overview

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Glossary

basic multicast--Interactive intra-domain multicast. Supports multicast applications within an enterprise campus. Also provides an additional integrity in the network with the inclusion of a reliable multicast transport, PGM.

bidir PIM--Bidirectional PIM is an extension to the PIM suite of protocols that implements shared sparse trees with bidirectional flow of data. In contrast to PIM-SM, bidir-PIM avoids keeping source specific state in router and thus allows trees to scale to an arbitrary number of sources.

broadcast--One-to-all transmission where the source sends one copy of the message to all nodes, whether they wish to receive it or not.

Cisco Group Management Protocol (CGMP)--Cisco-developed protocol that allows Layer 2 switches to leverage IGMP information on Cisco routers to make Layer 2 forwarding decisions. It allows the switches to forward multicast traffic to only those ports that are interested in the traffic.

dense mode (DM) (Internet Draft Spec)--Actively attempts to send multicast data to all potential receivers (flooding) and relies upon their self-pruning (removal from group) to achieve desired distribution.

designated router (DR)--The router in a PIM-SM tree that instigates the Join/Prune message cascade upstream to the RP in response to IGMP membership information it receives from IGMP hosts.

distribution tree--Multicast traffic flows from the source to the multicast group over a distribution tree that connects all of the sources to all of the receivers in the group. This tree may be shared by all sources (a shared-tree), or a separate distribution tree can be built for each source (a source-tree). The shared-tree may be one-way or bidirectional.

IGMP messages--IGMP messages are encapsulated in standard IP datagrams with an IP protocol number of 2 and the IP Router Alert option (RFC 2113).

IGMP snooping--IGMP snooping requires the LAN switch to examine, or "snoop," some Layer 3 information in the IGMP packet sent from the host to the router. When the switch hears an IGMP report from a host for a particular multicast group, the switch adds the host's port number to the associated multicast table entry. When it hears an IGMP Leave Group message from a host, it removes the host's port from the table entry.

IGMP unidirectional link routing--Cisco's other UDLR solution is to use IP multicast routing with IGMP, which has been enhanced to accommodate UDLR. This solution scales very well for many satellite links.

Internet Group Management Protocol v2 (IGMP)--Used by IP routers and their immediately connected hosts to communicate multicast group membership states.

Internet Group Management Protocol v3 (IGMP)--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.

multicast--A routing technique that allows IP traffic to be sent from one source or multiple sources and delivered to multiple destinations. Instead of sending individual packets to each destination, a single packet is sent to a group of destinations known as a multicast group, which is identified by a single IP destination group address. Multicast addressing supports the transmission of a single IP datagram to multiple hosts.

multicast routing monitor (MRM)--A management diagnostic tool that provides network fault detection and isolation in a large multicast routing infrastructure. It is designed to notify a network administrator of multicast routing problems in near real time.

Multicast Source Discovery Protocol (MSDP)--A mechanism to connect multiple PIM sparse mode (PIM-SM) domains. MSDP allows multicast sources for a group to be known to all rendezvous point(s) (RPs) in different domains. Each PIM-SM domain uses its own RPs and need not depend on RPs in other domains. An RP runs MSDP over TCP to discover multicast sources in other domains. MSDP is also used to announce sources sending to a group. These announcements must originate at the domain's RP. MSDP depends heavily on MBGP for interdomain operation.

Protocol Independent Multicast (PIM)--A multicast routing architecture defined by the IETF that enables IP multicast routing on existing IP networks. Its key point is its independence from any underlying unicast protocol such as OSPF or BGP.

prune--Multicast routing terminology indicating that the multicast-enabled router has sent the appropriate multicast messages to remove itself from the multicast tree for a particular multicast group. It will stop receiving the multicast data addressed to that group and, therefore, cannot deliver the data to any connected hosts until it rejoins the group.

query--IGMP messages originating from the router(s) to elicit multicast group membership information from its connected hosts.

rendezvous point (RP)--The multicast router that is the root of the PIM-SM shared multicast distribution tree.

report--IGMP messages originating from the hosts that are joining, maintaining, or leaving their membership in a multicast group.

source tree--A multicast distribution path that directly connects the source's and receivers' designated router (or the rendezvous point) to obtain the shortest path through the network. Results in most efficient routing of data between source and receivers, but may result in unnecessary data duplication throughout the network if built by anything other than the RP.

sparse mode (SM) (RFC 2362)--Relies upon an explicitly joining method before attempting to send multicast data to receivers of a multicast group.

UDLR tunnel--Uses a back channel (another link) so the routing protocols believe the one-way link is bidirectional. The back channel itself is a special, unidirectional, generic route encapsulation (GRE) tunnel through which control traffic flows in the opposite direction of the user data flow. This feature allows IP and its associated unicast and multicast routing protocols to believe the unidirectional link is logically bidirectional. This solution accommodates all IP unicast and multicast routing protocols without changing them. However, it does not scale and no more than 20 tunnels should feed into the upstream router. The purpose of the unidirectional GRE tunnel is to move control packets from a downstream node to an upstream node.

Unicast--Point-to-point transmission requiring the source to send an individual copy of a message to each requester.

unidirectional Link Routing Protocol (UDLR)--A routing protocol that provides a way to forward multicast packets over a physical unidirectional interface (such as a satellite link of high bandwidth) to stub networks that have a back channel.

URL rendezvous directory (URD)--URD is a multicast-lite solution that directly provides the network with information about the specific source of a content stream. It enables the network to quickly establish the most direct distribution path from the source to the receiver, thus significantly reducing the time and effort required in receiving the streaming media. URD allows an application to identify the source of the content stream through a web page link or web directly. When that information is sent back to the application it is then conveyed back to the network using URD.

In this feature, a URD-capable web page provides information about the source, the group, and the application (via media-type) on a web page. An interested host will click on the web page pulling across the information in an HTTP transaction. The last-hop router to receiver would intercept this transaction and send it to a special port allocated by IANA. The last-hop router is also URD capable and uses the information to initiate the PIM source, group (S,G) join on behalf of the host.

CHAPTER 2

Configuring Basic IP Multicast

IP multicast is a bandwidth-conserving technology that reduces traffic by delivering a single stream of information simultaneously to potentially thousands of corporate businesses and homes. Applications that take advantage of multicast include video conferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news. This module describes the tasks used to configure basic IP multicast.

Prerequisites for Configuring Basic IP Multicast

- To determine which of the tasks contained in this module you will have to perform, you must decide which Protocol Independent Multicast (PIM) mode will be used. This determination is based on the applications you intend to support on your network.
- All access lists to be used with the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the “Creating an IP Access List and Applying It to an Interface” module of the *Security Configuration Guide: Access Control Lists* guide.

Information About Configuring Basic IP Multicast

Auto-RP Overview

The Role of Auto-RP in a PIM Network

Auto-RP automates the distribution of group-to-rendezvous point (RP) mappings in a PIM network. To make Auto-RP work, a device must be designated as an RP mapping agent, which receives the RP announcement messages from the RPs and arbitrates conflicts. The RP mapping agent then sends the consistent group-to-RP mappings to all other devices by way of dense mode flooding.

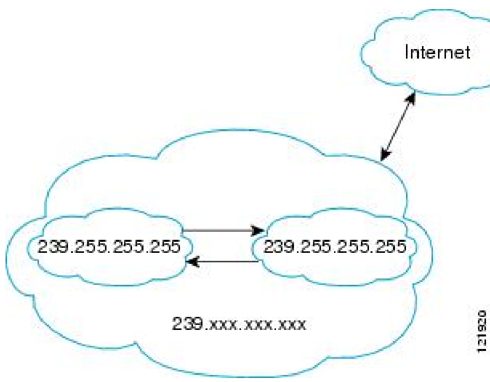
Thus, all routers automatically discover which RP to use for the groups they support. The Internet Assigned Numbers Authority (IANA) has assigned two group addresses, 224.0.1.39 and 224.0.1.40, for Auto-RP.

The mapping agent receives announcements of intention to become the RP from Candidate-RPs. The mapping agent then announces the winner of the RP election. This announcement is made independently of the decisions by the other mapping agents.

IP Multicast Boundary

As shown in the figure, address scoping defines domain boundaries so that domains with RPs that have the same IP address do not leak into each other. Scoping is performed on the subnet boundaries within large domains and on the boundaries between the domain and the Internet.

Figure 4: Address Scoping at Boundaries



You can set up an administratively scoped boundary on an interface for multicast group addresses using the **ip multicast boundary** command with the *access-list* argument. A standard access list defines the range of addresses affected. When a boundary is set up, no multicast data packets are allowed to flow across the boundary from either direction. The boundary allows the same multicast group address to be reused in different administrative domains.

The Internet Assigned Numbers Authority (IANA) has designated the multicast address range 239.0.0.0 to 239.255.255.255 as the administratively scoped addresses. This range of addresses can be reused in domains administered by different organizations. They would be considered local, not globally unique.

You can configure the **filter-autorp** keyword to examine and filter Auto-RP discovery and announcement messages at the administratively scoped boundary. Any Auto-RP group range announcements from the Auto-RP packets that are denied by the boundary access control list (ACL) are removed. An Auto-RP group range announcement is permitted and passed by the boundary only if all addresses in the Auto-RP group range are permitted by the boundary ACL. If any address is not permitted, the entire group range is filtered and removed from the Auto-RP message before the Auto-RP message is forwarded.

Benefits of Auto-RP in a PIM Network

- Auto-RP allows any change to the RP designation to be configured only on the devices that are RPs, not on the leaf routers.
- Auto-RP offers the ability to scope the RP address within a domain.

Anycast RP Overview

Anycast RP is a useful application of MSDP. Originally developed for interdomain multicast applications, MSDP used for Anycast RP is an intradomain feature that provides redundancy and load-sharing capabilities. Enterprise customers typically use Anycast RP for configuring a Protocol Independent Multicast sparse mode (PIM-SM) network to meet fault tolerance requirements within a single multicast domain.

In anycast RP, two or more RPs are configured with the same IP address on loopback interfaces. The anycast RP loopback address should be configured with a 32-bit mask, making it a host address. All the downstream routers should be configured so that the anycast RP loopback address is the IP address of their local RP. IP routing will automatically select the topologically closest RP for each source and receiver. Assuming that the sources are evenly spaced around the network, an equal number of sources will register with each RP. That is, the process of registering the sources will be shared equally by all the RPs in the network.

Because a source may register with one RP and receivers may join to a different RP, a method is needed for the RPs to exchange information about active sources. This information exchange is done with MSDP.

In anycast RP, all the RPs are configured to be MSDP peers of each other. When a source registers with one RP, an SA message will be sent to the other RPs informing them that there is an active source for a particular multicast group. The result is that each RP will know about the active sources in the area of the other RPs. If any of the RPs were to fail, IP routing would converge, and one of the RPs would become the active RP in more than one area. New sources would register with the backup RP. Receivers would join the new RP and connectivity would be maintained.

The RP is normally needed only to start new sessions with sources and receivers. The RP facilitates the shared tree so that sources and receivers can establish a direct multicast data flow. If a multicast data flow is already established between a source and the receiver, an RP failure will not affect that session. Anycast RP ensures that new sessions with sources and receivers can begin at any time.

BSR Overview

BSR Election and Functionality

PIM uses the BSR to discover and announce RP-set information for each group prefix to all the routers in a PIM domain. This is the same function performed by Auto-RP, but the BSR is part of the PIM Version 2 specification. The BSR mechanism interoperates with Auto-RP on Cisco routers.

To avoid a single point of failure, you can configure several candidate BSRs in a PIM domain. A BSR is elected among the candidate BSRs automatically; they use bootstrap messages to discover which BSR has the highest priority. This router then announces to all PIM routers in the PIM domain that it is the BSR.

Following the election of the BSR, candidate RPs use unicast to announce to the BSR their willingness to be the RP. The BSR advertises the entire group-to-RP mapping set to the router link local address 224.0.0.13. Unlike the RP mapping agent in Auto-RP, which is used by Auto-RP to select the RP, every router in the BSR network is responsible for selecting the RP.

BSR lacks the ability to scope RP advertisements; however, BSR is used when vendor interoperability or open standard adherence is a requirement.

BSR Border Interface

A border interface in a PIM sparse mode domain requires precautions to prevent exchange of certain traffic with a neighboring domain reachable through that interface, especially if that domain is also running PIM sparse mode. BSR and Auto-RP messages should not be exchanged between different domains, because routers in one domain may elect RPs in the other domain, resulting in protocol malfunction or loss of isolation between the domains. Configure a BSR border interface to prevent BSR messages from being sent or received through an interface.

Static RP Overview

If you are configuring PIM sparse mode, you must configure a PIM RP for a multicast group. An RP can either be configured statically in each device, or learned through a dynamic mechanism. This task explains how to statically configure an RP, as opposed to the router learning the RP through a dynamic mechanism such as Auto-RP.

PIM designated routers (DRs) forward data from directly connected multicast sources to the RP for distribution down the shared tree. Data is forwarded to the RP in one of two ways. It is encapsulated in register packets and unicast directly to the RP, or, if the RP has itself joined the source tree, it is multicast forwarded per the RPF forwarding algorithm. Last hop routers directly connected to receivers may, at their discretion, join themselves to the source tree and prune themselves from the shared tree.

A single RP can be configured for multiple groups that are defined by an access list. If no RP is configured for a group, the router treats the group as dense using the PIM dense mode techniques. (You can prevent this occurrence by configuring the **no ip pim dm-fallback**command.)

If dynamic and static group-to-RP mappings are used together and there is an RP address conflict, the RP address configured for a static group-to-RP mapping (with the **ip pim rp-address override**command) will take precedence.

Note If the **override** keyword is not specified and there is RP address conflict, dynamic group-to-RP mappings will take precedence over static group-to-RP mappings.

SSM Overview

Source Specific Multicast (SSM). SSM is an extension of IP multicast where datagram traffic is forwarded to receivers from only those multicast sources that the receivers have explicitly joined. For multicast groups configured for SSM, only source-specific multicast distribution trees (not shared trees) are created.

SSM Components

Source Specific Multicast (SSM) is a datagram delivery model that best supports one-to-many applications, also known as broadcast applications. SSM is a core networking technology for the Cisco implementation of IP multicast solutions targeted for audio and video broadcast application environments and is described in RFC 3569. The following two components together support the implementation of SSM:

- Protocol Independent Multicast source-specific mode (PIM-SSM)
- Internet Group Management Protocol Version 3 (IGMPv3)

Protocol Independent Multicast (PIM) SSM, or PIM-SSM, is the routing protocol that supports the implementation of SSM and is derived from PIM sparse mode (PIM-SM). IGMP is the Internet Engineering Task Force (IETF) standards track protocol used for hosts to signal multicast group membership to routers. IGMP Version 3 supports source filtering, which is required for SSM. In order for SSM to run with IGMPv3, SSM must be supported in the device, the host where the application is running, and the application itself.

How SSM Differs from Internet Standard Multicast

The standard IP multicast infrastructure in the Internet and many enterprise intranets is based on the PIM-SM protocol and Multicast Source Discovery Protocol (MSDP). These protocols have proved to be reliable, extensive, and efficient. However, they are bound to the complexity and functionality limitations of the Internet Standard Multicast (ISM) service model. For example, with ISM, the network must maintain knowledge about which hosts in the network are actively sending multicast traffic. With SSM, this information is provided by receivers through the source addresses relayed to the last-hop devices by IGMPv3. SSM is an incremental response to the issues associated with ISM and is intended to coexist in the network with the protocols developed for ISM. In general, SSM provides IP multicast service for applications that utilize SSM.

ISM service is described in RFC 1112. This service consists of the delivery of IP datagrams from any source to a group of receivers called the multicast host group. The datagram traffic for the multicast host group consists of datagrams with an arbitrary IP unicast source address S and the multicast group address G as the IP destination address. Systems will receive this traffic by becoming members of the host group. Membership in a host group simply requires signaling the host group through IGMP Version 1, 2, or 3.

In SSM, delivery of datagrams is based on (S, G) channels. Traffic for one (S, G) channel consists of datagrams with an IP unicast source address S and the multicast group address G as the IP destination address. Systems will receive this traffic by becoming members of the (S, G) channel. In both SSM and ISM, no signaling is required to become a source. However, in SSM, receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources. In other words, receivers can receive traffic only from (S, G) channels to which they are subscribed, whereas in ISM, receivers need not know the IP addresses of sources from which they receive their traffic. The proposed standard approach for channel subscription signaling utilizes IGMP INCLUDE mode membership reports, which are supported only in IGMP Version 3.

SSM Operations

An established network in which IP multicast service is based on PIM-SM can support SSM services. SSM can also be deployed alone in a network without the full range of protocols that are required for interdomain PIM-SM. That is, SSM does not require an RP, so there is no need for an RP mechanism such as Auto-RP, MSDP, or bootstrap router (BSR).

If SSM is deployed in a network that is already configured for PIM-SM, then only the last-hop devices must be upgraded to a software image that supports SSM. Routers that are not directly connected to receivers do not have to upgrade to a software image that supports SSM. In general, these non-last-hop devices must only run PIM-SM in the SSM range. They may need additional access control configuration to suppress MSDP signaling, registering, or PIM-SM shared-tree operations from occurring within the SSM range.

The SSM mode of operation is enabled by configuring the SSM range using the **ip pim ssm** global configuration command. This configuration has the following effects:

- For groups within the SSM range, (S, G) channel subscriptions are accepted through IGMPv3 INCLUDE mode membership reports.
- PIM operations within the SSM range of addresses change to PIM-SSM, a mode derived from PIM-SM. In this mode, only PIM (S, G) Join and Prune messages are generated by the device. Incoming messages related to rendezvous point tree (RPT) operations are ignored or rejected, and incoming PIM register messages are immediately answered with Register-Stop messages. PIM-SSM is backward-compatible with PIM-SM unless a device is a last-hop device. Therefore, devices that are not last-hop devices can run PIM-SM for SSM groups (for example, if they do not yet support SSM).
- For groups within the SSM range, no MSDP Source-Active (SA) messages within the SSM range will be accepted, generated, or forwarded.

IGMPv3 Host Signaling

IGMPv3 is the third version of the IETF standards track protocol in which hosts signal membership to last-hop devices of multicast groups. IGMPv3 introduces the ability for hosts to signal group membership that allows filtering capabilities with respect to sources. A host can signal either that it wants to receive traffic from all sources sending to a group except for some specific sources (a mode called EXCLUDE) or that it wants to receive traffic only from some specific sources sending to the group (a mode called INCLUDE).

IGMPv3 can operate with both ISM and SSM. In ISM, both EXCLUDE and INCLUDE mode reports are accepted by the last-hop router. In SSM, only INCLUDE mode reports are accepted by the last-hop router.

Benefits of Source Specific Multicast

IP Multicast Address Management Not Required

In the ISM service, applications must acquire a unique IP multicast group address because traffic distribution is based only on the IP multicast group address used. If two applications with different sources and receivers use the same IP multicast group address, then receivers of both applications will receive traffic from the senders of both applications. Even though the receivers, if programmed appropriately, can filter out the unwanted traffic, this situation would cause generally unacceptable levels of unwanted traffic.

Allocating a unique IP multicast group address for an application is still a problem. Most short-lived applications use mechanisms like Session Description Protocol (SDP) and Session Announcement Protocol (SAP) to get a random address, a solution that does not work well with a rising number of applications in the Internet. The best current solution for long-lived applications is described in RFC 2770, but this solution suffers from the restriction that each autonomous system is limited to only 255 usable IP multicast addresses.

In SSM, traffic from each source is forwarded between devices in the network independent of traffic from other sources. Thus different sources can reuse multicast group addresses in the SSM range.

Denial of Service Attacks from Unwanted Sources Inhibited

In SSM, multicast traffic from each individual source will be transported across the network only if it was requested (through IGMPv3, IGMP v3lite, or URD memberships) from a receiver. In contrast, ISM forwards traffic from any active source sending to a multicast group to all receivers requesting that multicast group. In Internet broadcast applications, this ISM behavior is highly undesirable because it allows unwanted sources to easily disturb the actual Internet broadcast source by simply sending traffic to the same multicast group. This situation depletes bandwidth at the receiver side with unwanted traffic and thus disrupts the undisturbed reception of the Internet broadcast. In SSM, this type of denial of service (DoS) attack cannot be made by simply sending traffic to a multicast group.

Easy to Install and Manage

SSM is easy to install and provision in a network because it does not require the network to maintain which active sources are sending to multicast groups. This requirement exists in ISM (with IGMPv1, IGMPv2, or IGMPv3).

The current standard solutions for ISM service are PIM-SM and MSDP. Rendezvous point (RP) management in PIM-SM (including the necessity for Auto-RP or BSR) and MSDP is required only for the network to learn about active sources. This management is not necessary in SSM, which makes SSM easier than ISM to install and manage, and therefore easier than ISM to operationally scale in deployment. Another factor that contributes to the ease of installation of SSM is the fact that it can leverage preexisting PIM-SM networks and requires only the upgrade of last hop devices to support IGMPv3, IGMP v3lite, or URD.

Ideal for Internet Broadcast Applications

The three benefits previously described make SSM ideal for Internet broadcast-style applications for the following reasons:

- The ability to provide Internet broadcast services through SSM without the need for unique IP multicast addresses allows content providers to easily offer their service (IP multicast address allocation has been a serious problem for content providers in the past).
- The prevention against DoS attacks is an important factor for Internet broadcast services because, with their exposure to a large number of receivers, they are the most common targets for such attacks.
- The ease of installation and operation of SSM makes it ideal for network operators, especially in those cases where content needs to be forwarded between multiple independent PIM domains (because there is no need to manage MSDP for SSM between PIM domains).

Bidir-PIM Overview

Bidir-PIM shares many of its shortest path tree (SPT) operations with PIM-SM. Bidir-PIM also has unconditional forwarding of source traffic toward the RP upstream on the shared tree, but has no registering process for sources as in PIM-SM. These modifications allow forwarding of traffic in all routers based solely on the (*, G) multicast routing entries. This form of forwarding eliminates any source-specific state and allows scaling capability to an arbitrary number of sources.

Multicast Group Modes

In PIM, packet traffic for a multicast group is routed according to the rules of the mode configured for that multicast group. The Cisco implementation of PIM supports four modes for a multicast group:

- PIM bidirectional mode
- PIM dense mode
- PIM sparse mode
- PIM Source Specific Mode (SSM)

A router can simultaneously support all four modes or any combination of them for different multicast groups.

Bidirectional Shared Tree

In bidirectional mode, traffic is routed only along a bidirectional shared tree that is rooted at the rendezvous point (RP) for the group. In bidir-PIM, the IP address of the RP acts as the key to having all routers establish a loop-free spanning tree topology rooted in that IP address. This IP address need not be a router, but can be any unassigned IP address on a network that is reachable throughout the PIM domain. This technique is the preferred configuration method for establishing a redundant RP configuration for bidir-PIM.

Membership in a bidirectional group is signaled by way of explicit Join messages. Traffic from sources is unconditionally sent up the shared tree toward the RP and passed down the tree toward the receivers on each branch of the tree.

The figures below show the difference in state created per router for a unidirectional shared tree and source tree versus a bidirectional shared tree.

Figure 5: Unidirectional Shared Tree and Source Tree

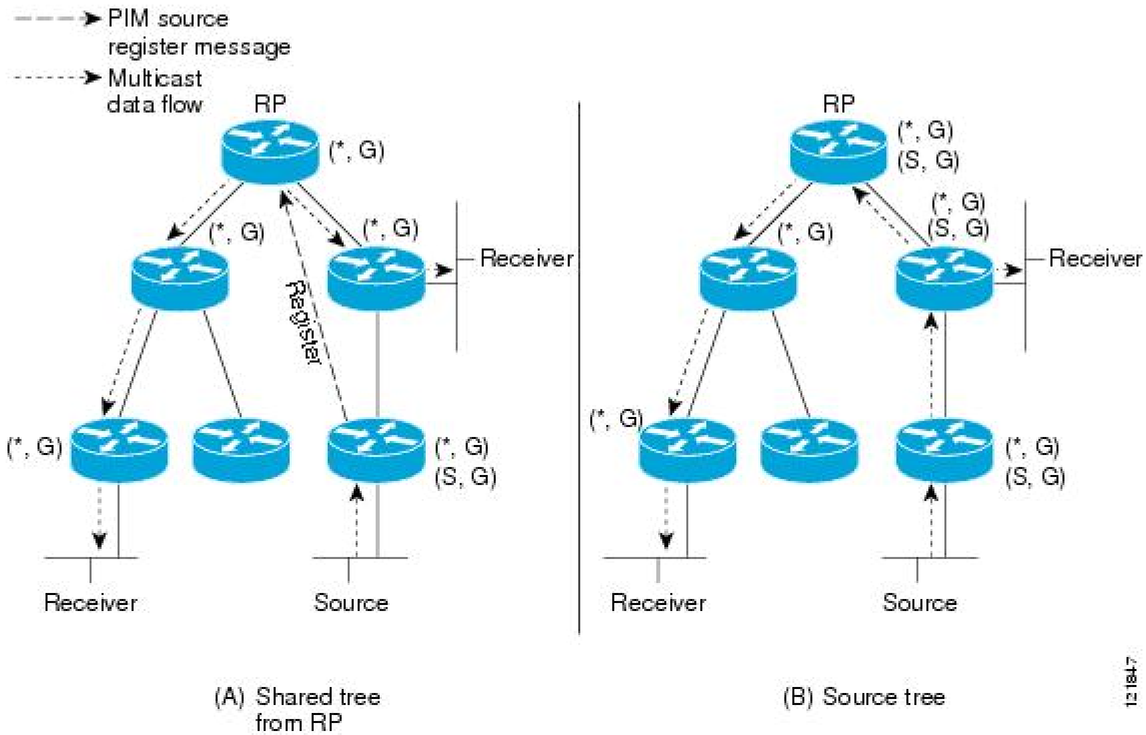
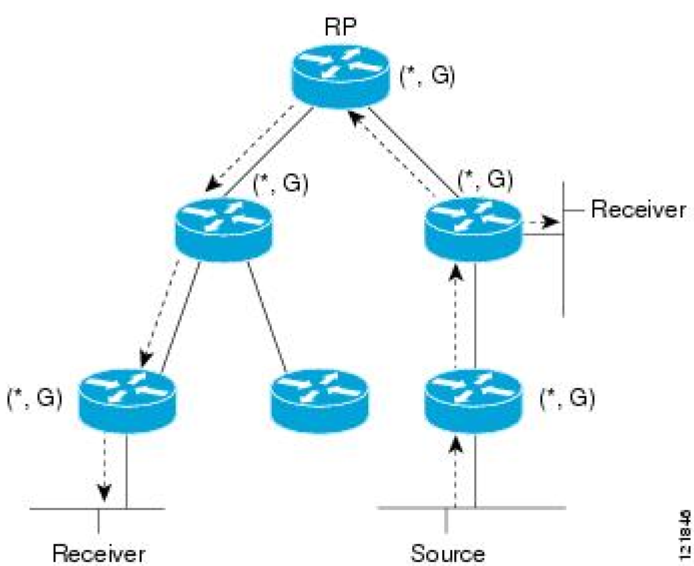


Figure 6: Bidirectional Shared Tree



For packets that are forwarded downstream from the RP toward receivers, there are no fundamental differences between bidir-PIM and PIM-SM. Bidir-PIM deviates substantially from PIM-SM for traffic that is passed from sources upstream toward the RP.

PIM-SM cannot forward traffic in the upstream direction of a tree because it accepts traffic from only one Reverse Path Forwarding (RPF) interface. This interface (for the shared tree) points toward the RP, thus allowing only downstream traffic flow. Upstream traffic is first encapsulated into unicast register messages, which are passed from the designated router (DR) of the source toward the RP. Second, the RP joins an SPT that is rooted at the source. Therefore, in PIM-SM, traffic from sources destined for the RP does not flow upstream in the shared tree, but downstream along the SPT of the source until it reaches the RP. From the RP, traffic flows along the shared tree toward all receivers.

In bidir-PIM, the packet-forwarding rules have been improved over PIM-SM, allowing traffic to be passed up the shared tree toward the RP. To avoid multicast packet looping, bidir-PIM introduces a new mechanism called designated forwarder (DF) election, which establishes a loop-free SPT rooted at the RP.

DF Election

On every network segment and point-to-point link, all PIM routers participate in a procedure called designated forwarder (DF) election. The procedure selects one router as the DF for every RP of bidirectional groups. This router is responsible for forwarding multicast packets received on that network.

The DF election is based on unicast routing metrics. The router with the most preferred unicast routing metric to the RP becomes the DF. Use of this method ensures that only one copy of every packet will be sent to the RP, even if there are parallel equal-cost paths to the RP.

A DF is selected for every RP of bidirectional groups. As a result, multiple routers may be elected as DF on any network segment, one for each RP. Any particular router may be elected as DF on more than one interface.

Bidirectional Group Tree Building

The procedure for joining the shared tree of a bidirectional group is almost identical to that used in PIM-SM. One main difference is that, for bidirectional groups, the role of the DR is assumed by the DF for the RP.

On a network that has local receivers, only the router elected as the DF populates the outgoing interface list (olist) upon receiving Internet Group Management Protocol (IGMP) Join messages, and sends (*, G) Join and Leave messages upstream toward the RP. When a downstream router wishes to join the shared tree, the RPF neighbor in the PIM Join and Leave messages is always the DF elected for the interface that lead to the RP.

When a router receives a Join or Leave message, and the router is not the DF for the receiving interface, the message is ignored. Otherwise, the router updates the shared tree in the same way as in sparse mode.

In a network where all routers support bidirectional shared trees, (S, G) Join and Leave messages are ignored. There is also no need to send PIM assert messages because the DF election procedure eliminates parallel downstream paths from any RP. An RP never joins a path back to the source, nor will it send any register stops.

Packet Forwarding

A router creates (*, G) entries only for bidirectional groups. The olist of a (*, G) entry includes all the interfaces for which the router has been elected DF and that have received either an IGMP or PIM Join message. If a router is located on a sender-only branch, it will also create a (*, G) state, but the olist will not include any interfaces.

If a packet is received from the RPF interface toward the RP, the packet is forwarded downstream according to the olist of the (*, G) entry. Otherwise, only the router that is the DF for the receiving interface forwards the packet upstream toward the RP; all other routers must discard the packet.

Benefits of Bidirectional PIM

- Bidir-PIM removes the performance cost of maintaining a routing state table for a large number of sources.

- Bidir-PIM is designed to be used for many-to-many applications within individual PIM domains. Multicast groups in bidirectional PIM mode can scale to an arbitrary number of sources without incurring overhead due to the number of sources.

How to Configure Basic IP Multicast

The tasks described in this section configure the basic IP multicast modes. No single task in this section is required; however, at least one of the tasks must be performed to configure IP multicast in a network. More than one of the tasks may be needed.

Configuring Sparse Mode with Auto-RP

Before you begin

- An interface configured in sparse-dense mode is treated in either sparse mode or dense mode of operation, depending on the mode in which the multicast group operates. You must decide how to configure your interfaces.
- All access lists that are needed when Auto-RP is configured should be configured prior to beginning the configuration task.

Note

- If a group has no known RP and the interface is configured to be sparse-dense mode, the interface is treated as if it were in dense mode, and data is flooded over the interface. To avoid this data flooding, configure the Auto-RP listener and then configure the interface as sparse mode.
- When configuring Auto-RP, you must either configure the Auto-RP listener feature (Step 5) and specify sparse mode (Step 7) or specify sparse-dense mode (Step 8) .
- When you configure sparse-dense mode, dense mode failover may result in a network dense-mode flood. To avoid this condition, use PIM sparse mode with the Auto-RP listener feature.

Follow this procedure to configure auto-rendezvous point (Auto-RP). Auto-RP can also be optionally used with anycast RP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. Either perform Steps 5 through 7 or perform Steps 6 and 8.
5. **ip pim autorp listener**
6. **interface** *type number*
7. **ip pim sparse-mode**
8. **ip pim sparse-dense-mode**
9. **exit**
10. Repeat Steps 1 through 9 on all PIM interfaces.

11. **ip pim send-rp-announce** {*interface-type interface-number* | *ip-address*} **scope** *ttl-value* [**group-list** *access-list*] [**interval** *seconds*] [**bidir**]
12. **ip pim send-rp-discovery** [*interface-type interface-number*] **scope** *ttl-value* [**interval** *seconds*]
13. **ip pim rp-announce-filter rp-list** *access-list* **group-list** *access-list*
14. **no ip pim dm-fallback**
15. **interface** *type number*
16. **ip multicast boundary** *access-list* [**filter-autorp**]
17. **end**
18. **show ip pim autorp**
19. **show ip pim rp** [**mapping**] [*rp-address*]
20. **show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]
21. **show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**]	Enables IP multicast routing. • Use the **distributed** keyword to enabled Multicast Distributed Switching.
Step 4	Either perform Steps 5 through 7 or perform Steps 6 and 8.	--
Step 5	**ip pim autorp listener**	Causes IP multicast traffic for the two Auto-RP groups 209.165.201.1 and 209.165.201.22 to be PIM dense mode flooded across interfaces operating in PIM sparse mode. • Skip this step if you are configuring sparse-dense mode in Step 8.
Step 6	**interface** *type number* **Example:**	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 7	**ip pim sparse-mode** **Example:**	Enables PIM sparse mode on an interface. When configuring Auto-RP in sparse mode, you must also configure the Auto-RP listener in the next step. • Skip this step if you are configuring sparse-dense mode in Step 8.
Step 8	**ip pim sparse-dense-mode** **Example:**	Enables PIM sparse-dense mode on an interface.

	Command or Action	Purpose
		• Skip this step if you configured sparse mode in Step 7.
Step 9	**exit** **Example:**	Exits interface configuration mode and returns to global configuration mode.
Step 10	Repeat Steps 1 through 9 on all PIM interfaces.	--
Step 11	**ip pim send-rp-announce** {*interface-type interface-number* | *ip-address*} **scope** *ttl-value* [**group-list** *access-list*] [**interval** *seconds*] [**bidir**] **Example:**	Sends RP announcements out all PIM-enabled interfaces. • Perform this step on the RP device only. • Use the *interface-type*and *interface-number* arguments to define which IP address is to be used as the RP address. • Use the *ip-address* argument to specify a directly connected IP address as the RP address. **Note** If the *ip-address* argument is configured for this command, the RP-announce message will be sourced by the interface to which this IP address is connected (that is, the source address in the IP header of the RP-announce message is the IP address of that interface). • This example shows that the interface is enabled with a maximum of 31 hops. The IP address by which the device wants to be identified as RP is the IP address associated with loopback interface 0. Access list 5 describes the groups for which this device serves as RP.
Step 12	**ip pim send-rp-discovery** [*interface-type interface-number*] **scope** *ttl-value* [**interval** *seconds*] **Example:**	Configures the device to be an RP mapping agent. • Perform this step on RP mapping agent devices or on combined RP/RP mapping agent devices. **Note** Auto-RP allows the RP function to run separately on one device and the RP mapping agent to run on one or multiple devices. It is possible to deploy the RP and the RP mapping agent on a combined RP/RP mapping agent device. • Use the optional *interface-type* and *interface-number* arguments to define which IP address is to be used as the source address of the RP mapping agent. • Use the **scope** keyword and *ttl-value* argument to specify the Time-to-Live (TTL) value in the IP header of Auto-RP discovery messages.

	Command or Action	Purpose
		• Use the optional **interval** keyword and *seconds* argument to specify the interval at which Auto-RP discovery messages are sent. **Note** Lowering the interval at which Auto-RP discovery messages are sent from the default value of 60 seconds results in more frequent floodings of the group-to-RP mappings. In some network environments, the disadvantages of lowering the interval (more control packet overhead) may outweigh the advantages (more frequent group-to-RP mapping updates). • The example shows limiting the Auto-RP discovery messages to 31 hops on loopback interface 1.
Step 13	**ip pim rp-announce-filter rp-list** *access-list* **group-list** *access-list* **Example:**	Filters incoming RP announcement messages sent from candidate RPs (C-RPs) to the RP mapping agent. • Perform this step on the RP mapping agent only.
Step 14	**no ip pim dm-fallback** **Example:**	(Optional) Prevents PIM dense mode fallback. • Skip this step if all interfaces have been configured to operate in PIM sparse mode. **Note** The **no ip pim dm-fallback** command behavior is enabled by default if all the interfaces are configured to operate in PIM sparse mode (using the **ip pim sparse-mode** command).
Step 15	**interface** *type number*	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 16	**ip multicast boundary** *access-list* [**filter-autorp**] **Example:**	Configures an administratively scoped boundary. • Perform this step on the interfaces that are boundaries to other devices. • The access list is not shown in this task. • An access list entry that uses the **deny** keyword creates a multicast boundary for packets that match that entry.
Step 17	**end**	Returns to global configuration mode.
Step 18	**show ip pim autorp**	(Optional) Displays the Auto-RP information.
Step 19	**show ip pim rp** [**mapping**] [*rp-address*]	(Optional) Displays RPs known in the network and shows how the device learned about each RP.

	Command or Action	Purpose
Step 20	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]	(Optional) Displays the multicast groups having receivers that are directly connected to the device and that were learned through Internet Group Management Protocol (IGMP). • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 21	**show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*] **Example:**	(Optional) Displays the contents of the IP multicast routing (mroute) table.

What to Do Next

Proceed to the " Verifying IP Multicast Operation " module.

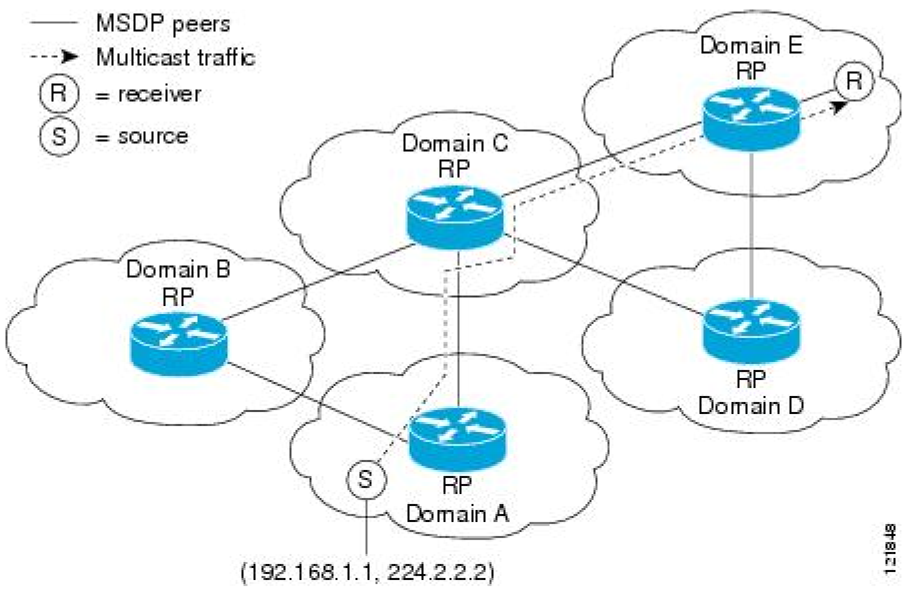
Configuring Sparse Mode with Anycast RP

This section describes how to configure sparse mode with anycast RP for RP redundancy.

Anycast RPs are configured statically, and interfaces are configured to operate in Protocol Independent Multicast-Sparse Mode (PIM-SM). In an anycast RP configuration, two or more RPs are configured with the same IP address on loopback interfaces. The Anycast RP loopback address should be configured with a 32-bit mask, making it a host address. An Anycast RP configuration is easy to configure and troubleshoot because the same host address is used as the RP address regardless of which router it is configured on.

Anycast RP allows two or more rendezvous points (RPs) to share the load for source registration and have the ability to act as hot backup routers for each other. Multicast Source Discovery Protocol (MSDP) is the key protocol that makes anycast RP possible.

Figure 7: MSDP Sharing Source Information Between RPs in Each Domain



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. **ip pim rp-address** *rp-address*
7. Repeat Steps 1 through 6 on two or more routers assigning the same RP address to each.
8. **interface loopback** [*interface-number*] **ip address** [*ip-address*] [*mask*]
9. **interface loopback** [*interface-number*] **ip address** [*ip-address*] [*mask*]
10. **exit**
11. **ip msdp peer** {*peer-name* | *peer-address*} [**connect-source** *interface-type* **interface-number**] [**remote-as** *as-number*]
12. **ip msdp originator-id loopback** [*interface*]
13. Repeat Steps 8 through 12 on the redundant RPs.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables sparse mode.
Step 6	**ip pim rp-address** *rp-address* **Example:** `Router(config-if)# ip pim rp-address 10.0.0.1`	Configures the address of a PIM RP for a particular group.
Step 7	Repeat Steps 1 through 6 on two or more routers assigning the same RP address to each.	--

Step 8	**interface loopback** [*interface-number*] **ip address** [*ip-address*] [*mask*]
	Example:
	`Router(config-if)# interface loopback 0`
	Example:
	`ip address 10.0.0.1 255.255.255.255`
		Configures the interface loopback IP address for the RP router.
		• Perform this step on the RP routers.

Step 9	**interface loopback** [*interface-number*] **ip address** [*ip-address*] [*mask*]
	Example:
	`Router(config-if)# interface loopback 1`
	Example:
	`ip address 10.1.1.1 255.255.255.255`
		Configures the interface loopback IP address for MSDP peering.

Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 11	**ip msdp peer** {*peer-name* | *peer-address*} [**connect-source** *interface-type* **interface-number**] [**remote-as** *as-number*] **Example:** `Router(config)# ip msdp peer 10.1.1.2 connect-source loopback 1`	Configures an MSDP peer. • Perform this step on the RP routers.
Step 12	**ip msdp originator-id loopback** [*interface*] **Example:** `Router(config)# ip msdp originator-id loopback 1`	Allows an MSDP speaker that originates a SA message to use the IP address of the interface as the RP address in the SA message. • Perform this step on the RP routers.
Step 13	Repeat Steps 8 through 12 on the redundant RPs.	--

What to Do Next

Proceed to the " Verifying IP Multicast Operation " module.

Configuring Sparse Mode with a Bootstrap Router

This section describes how to configure a bootstrap router (BSR), which provides a fault-tolerant, automated RP discovery and distribution mechanism so that routers learn the group-to-RP mappings dynamically.

Note The simultaneous deployment of Auto-RP and BSR is not supported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. **end**
7. Repeat Steps 1 through 6 on every multicast-enabled interface on every router.
8. **ip pim bsr-candidate** *interface-type interface-number* [*hash-mask-length* [*priority*]]
9. **ip pim rp-candidate** *interface-type interface-number* [group-list *access-list*] [**interval** seconds] [**priority** *value*]
10. Repeat Steps 8 through 10 on all RP and BSR routers.
11. **interface** *type number*
12. **ip pim bsr-border**
13. **end**
14. Repeat Steps 11 through 13 on all the routers that have boundary interfaces where the messages should not be sent or received.

15. **show ip pim rp** [**mapping**] [*rp-address*]
16. **show ip pim rp-hash** [*group-address*] [*group-name*]
17. **show ip pim bsr-router**
18. **show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]
19. **show ip mroute**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables sparse mode.
Step 6	**end** **Example:** `Router(config-if)# end`	Returns to global configuration mode.
Step 7	Repeat Steps 1 through 6 on every multicast-enabled interface on every router.	--
Step 8	**ip pim bsr-candidate** *interface-type interface-number* [*hash-mask-length* [*priority*]] **Example:** `Router(config)# ip pim bsr-candidate gigibitethernet 0/0/0 0 192`	Configures the router to announce its candidacy as a bootstrap router (BSR). • Perform this step on the RP or on combined RP/BSR routers.

	Command or Action	Purpose
		Note BSR allows the RP function to run separately on one router and the BSR to run on one or multiple routers. It is possible to deploy the RP and the BSR on a combined RP/BSR router. • This command configures the router to send BSR messages to all its PIM neighbors, with the address of the designated interface (configured for the *interface-type* and *interface-number*arguments) as the BSR address. • Use the optional *hash-mask-length* argument to set the length of a mask (32 bits maximum) that is to be ANDed with the group address before the PIMv2 hash function is called. All groups with the same seed hash (correspond) to the same RP. For example, if this value is 24, only the first 24 bits of the group addresses matter. The hash mask length allows one RP to be used for multiple groups. The default hash mask length is 0. • Use the optional *priority* argument (after you set the hash mask length) to specify the priority of the BSR as a C-RP. The priority range is from 0 to 255. The BSR C-RP with the highest priority (the lowest priority value) is preferred. If the priority values are the same, the router with the higher IP address is preferred. The default priority value is 0. **Note** The Cisco IOS and Cisco IOS XE implementation of PIM BSR uses the value 0 as the default priority for candidate RPs and BSRs. This implementation predates the draft-ietf-pim-sm-bsr IETF draft, the first IETF draft to specify 192 as the default priority value. The Cisco IOS and Cisco IOS XE implementation, thus, deviates from the IETF draft. To comply with the default priority value specified in the draft, you must explicitly set the priority value to 192.
Step 9	**ip pim rp-candidate** *interface-type interface-number* [group-list *access-list*] [**interval** seconds] [**priority** *value*] **Example:** `Router(config)# ip pim rp-candidate gigabitethernet 2/0/0 group-list 4 priority 192`	Configures the router to advertise itself as a PIM Version 2 candidate RP to the BSR. • Perform this step on the RP or on combined RP/BSR routers. **Note** BSR allows the RP function to run separately on one router and the BSR to run on one or multiple routers. It is possible to deploy the RP and the BSR on a combined RP/BSR router.

	Command or Action	Purpose
		• When an interval is specified, the candidate RP advertisement interval is set to the number of seconds specified. The default interval is 60 seconds. Tuning this interval down can reduce the time required to fail over to a secondary RP at the expense of generating more PIMv2 messages. • The Cisco IOS and Cisco IOS XE implementation of PIM BSR selects an RP from a set of candidate RPs using a method that is incompatible with the specification in RFC 2362. See the BSR and RFC 2362 Interoperable Candidate RP Example, on page 83 section for a configuration workaround. See CSCdy56806 using the Cisco Bug Toolkit for more information. **Note** The Cisco IOS and Cisco IOS XE implementation of PIM BSR uses the value 0 as the default priority for candidate RPs and BSRs. This implementation predates the draft-ietf-pim-sm-bsr IETF draft, the first IETF draft to specify 192 as the default priority value. The Cisco IOS and Cisco IOS XE implementation, thus, deviates from the IETF draft. To comply with the default priority value specified in the draft, you must explicitly set the priority value to 192.
Step 10	Repeat Steps 8 through 10 on all RP and BSR routers.	--
Step 11	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 12	**ip pim bsr-border** **Example:** `Router(config-if)# ip pim bsr-border`	Prevents the bootstrap router (BSR) messages from being sent or received through an interface. • See the BSR Border Interface, on page 57 section for more information.
Step 13	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 14	Repeat Steps 11 through 13 on all the routers that have boundary interfaces where the messages should not be sent or received.	--

	Command or Action	Purpose
Step 15	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `Router# show ip pim rp`	(Optional) Displays active rendezvous points (RPs) that are cached with associated multicast routing entries.
Step 16	**show ip pim rp-hash** [*group-address*] [*group-name*] **Example:** `Router# show ip pim rp-hash 239.1.1.1`	(Optional) Displays which rendezvous point (RP) is being selected for a specified group.
Step 17	**show ip pim bsr-router** **Example:** `Router# show ip pim bsr-router`	(Optional) Displays the bootstrap router (BSR) information.
Step 18	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**] **Example:** `Router# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the router and that were learned through IGMP. • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 19	**show ip mroute** **Example:** `Router# show ip mroute cbone-audio`	(Optional) Displays the contents of the IP mroute table.

What to Do Next

Proceed to the “ Verifying IP Multicast Operation ” module.

Configuring Sparse Mode with a Single Static RP(CLI)

A rendezvous point (RP) is required in networks running Protocol Independent Multicast sparse mode (PIM-SM). In PIM-SM, traffic will be forwarded only to network segments with active receivers that have explicitly requested multicast data.

This section describes how to configure sparse mode with a single static RP.

Before you begin

All access lists that are needed when sparse mode is configured with a single static RP should be configured prior to beginning the configuration task.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. Repeat Steps 1 through 5 on every interface that uses IP multicast.
7. **exit**
8. **ip pim rp-address** *rp-address* [*access-list*] [**override**]
9. **end**
10. **show ip pim rp** [**mapping**] [*rp-address*]
11. **show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]
12. **show ip mroute**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `device(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**interface** *type number* **Example:** `device(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode** **Example:** `device(config-if)# ip pim sparse-mode`	Enables PIM on an interface. You must use sparse mode.
Step 6	Repeat Steps 1 through 5 on every interface that uses IP multicast.	--
Step 7	**exit** **Example:** `device(config-if)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 8	**ip pim rp-address** *rp-address* [*access-list*] [**override**] **Example:** `device(config)# ip pim rp-address 192.168.0.0`	Configures the address of a PIM RP for a particular group. • The optional *access-list* argument is used to specify the number or name a standard access list that defines the multicast groups to be statically mapped to the RP. **Note** If no access list is defined, the RP will map to all multicast groups, 224/4. • The optional **override** keyword is used to specify that if dynamic and static group-to-RP mappings are used together and there is an RP address conflict, the RP address configured for a static group-to-RP mapping will take precedence. **Note** If the **override** keyword is not specified and there is RP address conflict, dynamic group-to-RP mappings will take precedence over static group-to-RP mappings.
Step 9	**end** **Example:** `device(config)# end`	Ends the current configuration session and returns to EXEC mode.
Step 10	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `device# show ip pim rp mapping`	(Optional) Displays RPs known in the network and shows how the router learned about each RP.
Step 11	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**] **Example:** `device# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the router and that were learned through IGMP. • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 12	**show ip mroute** **Example:** `device# show ip mroute`	(Optional) Displays the contents of the IP mroute table.

What to Do Next

Proceed to the " Verifying IP Multicast Operation " module.

Configuring Source Specific Multicast

Before you begin

If you want to use an access list to define the Source Specific Multicast (SSM) range, configure the access list before you reference the access list in the **ip pim ssm** command.

SUMMARY STEPS

1. **configure terminal**
2. **ip multicast-routing** [**distributed**]
3. **ip pim ssm** {**default** | **range** *access-list*}
4. **interface** *type number*
5. **ip pim sparse-mode**
6. Repeat Steps 1 through 6 on every interface that uses IP multicast.
7. **ip igmp version 3**
8. Repeat Step 8 on all host-facing interfaces.
9. **end**
10. **show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]
11. **show ip mroute**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `device# configure terminal`	Enters global configuration mode.
Step 2	**ip multicast-routing** [**distributed**] **Example:** `device(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 3	**ip pim ssm** {**default** | **range** *access-list*} **Example:** `device(config)# ip pim ssm default`	Configures SSM service. • The **default** keyword defines the SSM range access list as 232/8. • The **range** keyword specifies the standard IP access list number or name that defines the SSM range.
Step 4	**interface** *type number* **Example:** `device(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.

	Command or Action	Purpose
Step 5	**ip pim sparse-mode** **Example:** `device(config-if)# ip pim sparse-mode`	Enables PIM on an interface. You must use sparse mode.
Step 6	Repeat Steps 1 through 6 on every interface that uses IP multicast.	--
Step 7	**ip igmp version 3** **Example:** `device(config-if)# ip igmp version 3`	Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2. Version 3 is required by SSM.
Step 8	Repeat Step 8 on all host-facing interfaces.	--
Step 9	**end** **Example:** `device(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 10	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**] **Example:** `device# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the device and that were learned through IGMP. • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 11	**show ip mroute** **Example:** `device# show ip mroute`	(Optional) Displays the contents of the IP mroute table. • This command displays whether a multicast group is configured for SSM service or a source-specific host report has been received.

What to Do Next

Proceed to the " Verifying IP Multicast Operation " module.

Configuring Bidirectional PIM

Before you begin

All required access lists must be configured before configuring bidirectional PIM.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. **exit**
7. **ip pim bidir-enable**
8. **ip pim rp-address** *rp-address* [*access-list*] [**override**] **bidir**
9. **end**
10. Repeat Steps 2 through 9 on every multicast-enabled interface on every router.
11. **show ip pim rp** [**mapping**] [*rp-address*]
12. **show ip mroute**
13. **show ip pim interface** [*type number*] [**df** | **count**] [*rp-address*]
14. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:**	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**interface** *type number*	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode**	Enables sparse mode.
Step 6	**exit**	Returns to global configuration mode.
Step 7	**ip pim bidir-enable**	Enables bidir-PIM on a router. • Perform this step on every router.
Step 8	**ip pim rp-address** *rp-address* [*access-list*] [**override**] **bidir**	Configures the address of a PIM RP for a particular group. • Perform this step on every router. • This command defines the RP as bidirectional and defines the bidirectional group by way of the access list. • The optional **override** keyword is used to specify that if dynamic and static group-to-RP mappings are used together and there is an RP address conflict, the RP address configured for a static group-to-RP mapping will take precedence.

	Command or Action	Purpose
		Note If the **override** keyword is not specified and there is RP address conflict, dynamic group-to-RP mappings will take precedence over static group-to-RP mappings.
Step 9	**end**	Exits interface configuration mode and returns to privileged EXEC mode.
Step 10	Repeat Steps 2 through 9 on every multicast-enabled interface on every router.	--
Step 11	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `Device# show ip pim rp`	(Optional) Displays active RPs that are cached with associated multicast routing entries.
Step 12	**show ip mroute**	(Optional) Displays the contents of the IP mroute table.
Step 13	**show ip pim interface** [*type number*] [**df** | **count**] [*rp-address*] **Example:** `Device# show ip pim interface`	(Optional) Displays information about the elected DF for each RP of an interface, along with the unicast routing metric associated with the DF.
Step 14	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuration Examples for Basic IP Multicast

Example: Sparse Mode with Auto-RP

The following example configures sparse mode with Auto-RP:

```
ip multicast-routing
ip pim autorp listener
ip pim send-rp-announce Loopback0 scope 16 group-list 1
ip pim send-rp-discovery Loopback1 scope 16
no ip pim dm-fallback
access-list 1 permit 239.254.2.0 0.0.0.255
access-list 1 permit 239.254.3.0 0.0.0.255
.
.
.
access-list 10 permit 224.0.1.39
access-list 10 permit 224.0.1.40
access-list 10 permit 239.254.2.0 0.0.0.255
access-list 10 permit 239.254.3.0 0.0.0.255
```

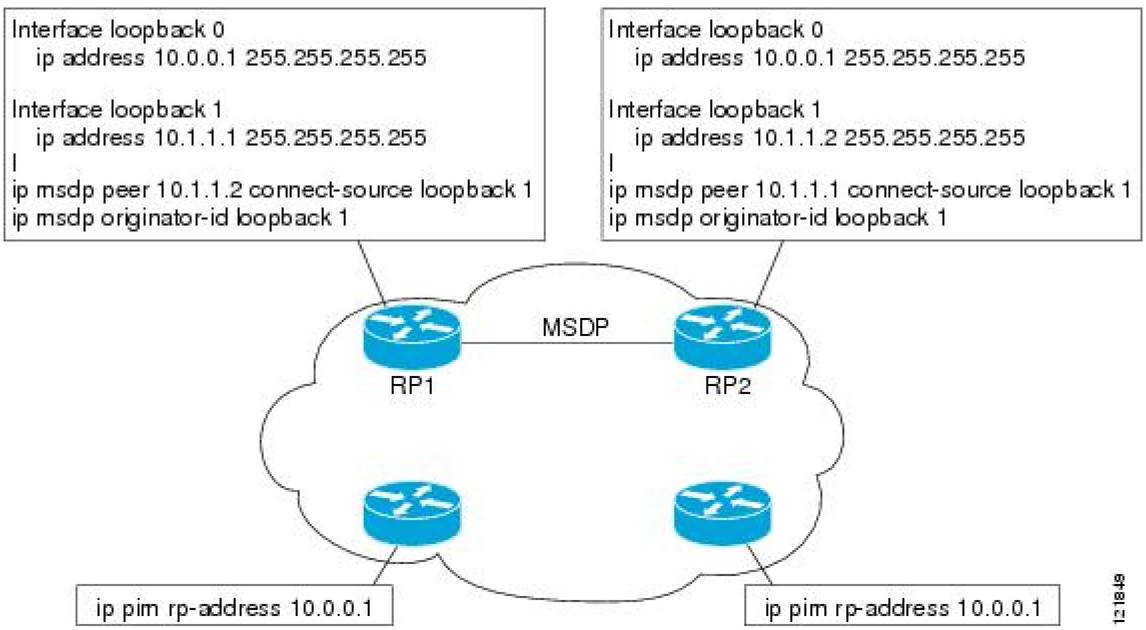
Sparse Mode with Anycast RP Example

The main purpose of an Anycast RP implementation is that the downstream multicast routers will have just one address for an RP. The example given in the figure below shows how loopback interface 0 of the RPs (RP1 and RP2) is configured with the 10.0.0.1 IP address. If this 10.0.0.1 address is configured on all RPs as the address for loopback interface 0 and then configured as the RP address, IP routing will converge on the closest RP. This address must be a host route; note the 255.255.255.255 subnet mask.

The downstream routers must be informed about the 10.0.0.1 RP address. In the figure below, the routers are configured statically with the **ip pim rp-address 10.0.0.1**global configuration command. This configuration could also be accomplished using the Auto-RP or bootstrap router (BSR) features.

The RPs in the figure must also share source information using MSDP. In this example, loopback interface 1 of the RPs (RP1 and RP2) is configured for MSDP peering. The MSDP peering address must be different from the anycast RP address.

Figure 8: AnyCast RP Configuration



Many routing protocols choose the highest IP address on loopback interfaces for the router ID. A problem may arise if the router selects the anycast RP address for the router ID. It is recommended that you avoid this problem by manually setting the router ID on the RPs to the same address as the MSDP peering address (for example, the loopback 1 address in the figure above). In Open Shortest Path First (OSPF), the router ID is configured using the **router-id**router configuration command. In Border Gateway Protocol (BGP), the router ID is configured using the **bgp router-id** router configuration command. In many BGP topologies, the MSDP peering address and the BGP peering address must be the same in order to pass the RPF check. The BGP peering address can be set using the **neighbor update-source** router configuration command.

The anycast RP example above uses IP addresses taken from RFC 1918. These IP addresses are normally blocked at interdomain borders and therefore are not accessible to other ISPs. You must use valid IP addresses if you want the RPs to be reachable from other domains.

The following example shows how to perform an Anycast RP configuration.

On RP 1

```
ip pim rp-address 10.0.0.1
interface loopback 0
 ip address 10.0.0.1 255.255.255.255
!
interface loopback 1
 ip address 10.1.1.1. 255.255.255.255
!
 ip msdp peer 10.1.1.2 connect-source loopback 1
 ip msdp originator-id loopback 1
```

On RP 2

```
ip pim rp-address 10.0.0.1
interface loopback 0
 ip address 10.0.0.1 255.255.255.255
interface loopback 1
 ip address 10.1.1.2. 255.255.255.255
!
 ip msdp peer 10.1.1.1 connect-source loopback 1
 ip msdp originator-id loopback 1
```

All Other Routers

```
ip pim rp-address 10.0.0.1
```

Sparse Mode with Bootstrap Router Example

The following example is a configuration for a candidate BSR, which also happens to be a candidate RP:

```
!
ip multicast-routing
!
interface GigabitEthernet0/0/0
 ip address 172.69.62.35 255.255.255.240
 ip pim sparse-mode
!
interface GigabitEthernet1/0/0
 ip address 172.21.24.18 255.255.255.248
 ip pim sparse-mode
!
interface GigabitEthernet2/0/0
 ip address 172.21.24.12 255.255.255.248
 ip pim sparse-mode
!
ip pim bsr-candidate GigabitEthernet2/0/0 30 10
ip pim rp-candidate GigabitEthernet2/0/0 group-list 5
access-list 5 permit 239.255.2.0 0.0.0.255
```

BSR and RFC 2362 Interoperable Candidate RP Example

When Cisco and non-Cisco routers are being operated in a single PIM domain with PIM Version 2 BSR, care must be taken when configuring candidate RPs because the Cisco implementation of the BSR RP selection is not fully compatible with RFC 2362.

RFC 2362 specifies that the BSR RP be selected as follows (RFC 2362, 3.7):

1. Select the candidate RP with the highest priority (lowest configured priority value).
2. If there is a tie in the priority level, select the candidate RP with the highest hash function value.
3. If there is a tie in the hash function value, select the candidate RP with the highest IP address.

Cisco routers always select the candidate RP based on the longest match on the announced group address prefix before selecting an RP based on priority, hash function, or IP address.

Inconsistent candidate RP selection between Cisco and non-Cisco RFC 2362-compliant routers in the same domain if multiple candidate RPs with partially overlapping group address ranges are configured can occur. Inconsistent candidate RP selection can prevent connectivity between sources and receivers in the PIM domain. A source may register with one candidate RP and a receiver may connect to a different candidate RP even though it is in the same group.

The following example shows a configuration that can cause inconsistent RP selection between a Cisco and a non-Cisco router in a single PIM domain with PIM Version 2 BSR:

```
access-list 10 permit 224.0.0.0  7.255.255.255
ip pim rp-candidate gigabitethernet1/0/0 group-list 10 priority 20
access-list 20 permit 224.0.0.0 15.255.255.255
ip pim rp-candidate gigabitethernet2/0/0 group-list 20 priority 10
```

In this example, a candidate RP on GigabitEthernet interface 1/0/0 announces a longer group prefix of 224.0.0.0/5 with a lower priority of 20. The candidate RP on GigabitEthernet interface 2/0/0 announces a shorter group prefix of 224.0.0.0/4 with a higher priority of 10. For all groups that match both ranges a Cisco router will always select the candidate RP on Ethernet interface 1 because it has the longer announced group prefix. A non-Cisco fully RFC 2362-compliant router will always select the candidate RP on GigabitEthernet interface 2/0/0 because it is configured with a higher priority.

To avoid this interoperability issue, do not configure different candidate RPs to announce partially overlapping group address prefixes. Configure any group prefixes that you want to announce from more than one candidate RP with the same group prefix length.

The following example shows how to configure the previous example so that there is no incompatibility between a Cisco router and a non-Cisco router in a single PIM domain with PIM Version 2 BSR:

```
access-list 10 permit 224.0.0.0  7.255.255.255
ip pim rp-candidate gigabitethernet1/0/0 group-list 10 priority 20
access-list 20 permit 224.0.0.0  7.255.255.255
access-list 20 permit 232.0.0.0  7.255.255.255
ip pim rp-candidate gigabitethernet2/0/0 group-list 20 priority 10
```

In this configuration the candidate RP on Ethernet interface 2 announces group address 224.0.0.0/5 and 232.0.0.0/5 which equal 224.0.0.0/4, but gives the interface the same group prefix length (5) as the candidate RP on Ethernet 1. As a result, both a Cisco router and an RFC 2362-compliant router will select the RP Ethernet interface 2.

Example: Sparse Mode with a Single Static RP

The following example sets the PIM RP address to 192.168.1.1 for all multicast groups and defines all groups to operate in sparse mode:

```
ip multicast-routing
interface gigiabitethernet 1/0/0
```

```
 ip pim sparse-mode
ip pim rp-address 192.168.1.1
```

Note The same RP cannot be used for both bidirectional and sparse mode groups.

The following example sets the PIM RP address to 172.16.1.1 for the multicast group 225.2.2.2 only:

```
access list 1 225.2.2.2 0.0.0.0
 ip pim rp-address 172.17.1.1
```

SSM with IGMPv3 Example

The following example shows how to configure a device (running IGMPv3) for SSM:

```
ip multicast-routing
!
interface GigabitEthernet3/1/0
 ip address 172.21.200.203 255.255.255.0
 description backbone interface
 ip pim sparse-mode
!
interface GigabitEthernet3/2/0
 ip address 131.108.1.2 255.255.255.0
 ip pim sparse-mode
 description ethernet connected to hosts
 ip igmp version 3
!
ip pim ssm default
```

SSM Filtering Example

The following example shows how to configure filtering on legacy RP routers running software releases that do not support SSM routing. This filtering will suppress all unwanted PIM-SM and MSDP traffic in the SSM range. Without this filtering, SSM will still operate, but there may be additional RPT traffic if legacy first hop and last hop routers exist in the network.

```
ip access-list extended  no-ssm-range
  deny   ip any 232.0.0.0 0.255.255.255 ! SSM range
  permit ip any any
! Deny registering in SSM range
ip pim accept-register list no-ssm-range
ip access-list extended msdp-nono-list
  deny   ip any 232.0.0.0 0.255.255.255 ! SSM Range
  ! .
  ! .
  ! .
  ! See ftp://ftpeng.cisco.com/ipmulticast/config-notes/msdp-sa-filter.txt for other SA
  ! messages that typically need to be filtered.
  permit ip any any
! Filter generated SA messages in SSM range. This configuration is only needed if there
! are directly connected sources to this router. The “ip pim accept-register” command
! filters remote sources.
ip msdp redistribute list msdp-nono-list
! Filter received SA messages in SSM range. “Filtered on receipt” means messages are
! neither processed or forwarded. Needs to be configured for each MSDP peer.
ip msdp sa-filter in msdp-peer1 list msdp-nono-list
```

```
! .
! .
! .
ip msdp sa-filter in msdp-peerN list msdp-nono-list
```

Bidir-PIM Example

By default, a bidirectional RP advertises all groups as bidirectional. An access list on the RP can be used to specify a list of groups to be advertised as bidirectional. Groups with the **deny** keyword will operate in dense mode. A different, nonbidirectional RP address is required for groups that operate in sparse mode because a single access list only allows either a **permit** or **deny** keyword.

The following example shows how to configure an RP for both sparse mode and bidirectional mode groups. The groups identified as 224/8 and 227/8 are bidirectional groups, and 226/8 is a sparse mode group. The RP must be configured to use different IP addresses for the sparse mode and bidirectional mode operations. Two loopback interfaces are used to allow this configuration. The addresses of these loopback interfaces must be routed throughout the PIM domain in such a way that the other routers in the PIM domain can communicate with the RP.

```
ip multicast-routing
!
.
.
.
!
interface loopback 0
 description One loopback address for this router’s Bidir Mode RP function
 ip address 10.0.1.1 255.255.255.0
!
interface loopback 1
description One loopback address for this router’s Sparse Mode RP function
 ip address 10.0.2.1 255.255.255.0
!
.
.
.
!
ip pim bidir-enable
ip pim rp-address 10.0.1.1 45 bidir
ip pim rp-address 10.0.2.1 46
!
access-list 45 permit 224.0.0.0 0.255.255.255
access-list 45 permit 227.0.0.0 0.255.255.255
access-list 46 permit 226.0.0.0 0.255.255.255
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference

Standards and RFCs

Standard/RFC	Title
draft-kouvelas-pim-bidir-new-00.txt	A New Proposal for Bi-directional PIM
RFC 1112	Host Extensions for IP Multicasting
RFC 1918	Address Allocation for Private Internets
RFC 2770	GLOP Addressing in 233/8
RFC 3569	An Overview of Source-Specific Multicast (SSM)

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Basic IP Multicast in IPv4 Networks

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 3

Configuring Basic IP Multicast

IP multicast is a bandwidth-conserving technology that reduces traffic by delivering a single stream of information simultaneously to potentially thousands of corporate businesses and homes. Applications that take advantage of multicast include video conferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news. This module describes the tasks used to configure basic IP multicast.

Prerequisites for Configuring Basic IP Multicast

- To determine which of the tasks contained in this module you will have to perform, you must decide which Protocol Independent Multicast (PIM) mode will be used. This determination is based on the applications you intend to support on your network.
- All access lists to be used with the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the "Creating an IP Access List and Applying It to an Interface" module of the *Security Configuration Guide: Access Control Lists* guide.

Information About Configuring Basic IP Multicast

Auto-RP Overview

The Role of Auto-RP in a PIM Network

Auto-RP automates the distribution of group-to-rendezvous point (RP) mappings in a PIM network. To make Auto-RP work, a device must be designated as an RP mapping agent, which receives the RP announcement messages from the RPs and arbitrates conflicts. The RP mapping agent then sends the consistent group-to-RP mappings to all other devices by way of dense mode flooding.

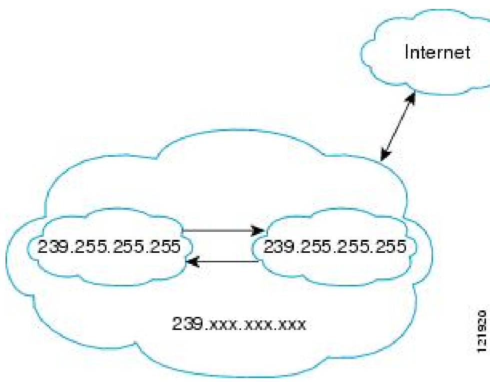
Thus, all routers automatically discover which RP to use for the groups they support. The Internet Assigned Numbers Authority (IANA) has assigned two group addresses, 224.0.1.39 and 224.0.1.40, for Auto-RP.

The mapping agent receives announcements of intention to become the RP from Candidate-RPs. The mapping agent then announces the winner of the RP election. This announcement is made independently of the decisions by the other mapping agents.

IP Multicast Boundary

As shown in the figure, address scoping defines domain boundaries so that domains with RPs that have the same IP address do not leak into each other. Scoping is performed on the subnet boundaries within large domains and on the boundaries between the domain and the Internet.

Figure 9: Address Scoping at Boundaries



You can set up an administratively scoped boundary on an interface for multicast group addresses using the **ip multicast boundary** command with the *access-list* argument. A standard access list defines the range of addresses affected. When a boundary is set up, no multicast data packets are allowed to flow across the boundary from either direction. The boundary allows the same multicast group address to be reused in different administrative domains.

The Internet Assigned Numbers Authority (IANA) has designated the multicast address range 239.0.0.0 to 239.255.255.255 as the administratively scoped addresses. This range of addresses can be reused in domains administered by different organizations. They would be considered local, not globally unique.

You can configure the **filter-autorp** keyword to examine and filter Auto-RP discovery and announcement messages at the administratively scoped boundary. Any Auto-RP group range announcements from the Auto-RP packets that are denied by the boundary access control list (ACL) are removed. An Auto-RP group range announcement is permitted and passed by the boundary only if all addresses in the Auto-RP group range are permitted by the boundary ACL. If any address is not permitted, the entire group range is filtered and removed from the Auto-RP message before the Auto-RP message is forwarded.

Benefits of Auto-RP in a PIM Network

- Auto-RP allows any change to the RP designation to be configured only on the devices that are RPs, not on the leaf routers.
- Auto-RP offers the ability to scope the RP address within a domain.

Anycast RP Overview

Anycast RP is a useful application of MSDP. Originally developed for interdomain multicast applications, MSDP used for Anycast RP is an intradomain feature that provides redundancy and load-sharing capabilities. Enterprise customers typically use Anycast RP for configuring a Protocol Independent Multicast sparse mode (PIM-SM) network to meet fault tolerance requirements within a single multicast domain.

In anycast RP, two or more RPs are configured with the same IP address on loopback interfaces. The anycast RP loopback address should be configured with a 32-bit mask, making it a host address. All the downstream routers should be configured so that the anycast RP loopback address is the IP address of their local RP. IP routing will automatically select the topologically closest RP for each source and receiver. Assuming that the sources are evenly spaced around the network, an equal number of sources will register with each RP. That is, the process of registering the sources will be shared equally by all the RPs in the network.

Because a source may register with one RP and receivers may join to a different RP, a method is needed for the RPs to exchange information about active sources. This information exchange is done with MSDP.

In anycast RP, all the RPs are configured to be MSDP peers of each other. When a source registers with one RP, an SA message will be sent to the other RPs informing them that there is an active source for a particular multicast group. The result is that each RP will know about the active sources in the area of the other RPs. If any of the RPs were to fail, IP routing would converge, and one of the RPs would become the active RP in more than one area. New sources would register with the backup RP. Receivers would join the new RP and connectivity would be maintained.

The RP is normally needed only to start new sessions with sources and receivers. The RP facilitates the shared tree so that sources and receivers can establish a direct multicast data flow. If a multicast data flow is already established between a source and the receiver, an RP failure will not affect that session. Anycast RP ensures that new sessions with sources and receivers can begin at any time.

BSR Overview

BSR Election and Functionality

PIM uses the BSR to discover and announce RP-set information for each group prefix to all the routers in a PIM domain. This is the same function performed by Auto-RP, but the BSR is part of the PIM Version 2 specification. The BSR mechanism interoperates with Auto-RP on Cisco routers.

To avoid a single point of failure, you can configure several candidate BSRs in a PIM domain. A BSR is elected among the candidate BSRs automatically; they use bootstrap messages to discover which BSR has the highest priority. This router then announces to all PIM routers in the PIM domain that it is the BSR.

Following the election of the BSR, candidate RPs use unicast to announce to the BSR their willingness to be the RP. The BSR advertises the entire group-to-RP mapping set to the router link local address 224.0.0.13. Unlike the RP mapping agent in Auto-RP, which is used by Auto-RP to select the RP, every router in the BSR network is responsible for selecting the RP.

BSR lacks the ability to scope RP advertisements; however, BSR is used when vendor interoperability or open standard adherence is a requirement.

BSR Border Interface

A border interface in a PIM sparse mode domain requires precautions to prevent exchange of certain traffic with a neighboring domain reachable through that interface, especially if that domain is also running PIM sparse mode. BSR and Auto-RP messages should not be exchanged between different domains, because routers in one domain may elect RPs in the other domain, resulting in protocol malfunction or loss of isolation between the domains. Configure a BSR border interface to prevent BSR messages from being sent or received through an interface.

Static RP Overview

If you are configuring PIM sparse mode, you must configure a PIM RP for a multicast group. An RP can either be configured statically in each device, or learned through a dynamic mechanism. This task explains how to statically configure an RP, as opposed to the router learning the RP through a dynamic mechanism such as Auto-RP.

PIM designated routers (DRs) forward data from directly connected multicast sources to the RP for distribution down the shared tree. Data is forwarded to the RP in one of two ways. It is encapsulated in register packets and unicast directly to the RP, or, if the RP has itself joined the source tree, it is multicast forwarded per the RPF forwarding algorithm. Last hop routers directly connected to receivers may, at their discretion, join themselves to the source tree and prune themselves from the shared tree.

A single RP can be configured for multiple groups that are defined by an access list. If no RP is configured for a group, the router treats the group as dense using the PIM dense mode techniques. (You can prevent this occurrence by configuring the **no ip pim dm-fallback**command.)

If dynamic and static group-to-RP mappings are used together and there is an RP address conflict, the RP address configured for a static group-to-RP mapping (with the **ip pim rp-address override**command) will take precedence.

Note If the **override** keyword is not specified and there is RP address conflict, dynamic group-to-RP mappings will take precedence over static group-to-RP mappings.

SSM Overview

Source Specific Multicast (SSM). SSM is an extension of IP multicast where datagram traffic is forwarded to receivers from only those multicast sources that the receivers have explicitly joined. For multicast groups configured for SSM, only source-specific multicast distribution trees (not shared trees) are created.

SSM Components

Source Specific Multicast (SSM) is a datagram delivery model that best supports one-to-many applications, also known as broadcast applications. SSM is a core networking technology for the Cisco implementation of IP multicast solutions targeted for audio and video broadcast application environments and is described in RFC 3569. The following two components together support the implementation of SSM:

- Protocol Independent Multicast source-specific mode (PIM-SSM)
- Internet Group Management Protocol Version 3 (IGMPv3)

Protocol Independent Multicast (PIM) SSM, or PIM-SSM, is the routing protocol that supports the implementation of SSM and is derived from PIM sparse mode (PIM-SM). IGMP is the Internet Engineering Task Force (IETF) standards track protocol used for hosts to signal multicast group membership to routers. IGMP Version 3 supports source filtering, which is required for SSM. In order for SSM to run with IGMPv3, SSM must be supported in the device, the host where the application is running, and the application itself.

How SSM Differs from Internet Standard Multicast

The standard IP multicast infrastructure in the Internet and many enterprise intranets is based on the PIM-SM protocol and Multicast Source Discovery Protocol (MSDP). These protocols have proved to be reliable, extensive, and efficient. However, they are bound to the complexity and functionality limitations of the Internet Standard Multicast (ISM) service model. For example, with ISM, the network must maintain knowledge about which hosts in the network are actively sending multicast traffic. With SSM, this information is provided by receivers through the source addresses relayed to the last-hop devices by IGMPv3. SSM is an incremental response to the issues associated with ISM and is intended to coexist in the network with the protocols developed for ISM. In general, SSM provides IP multicast service for applications that utilize SSM.

ISM service is described in RFC 1112. This service consists of the delivery of IP datagrams from any source to a group of receivers called the multicast host group. The datagram traffic for the multicast host group consists of datagrams with an arbitrary IP unicast source address S and the multicast group address G as the IP destination address. Systems will receive this traffic by becoming members of the host group. Membership in a host group simply requires signaling the host group through IGMP Version 1, 2, or 3.

In SSM, delivery of datagrams is based on (S, G) channels. Traffic for one (S, G) channel consists of datagrams with an IP unicast source address S and the multicast group address G as the IP destination address. Systems will receive this traffic by becoming members of the (S, G) channel. In both SSM and ISM, no signaling is required to become a source. However, in SSM, receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources. In other words, receivers can receive traffic only from (S, G) channels to which they are subscribed, whereas in ISM, receivers need not know the IP addresses of sources from which they receive their traffic. The proposed standard approach for channel subscription signaling utilizes IGMP INCLUDE mode membership reports, which are supported only in IGMP Version 3.

SSM Operations

An established network in which IP multicast service is based on PIM-SM can support SSM services. SSM can also be deployed alone in a network without the full range of protocols that are required for interdomain PIM-SM. That is, SSM does not require an RP, so there is no need for an RP mechanism such as Auto-RP, MSDP, or bootstrap router (BSR).

If SSM is deployed in a network that is already configured for PIM-SM, then only the last-hop devices must be upgraded to a software image that supports SSM. Routers that are not directly connected to receivers do not have to upgrade to a software image that supports SSM. In general, these non-last-hop devices must only run PIM-SM in the SSM range. They may need additional access control configuration to suppress MSDP signaling, registering, or PIM-SM shared-tree operations from occurring within the SSM range.

The SSM mode of operation is enabled by configuring the SSM range using the **ip pim ssm** global configuration command. This configuration has the following effects:

- For groups within the SSM range, (S, G) channel subscriptions are accepted through IGMPv3 INCLUDE mode membership reports.
- PIM operations within the SSM range of addresses change to PIM-SSM, a mode derived from PIM-SM. In this mode, only PIM (S, G) Join and Prune messages are generated by the device. Incoming messages related to rendezvous point tree (RPT) operations are ignored or rejected, and incoming PIM register messages are immediately answered with Register-Stop messages. PIM-SSM is backward-compatible with PIM-SM unless a device is a last-hop device. Therefore, devices that are not last-hop devices can run PIM-SM for SSM groups (for example, if they do not yet support SSM).
- For groups within the SSM range, no MSDP Source-Active (SA) messages within the SSM range will be accepted, generated, or forwarded.

IGMPv3 Host Signaling

IGMPv3 is the third version of the IETF standards track protocol in which hosts signal membership to last-hop devices of multicast groups. IGMPv3 introduces the ability for hosts to signal group membership that allows filtering capabilities with respect to sources. A host can signal either that it wants to receive traffic from all sources sending to a group except for some specific sources (a mode called EXCLUDE) or that it wants to receive traffic only from some specific sources sending to the group (a mode called INCLUDE).

IGMPv3 can operate with both ISM and SSM. In ISM, both EXCLUDE and INCLUDE mode reports are accepted by the last-hop router. In SSM, only INCLUDE mode reports are accepted by the last-hop router.

Benefits of Source Specific Multicast

IP Multicast Address Management Not Required

In the ISM service, applications must acquire a unique IP multicast group address because traffic distribution is based only on the IP multicast group address used. If two applications with different sources and receivers use the same IP multicast group address, then receivers of both applications will receive traffic from the senders of both applications. Even though the receivers, if programmed appropriately, can filter out the unwanted traffic, this situation would cause generally unacceptable levels of unwanted traffic.

Allocating a unique IP multicast group address for an application is still a problem. Most short-lived applications use mechanisms like Session Description Protocol (SDP) and Session Announcement Protocol (SAP) to get a random address, a solution that does not work well with a rising number of applications in the Internet. The best current solution for long-lived applications is described in RFC 2770, but this solution suffers from the restriction that each autonomous system is limited to only 255 usable IP multicast addresses.

In SSM, traffic from each source is forwarded between devices in the network independent of traffic from other sources. Thus different sources can reuse multicast group addresses in the SSM range.

Denial of Service Attacks from Unwanted Sources Inhibited

In SSM, multicast traffic from each individual source will be transported across the network only if it was requested (through IGMPv3, IGMP v3lite, or URD memberships) from a receiver. In contrast, ISM forwards traffic from any active source sending to a multicast group to all receivers requesting that multicast group. In Internet broadcast applications, this ISM behavior is highly undesirable because it allows unwanted sources to easily disturb the actual Internet broadcast source by simply sending traffic to the same multicast group. This situation depletes bandwidth at the receiver side with unwanted traffic and thus disrupts the undisturbed reception of the Internet broadcast. In SSM, this type of denial of service (DoS) attack cannot be made by simply sending traffic to a multicast group.

Easy to Install and Manage

SSM is easy to install and provision in a network because it does not require the network to maintain which active sources are sending to multicast groups. This requirement exists in ISM (with IGMPv1, IGMPv2, or IGMPv3).

The current standard solutions for ISM service are PIM-SM and MSDP. Rendezvous point (RP) management in PIM-SM (including the necessity for Auto-RP or BSR) and MSDP is required only for the network to learn about active sources. This management is not necessary in SSM, which makes SSM easier than ISM to install and manage, and therefore easier than ISM to operationally scale in deployment. Another factor that contributes to the ease of installation of SSM is the fact that it can leverage preexisting PIM-SM networks and requires only the upgrade of last hop devices to support IGMPv3, IGMP v3lite, or URD.

Ideal for Internet Broadcast Applications

The three benefits previously described make SSM ideal for Internet broadcast-style applications for the following reasons:

- The ability to provide Internet broadcast services through SSM without the need for unique IP multicast addresses allows content providers to easily offer their service (IP multicast address allocation has been a serious problem for content providers in the past).
- The prevention against DoS attacks is an important factor for Internet broadcast services because, with their exposure to a large number of receivers, they are the most common targets for such attacks.
- The ease of installation and operation of SSM makes it ideal for network operators, especially in those cases where content needs to be forwarded between multiple independent PIM domains (because there is no need to manage MSDP for SSM between PIM domains).

Bidir-PIM Overview

Bidir-PIM shares many of its shortest path tree (SPT) operations with PIM-SM. Bidir-PIM also has unconditional forwarding of source traffic toward the RP upstream on the shared tree, but has no registering process for sources as in PIM-SM. These modifications allow forwarding of traffic in all routers based solely on the (*, G) multicast routing entries. This form of forwarding eliminates any source-specific state and allows scaling capability to an arbitrary number of sources.

Multicast Group Modes

In PIM, packet traffic for a multicast group is routed according to the rules of the mode configured for that multicast group. The Cisco implementation of PIM supports four modes for a multicast group:

- PIM bidirectional mode
- PIM dense mode
- PIM sparse mode
- PIM Source Specific Mode (SSM)

A router can simultaneously support all four modes or any combination of them for different multicast groups.

Bidirectional Shared Tree

In bidirectional mode, traffic is routed only along a bidirectional shared tree that is rooted at the rendezvous point (RP) for the group. In bidir-PIM, the IP address of the RP acts as the key to having all routers establish a loop-free spanning tree topology rooted in that IP address. This IP address need not be a router, but can be any unassigned IP address on a network that is reachable throughout the PIM domain. This technique is the preferred configuration method for establishing a redundant RP configuration for bidir-PIM.

Membership in a bidirectional group is signaled by way of explicit Join messages. Traffic from sources is unconditionally sent up the shared tree toward the RP and passed down the tree toward the receivers on each branch of the tree.

The figures below show the difference in state created per router for a unidirectional shared tree and source tree versus a bidirectional shared tree.

Figure 10: Unidirectional Shared Tree and Source Tree

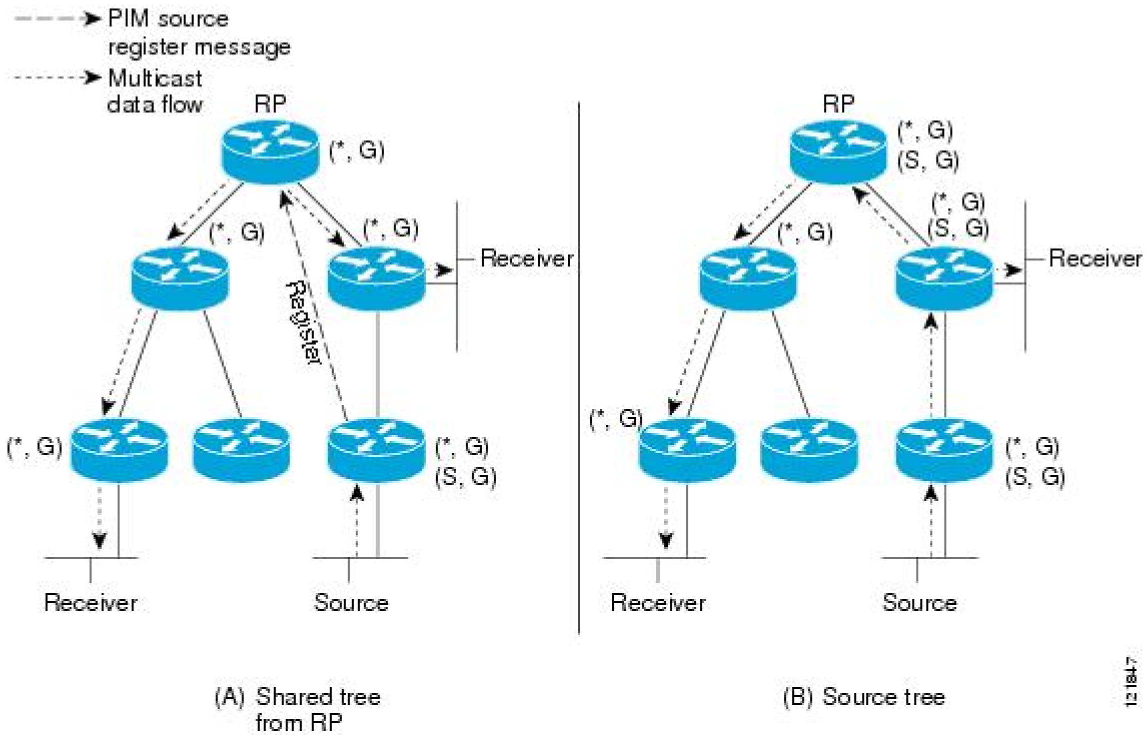
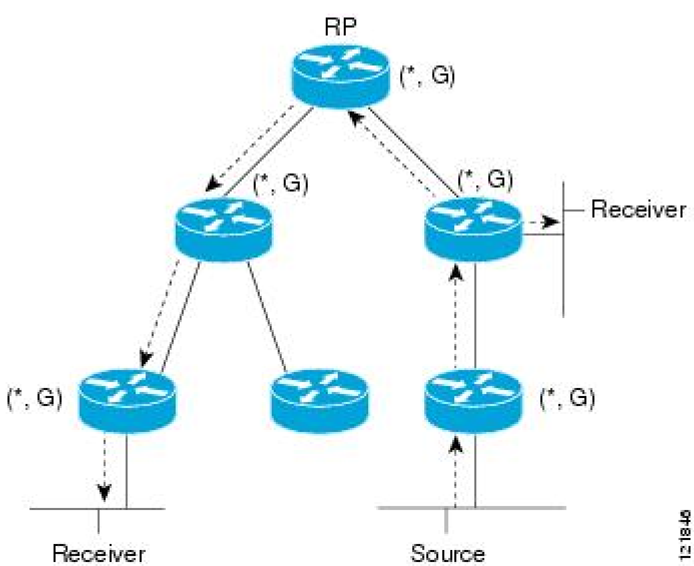


Figure 11: Bidirectional Shared Tree



For packets that are forwarded downstream from the RP toward receivers, there are no fundamental differences between bidir-PIM and PIM-SM. Bidir-PIM deviates substantially from PIM-SM for traffic that is passed from sources upstream toward the RP.

PIM-SM cannot forward traffic in the upstream direction of a tree because it accepts traffic from only one Reverse Path Forwarding (RPF) interface. This interface (for the shared tree) points toward the RP, thus allowing only downstream traffic flow. Upstream traffic is first encapsulated into unicast register messages, which are passed from the designated router (DR) of the source toward the RP. Second, the RP joins an SPT that is rooted at the source. Therefore, in PIM-SM, traffic from sources destined for the RP does not flow upstream in the shared tree, but downstream along the SPT of the source until it reaches the RP. From the RP, traffic flows along the shared tree toward all receivers.

In bidir-PIM, the packet-forwarding rules have been improved over PIM-SM, allowing traffic to be passed up the shared tree toward the RP. To avoid multicast packet looping, bidir-PIM introduces a new mechanism called designated forwarder (DF) election, which establishes a loop-free SPT rooted at the RP.

DF Election

On every network segment and point-to-point link, all PIM routers participate in a procedure called designated forwarder (DF) election. The procedure selects one router as the DF for every RP of bidirectional groups. This router is responsible for forwarding multicast packets received on that network.

The DF election is based on unicast routing metrics. The router with the most preferred unicast routing metric to the RP becomes the DF. Use of this method ensures that only one copy of every packet will be sent to the RP, even if there are parallel equal-cost paths to the RP.

A DF is selected for every RP of bidirectional groups. As a result, multiple routers may be elected as DF on any network segment, one for each RP. Any particular router may be elected as DF on more than one interface.

Bidirectional Group Tree Building

The procedure for joining the shared tree of a bidirectional group is almost identical to that used in PIM-SM. One main difference is that, for bidirectional groups, the role of the DR is assumed by the DF for the RP.

On a network that has local receivers, only the router elected as the DF populates the outgoing interface list (olist) upon receiving Internet Group Management Protocol (IGMP) Join messages, and sends (*, G) Join and Leave messages upstream toward the RP. When a downstream router wishes to join the shared tree, the RPF neighbor in the PIM Join and Leave messages is always the DF elected for the interface that lead to the RP.

When a router receives a Join or Leave message, and the router is not the DF for the receiving interface, the message is ignored. Otherwise, the router updates the shared tree in the same way as in sparse mode.

In a network where all routers support bidirectional shared trees, (S, G) Join and Leave messages are ignored. There is also no need to send PIM assert messages because the DF election procedure eliminates parallel downstream paths from any RP. An RP never joins a path back to the source, nor will it send any register stops.

Packet Forwarding

A router creates (*, G) entries only for bidirectional groups. The olist of a (*, G) entry includes all the interfaces for which the router has been elected DF and that have received either an IGMP or PIM Join message. If a router is located on a sender-only branch, it will also create a (*, G) state, but the olist will not include any interfaces.

If a packet is received from the RPF interface toward the RP, the packet is forwarded downstream according to the olist of the (*, G) entry. Otherwise, only the router that is the DF for the receiving interface forwards the packet upstream toward the RP; all other routers must discard the packet.

Benefits of Bidirectional PIM

- Bidir-PIM removes the performance cost of maintaining a routing state table for a large number of sources.

- Bidir-PIM is designed to be used for many-to-many applications within individual PIM domains. Multicast groups in bidirectional PIM mode can scale to an arbitrary number of sources without incurring overhead due to the number of sources.

How to Configure Basic IP Multicast

The tasks described in this section configure the basic IP multicast modes. No single task in this section is required; however, at least one of the tasks must be performed to configure IP multicast in a network. More than one of the tasks may be needed.

Configuring Sparse Mode with Auto-RP

Before you begin

- An interface configured in sparse-dense mode is treated in either sparse mode or dense mode of operation, depending on the mode in which the multicast group operates. You must decide how to configure your interfaces.
- All access lists that are needed when Auto-RP is configured should be configured prior to beginning the configuration task.

Note

- If a group has no known RP and the interface is configured to be sparse-dense mode, the interface is treated as if it were in dense mode, and data is flooded over the interface. To avoid this data flooding, configure the Auto-RP listener and then configure the interface as sparse mode.
- When configuring Auto-RP, you must either configure the Auto-RP listener feature (Step 5) and specify sparse mode (Step 7) or specify sparse-dense mode (Step 8) .
- When you configure sparse-dense mode, dense mode failover may result in a network dense-mode flood. To avoid this condition, use PIM sparse mode with the Auto-RP listener feature.

Follow this procedure to configure auto-rendezvous point (Auto-RP). Auto-RP can also be optionally used with anycast RP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. Either perform Steps 5 through 7 or perform Steps 6 and 8.
5. **ip pim autorp listener**
6. **interface** *type number*
7. **ip pim sparse-mode**
8. **ip pim sparse-dense-mode**
9. **exit**
10. Repeat Steps 1 through 9 on all PIM interfaces.

11. **ip pim send-rp-announce** {*interface-type interface-number* | *ip-address*} **scope** *ttl-value* [**group-list** *access-list*] [**interval** *seconds*] [**bidir**]
12. **ip pim send-rp-discovery** [*interface-type interface-number*] **scope** *ttl-value* [**interval** *seconds*]
13. **ip pim rp-announce-filter rp-list** *access-list* **group-list** *access-list*
14. **no ip pim dm-fallback**
15. **interface** *type number*
16. **ip multicast boundary** *access-list* [**filter-autorp**]
17. **end**
18. **show ip pim autorp**
19. **show ip pim rp** [**mapping**] [*rp-address*]
20. **show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]
21. **show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**]	Enables IP multicast routing. • Use the **distributed** keyword to enabled Multicast Distributed Switching.
Step 4	Either perform Steps 5 through 7 or perform Steps 6 and 8.	--
Step 5	**ip pim autorp listener**	Causes IP multicast traffic for the two Auto-RP groups 209.165.201.1 and 209.165.201.22 to be PIM dense mode flooded across interfaces operating in PIM sparse mode. • Skip this step if you are configuring sparse-dense mode in Step 8.
Step 6	**interface** *type number* **Example:**	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 7	**ip pim sparse-mode** **Example:**	Enables PIM sparse mode on an interface. When configuring Auto-RP in sparse mode, you must also configure the Auto-RP listener in the next step. • Skip this step if you are configuring sparse-dense mode in Step 8.
Step 8	**ip pim sparse-dense-mode** **Example:**	Enables PIM sparse-dense mode on an interface.

	Command or Action	Purpose
		• Skip this step if you configured sparse mode in Step 7.
Step 9	**exit** **Example:**	Exits interface configuration mode and returns to global configuration mode.
Step 10	Repeat Steps 1 through 9 on all PIM interfaces.	--
Step 11	**ip pim send-rp-announce** {*interface-type interface-number* | *ip-address*} **scope** *ttl-value* [**group-list** *access-list*] [**interval** *seconds*] [**bidir**] **Example:**	Sends RP announcements out all PIM-enabled interfaces. • Perform this step on the RP device only. • Use the *interface-type*and *interface-number* arguments to define which IP address is to be used as the RP address. • Use the *ip-address* argument to specify a directly connected IP address as the RP address. **Note** If the *ip-address* argument is configured for this command, the RP-announce message will be sourced by the interface to which this IP address is connected (that is, the source address in the IP header of the RP-announce message is the IP address of that interface). • This example shows that the interface is enabled with a maximum of 31 hops. The IP address by which the device wants to be identified as RP is the IP address associated with loopback interface 0. Access list 5 describes the groups for which this device serves as RP.
Step 12	**ip pim send-rp-discovery** [*interface-type interface-number*] **scope** *ttl-value* [**interval** *seconds*] **Example:**	Configures the device to be an RP mapping agent. • Perform this step on RP mapping agent devices or on combined RP/RP mapping agent devices. **Note** Auto-RP allows the RP function to run separately on one device and the RP mapping agent to run on one or multiple devices. It is possible to deploy the RP and the RP mapping agent on a combined RP/RP mapping agent device. • Use the optional *interface-type* and *interface-number* arguments to define which IP address is to be used as the source address of the RP mapping agent. • Use the **scope** keyword and *ttl-value* argument to specify the Time-to-Live (TTL) value in the IP header of Auto-RP discovery messages.

	Command or Action	Purpose
		• Use the optional **interval** keyword and *seconds* argument to specify the interval at which Auto-RP discovery messages are sent. **Note** Lowering the interval at which Auto-RP discovery messages are sent from the default value of 60 seconds results in more frequent floodings of the group-to-RP mappings. In some network environments, the disadvantages of lowering the interval (more control packet overhead) may outweigh the advantages (more frequent group-to-RP mapping updates). • The example shows limiting the Auto-RP discovery messages to 31 hops on loopback interface 1.
Step 13	**ip pim rp-announce-filter rp-list** *access-list* **group-list** *access-list* **Example:**	Filters incoming RP announcement messages sent from candidate RPs (C-RPs) to the RP mapping agent. • Perform this step on the RP mapping agent only.
Step 14	**no ip pim dm-fallback** **Example:**	(Optional) Prevents PIM dense mode fallback. • Skip this step if all interfaces have been configured to operate in PIM sparse mode. **Note** The **no ip pim dm-fallback** command behavior is enabled by default if all the interfaces are configured to operate in PIM sparse mode (using the **ip pim sparse-mode** command).
Step 15	**interface** *type number*	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 16	**ip multicast boundary** *access-list* [**filter-autorp**] **Example:**	Configures an administratively scoped boundary. • Perform this step on the interfaces that are boundaries to other devices. • The access list is not shown in this task. • An access list entry that uses the **deny** keyword creates a multicast boundary for packets that match that entry.
Step 17	**end**	Returns to global configuration mode.
Step 18	**show ip pim autorp**	(Optional) Displays the Auto-RP information.
Step 19	**show ip pim rp** [**mapping**] [*rp-address*]	(Optional) Displays RPs known in the network and shows how the device learned about each RP.

	Command or Action	Purpose
Step 20	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]	(Optional) Displays the multicast groups having receivers that are directly connected to the device and that were learned through Internet Group Management Protocol (IGMP). • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 21	**show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*] **Example:**	(Optional) Displays the contents of the IP multicast routing (mroute) table.

What to Do Next

Proceed to the “ Verifying IP Multicast Operation ” module.

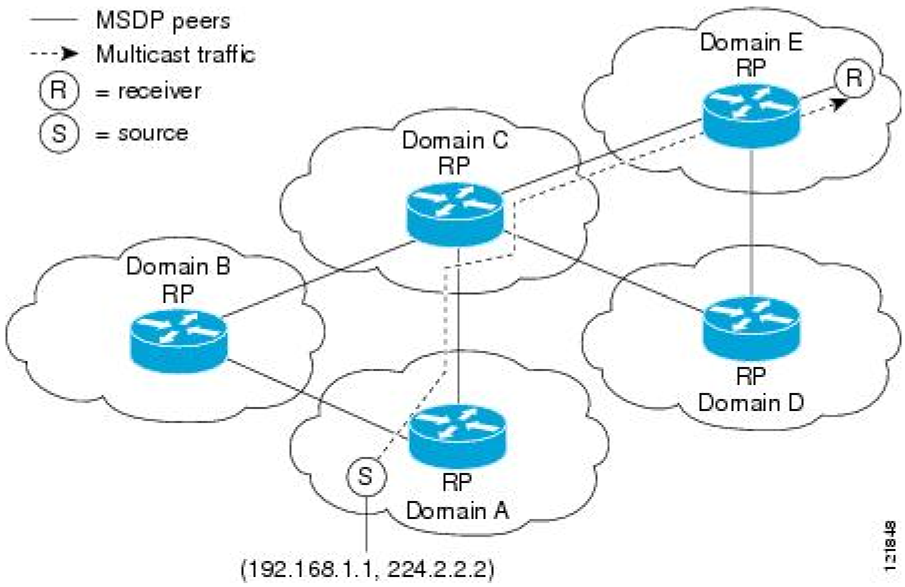
Configuring Sparse Mode with Anycast RP

This section describes how to configure sparse mode with anycast RP for RP redundancy.

Anycast RPs are configured statically, and interfaces are configured to operate in Protocol Independent Multicast-Sparse Mode (PIM-SM). In an anycast RP configuration, two or more RPs are configured with the same IP address on loopback interfaces. The Anycast RP loopback address should be configured with a 32-bit mask, making it a host address. An Anycast RP configuration is easy to configure and troubleshoot because the same host address is used as the RP address regardless of which router it is configured on.

Anycast RP allows two or more rendezvous points (RPs) to share the load for source registration and have the ability to act as hot backup routers for each other. Multicast Source Discovery Protocol (MSDP) is the key protocol that makes anycast RP possible.

Figure 12: MSDP Sharing Source Information Between RPs in Each Domain



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. **ip pim rp-address** *rp-address*
7. Repeat Steps 1 through 6 on two or more routers assigning the same RP address to each.
8. **interface loopback** [*interface-number*] **ip address** [*ip-address*] [*mask*]
9. **interface loopback** [*interface-number*] **ip address** [*ip-address*] [*mask*]
10. **exit**
11. **ip msdp peer** {*peer-name* | *peer-address*} [**connect-source** *interface-type* **interface-number**] [**remote-as** *as-number*]
12. **ip msdp originator-id loopback** [*interface*]
13. Repeat Steps 8 through 12 on the redundant RPs.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables sparse mode.
Step 6	**ip pim rp-address** *rp-address* **Example:** `Router(config-if)# ip pim rp-address 10.0.0.1`	Configures the address of a PIM RP for a particular group.
Step 7	Repeat Steps 1 through 6 on two or more routers assigning the same RP address to each.	--

Step 8	**interface loopback** [*interface-number*] **ip address** [*ip-address*] [*mask*]
	Example:
	`Router(config-if)# interface loopback 0`
	Example:
	`ip address 10.0.0.1 255.255.255.255`
		Configures the interface loopback IP address for the RP router.
		• Perform this step on the RP routers.

Step 9	**interface loopback** [*interface-number*] **ip address** [*ip-address*] [*mask*]
	Example:
	`Router(config-if)# interface loopback 1`
	Example:
	`ip address 10.1.1.1 255.255.255.255`
		Configures the interface loopback IP address for MSDP peering.

Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 11	**ip msdp peer** {*peer-name* | *peer-address*} [**connect-source** *interface-type* **interface-number**] [**remote-as** *as-number*] **Example:** `Router(config)# ip msdp peer 10.1.1.2 connect-source loopback 1`	Configures an MSDP peer. • Perform this step on the RP routers.
Step 12	**ip msdp originator-id loopback** [*interface*] **Example:** `Router(config)# ip msdp originator-id loopback 1`	Allows an MSDP speaker that originates a SA message to use the IP address of the interface as the RP address in the SA message. • Perform this step on the RP routers.
Step 13	Repeat Steps 8 through 12 on the redundant RPs.	--

What to Do Next

Proceed to the " Verifying IP Multicast Operation " module.

Configuring Sparse Mode with a Bootstrap Router

This section describes how to configure a bootstrap router (BSR), which provides a fault-tolerant, automated RP discovery and distribution mechanism so that routers learn the group-to-RP mappings dynamically.

Note The simultaneous deployment of Auto-RP and BSR is not supported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. **end**
7. Repeat Steps 1 through 6 on every multicast-enabled interface on every router.
8. **ip pim bsr-candidate** *interface-type interface-number* [*hash-mask-length* [*priority*]]
9. **ip pim rp-candidate** *interface-type interface-number* [group-list *access-list*] [**interval** seconds] [**priority** *value*]
10. Repeat Steps 8 through 10 on all RP and BSR routers.
11. **interface** *type number*
12. **ip pim bsr-border**
13. **end**
14. Repeat Steps 11 through 13 on all the routers that have boundary interfaces where the messages should not be sent or received.

15. **show ip pim rp** [**mapping**] [*rp-address*]
16. **show ip pim rp-hash** [*group-address*] [*group-name*]
17. **show ip pim bsr-router**
18. **show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]
19. **show ip mroute**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables sparse mode.
Step 6	**end** **Example:** `Router(config-if)# end`	Returns to global configuration mode.
Step 7	Repeat Steps 1 through 6 on every multicast-enabled interface on every router.	--
Step 8	**ip pim bsr-candidate** *interface-type interface-number* [*hash-mask-length* [*priority*]] **Example:** `Router(config)# ip pim bsr-candidate gigabitethernet 0/0/0 0 192`	Configures the router to announce its candidacy as a bootstrap router (BSR). • Perform this step on the RP or on combined RP/BSR routers.

	Command or Action	Purpose
		Note BSR allows the RP function to run separately on one router and the BSR to run on one or multiple routers. It is possible to deploy the RP and the BSR on a combined RP/BSR router. • This command configures the router to send BSR messages to all its PIM neighbors, with the address of the designated interface (configured for the *interface-type* and *interface-number*arguments) as the BSR address. • Use the optional *hash-mask-length* argument to set the length of a mask (32 bits maximum) that is to be ANDed with the group address before the PIMv2 hash function is called. All groups with the same seed hash (correspond) to the same RP. For example, if this value is 24, only the first 24 bits of the group addresses matter. The hash mask length allows one RP to be used for multiple groups. The default hash mask length is 0. • Use the optional *priority* argument (after you set the hash mask length) to specify the priority of the BSR as a C-RP. The priority range is from 0 to 255. The BSR C-RP with the highest priority (the lowest priority value) is preferred. If the priority values are the same, the router with the higher IP address is preferred. The default priority value is 0. **Note** The Cisco IOS and Cisco IOS XE implementation of PIM BSR uses the value 0 as the default priority for candidate RPs and BSRs. This implementation predates the draft-ietf-pim-sm-bsr IETF draft, the first IETF draft to specify 192 as the default priority value. The Cisco IOS and Cisco IOS XE implementation, thus, deviates from the IETF draft. To comply with the default priority value specified in the draft, you must explicitly set the priority value to 192.
Step 9	**ip pim rp-candidate** *interface-type interface-number* [group-list *access-list*] [**interval** seconds] [**priority** *value*] **Example:** `Router(config)# ip pim rp-candidate gigabitethernet 2/0/0 group-list 4 priority 192`	Configures the router to advertise itself as a PIM Version 2 candidate RP to the BSR. • Perform this step on the RP or on combined RP/BSR routers. **Note** BSR allows the RP function to run separately on one router and the BSR to run on one or multiple routers. It is possible to deploy the RP and the BSR on a combined RP/BSR router.

	Command or Action	Purpose
		• When an interval is specified, the candidate RP advertisement interval is set to the number of seconds specified. The default interval is 60 seconds. Tuning this interval down can reduce the time required to fail over to a secondary RP at the expense of generating more PIMv2 messages. • The Cisco IOS and Cisco IOS XE implementation of PIM BSR selects an RP from a set of candidate RPs using a method that is incompatible with the specification in RFC 2362. See the BSR and RFC 2362 Interoperable Candidate RP Example, on page 83 section for a configuration workaround. See CSCdy56806 using the Cisco Bug Toolkit for more information. **Note** The Cisco IOS and Cisco IOS XE implementation of PIM BSR uses the value 0 as the default priority for candidate RPs and BSRs. This implementation predates the draft-ietf-pim-sm-bsr IETF draft, the first IETF draft to specify 192 as the default priority value. The Cisco IOS and Cisco IOS XE implementation, thus, deviates from the IETF draft. To comply with the default priority value specified in the draft, you must explicitly set the priority value to 192.
Step 10	Repeat Steps 8 through 10 on all RP and BSR routers.	--
Step 11	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 12	**ip pim bsr-border** **Example:** `Router(config-if)# ip pim bsr-border`	Prevents the bootstrap router (BSR) messages from being sent or received through an interface. • See the BSR Border Interface, on page 57 section for more information.
Step 13	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 14	Repeat Steps 11 through 13 on all the routers that have boundary interfaces where the messages should not be sent or received.	--

	Command or Action	Purpose
Step 15	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `Router# show ip pim rp`	(Optional) Displays active rendezvous points (RPs) that are cached with associated multicast routing entries.
Step 16	**show ip pim rp-hash** [*group-address*] [*group-name*] **Example:** `Router# show ip pim rp-hash 239.1.1.1`	(Optional) Displays which rendezvous point (RP) is being selected for a specified group.
Step 17	**show ip pim bsr-router** **Example:** `Router# show ip pim bsr-router`	(Optional) Displays the bootstrap router (BSR) information.
Step 18	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**] **Example:** `Router# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the router and that were learned through IGMP. • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 19	**show ip mroute** **Example:** `Router# show ip mroute cbone-audio`	(Optional) Displays the contents of the IP mroute table.

What to Do Next

Proceed to the “ Verifying IP Multicast Operation ” module.

Configuring Sparse Mode with a Single Static RP(CLI)

A rendezvous point (RP) is required in networks running Protocol Independent Multicast sparse mode (PIM-SM). In PIM-SM, traffic will be forwarded only to network segments with active receivers that have explicitly requested multicast data.

This section describes how to configure sparse mode with a single static RP.

Before you begin

All access lists that are needed when sparse mode is configured with a single static RP should be configured prior to beginning the configuration task.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. Repeat Steps 1 through 5 on every interface that uses IP multicast.
7. **exit**
8. **ip pim rp-address** *rp-address* [*access-list*] [**override**]
9. **end**
10. **show ip pim rp** [**mapping**] [*rp-address*]
11. **show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]
12. **show ip mroute**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `device(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**interface** *type number* **Example:** `device(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode** **Example:** `device(config-if)# ip pim sparse-mode`	Enables PIM on an interface. You must use sparse mode.
Step 6	Repeat Steps 1 through 5 on every interface that uses IP multicast.	--
Step 7	**exit** **Example:** `device(config-if)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 8	**ip pim rp-address** *rp-address* [*access-list*] [**override**] **Example:** `device(config)# ip pim rp-address 192.168.0.0`	Configures the address of a PIM RP for a particular group. • The optional *access-list* argument is used to specify the number or name a standard access list that defines the multicast groups to be statically mapped to the RP. **Note** If no access list is defined, the RP will map to all multicast groups, 224/4. • The optional **override** keyword is used to specify that if dynamic and static group-to-RP mappings are used together and there is an RP address conflict, the RP address configured for a static group-to-RP mapping will take precedence. **Note** If the **override** keyword is not specified and there is RP address conflict, dynamic group-to-RP mappings will take precedence over static group-to-RP mappings.
Step 9	**end** **Example:** `device(config)# end`	Ends the current configuration session and returns to EXEC mode.
Step 10	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `device# show ip pim rp mapping`	(Optional) Displays RPs known in the network and shows how the router learned about each RP.
Step 11	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**] **Example:** `device# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the router and that were learned through IGMP. • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 12	**show ip mroute** **Example:** `device# show ip mroute`	(Optional) Displays the contents of the IP mroute table.

What to Do Next

Proceed to the “ Verifying IP Multicast Operation ” module.

Configuring Source Specific Multicast

Before you begin

If you want to use an access list to define the Source Specific Multicast (SSM) range, configure the access list before you reference the access list in the **ip pim ssm** command.

SUMMARY STEPS

1. **configure terminal**
2. **ip multicast-routing** [**distributed**]
3. **ip pim ssm** {**default** | **range** *access-list*}
4. **interface** *type number*
5. **ip pim sparse-mode**
6. Repeat Steps 1 through 6 on every interface that uses IP multicast.
7. **ip igmp version 3**
8. Repeat Step 8 on all host-facing interfaces.
9. **end**
10. **show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]
11. **show ip mroute**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `device# configure terminal`	Enters global configuration mode.
Step 2	**ip multicast-routing** [**distributed**] **Example:** `device(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 3	**ip pim ssm** {**default** | **range** *access-list*} **Example:** `device(config)# ip pim ssm default`	Configures SSM service. • The **default** keyword defines the SSM range access list as 232/8. • The **range** keyword specifies the standard IP access list number or name that defines the SSM range.
Step 4	**interface** *type number* **Example:** `device(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.

	Command or Action	Purpose
Step 5	**ip pim sparse-mode** **Example:** `device(config-if)# ip pim sparse-mode`	Enables PIM on an interface. You must use sparse mode.
Step 6	Repeat Steps 1 through 6 on every interface that uses IP multicast.	--
Step 7	**ip igmp version 3** **Example:** `device(config-if)# ip igmp version 3`	Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2. Version 3 is required by SSM.
Step 8	Repeat Step 8 on all host-facing interfaces.	--
Step 9	**end** **Example:** `device(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 10	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**] **Example:** `device# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the device and that were learned through IGMP. • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 11	**show ip mroute** **Example:** `device# show ip mroute`	(Optional) Displays the contents of the IP mroute table. • This command displays whether a multicast group is configured for SSM service or a source-specific host report has been received.

What to Do Next

Proceed to the “ Verifying IP Multicast Operation ” module.

Configuring Bidirectional PIM

Before you begin

All required access lists must be configured before configuring bidirectional PIM.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. **exit**
7. **ip pim bidir-enable**
8. **ip pim rp-address** *rp-address* [*access-list*] [**override**] **bidir**
9. **end**
10. Repeat Steps 2 through 9 on every multicast-enabled interface on every router.
11. **show ip pim rp** [**mapping**] [*rp-address*]
12. **show ip mroute**
13. **show ip pim interface** [*type number*] [**df** | **count**] [*rp-address*]
14. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:**	Enables IP multicast routing. • Use the **distributed** keyword to enable Multicast Distributed Switching.
Step 4	**interface** *type number*	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode**	Enables sparse mode.
Step 6	**exit**	Returns to global configuration mode.
Step 7	**ip pim bidir-enable**	Enables bidir-PIM on a router. • Perform this step on every router.
Step 8	**ip pim rp-address** *rp-address* [*access-list*] [**override**] **bidir**	Configures the address of a PIM RP for a particular group. • Perform this step on every router. • This command defines the RP as bidirectional and defines the bidirectional group by way of the access list. • The optional **override** keyword is used to specify that if dynamic and static group-to-RP mappings are used together and there is an RP address conflict, the RP address configured for a static group-to-RP mapping will take precedence.

	Command or Action	Purpose
		Note If the **override** keyword is not specified and there is RP address conflict, dynamic group-to-RP mappings will take precedence over static group-to-RP mappings.
Step 9	**end**	Exits interface configuration mode and returns to privileged EXEC mode.
Step 10	Repeat Steps 2 through 9 on every multicast-enabled interface on every router.	--
Step 11	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `Device# show ip pim rp`	(Optional) Displays active RPs that are cached with associated multicast routing entries.
Step 12	**show ip mroute**	(Optional) Displays the contents of the IP mroute table.
Step 13	**show ip pim interface** [*type number*] [**df** | **count**] [*rp-address*] **Example:** `Device# show ip pim interface`	(Optional) Displays information about the elected DF for each RP of an interface, along with the unicast routing metric associated with the DF.
Step 14	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuration Examples for Basic IP Multicast

Example: Sparse Mode with Auto-RP

The following example configures sparse mode with Auto-RP:

```
ip multicast-routing
ip pim autorp listener
ip pim send-rp-announce Loopback0 scope 16 group-list 1
ip pim send-rp-discovery Loopback1 scope 16
no ip pim dm-fallback
access-list 1 permit 239.254.2.0 0.0.0.255
access-list 1 permit 239.254.3.0 0.0.0.255
.
.
.
access-list 10 permit 224.0.1.39
access-list 10 permit 224.0.1.40
access-list 10 permit 239.254.2.0 0.0.0.255
access-list 10 permit 239.254.3.0 0.0.0.255
```

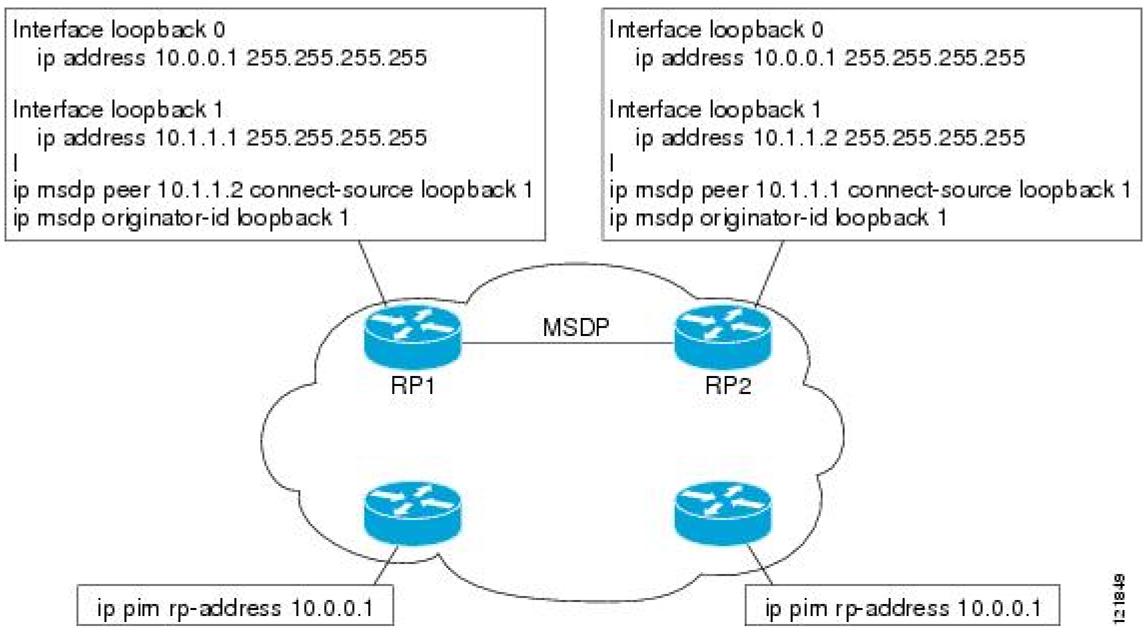
Sparse Mode with Anycast RP Example

The main purpose of an Anycast RP implementation is that the downstream multicast routers will have just one address for an RP. The example given in the figure below shows how loopback interface 0 of the RPs (RP1 and RP2) is configured with the 10.0.0.1 IP address. If this 10.0.0.1 address is configured on all RPs as the address for loopback interface 0 and then configured as the RP address, IP routing will converge on the closest RP. This address must be a host route; note the 255.255.255.255 subnet mask.

The downstream routers must be informed about the 10.0.0.1 RP address. In the figure below, the routers are configured statically with the **ip pim rp-address 10.0.0.1**global configuration command. This configuration could also be accomplished using the Auto-RP or bootstrap router (BSR) features.

The RPs in the figure must also share source information using MSDP. In this example, loopback interface 1 of the RPs (RP1 and RP2) is configured for MSDP peering. The MSDP peering address must be different from the anycast RP address.

Figure 13: AnyCast RP Configuration



Many routing protocols choose the highest IP address on loopback interfaces for the router ID. A problem may arise if the router selects the anycast RP address for the router ID. It is recommended that you avoid this problem by manually setting the router ID on the RPs to the same address as the MSDP peering address (for example, the loopback 1 address in the figure above). In Open Shortest Path First (OSPF), the router ID is configured using the **router-id**router configuration command. In Border Gateway Protocol (BGP), the router ID is configured using the **bgp router-id** router configuration command. In many BGP topologies, the MSDP peering address and the BGP peering address must be the same in order to pass the RPF check. The BGP peering address can be set using the **neighbor update-source** router configuration command.

The anycast RP example above uses IP addresses taken from RFC 1918. These IP addresses are normally blocked at interdomain borders and therefore are not accessible to other ISPs. You must use valid IP addresses if you want the RPs to be reachable from other domains.

The following example shows how to perform an Anycast RP configuration.

On RP 1

```
ip pim rp-address 10.0.0.1
interface loopback 0
 ip address 10.0.0.1 255.255.255.255
!
interface loopback 1
 ip address 10.1.1.1. 255.255.255.255
!
 ip msdp peer 10.1.1.2 connect-source loopback 1
 ip msdp originator-id loopback 1
```

On RP 2

```
ip pim rp-address 10.0.0.1
interface loopback 0
 ip address 10.0.0.1 255.255.255.255
interface loopback 1
 ip address 10.1.1.2. 255.255.255.255
!
 ip msdp peer 10.1.1.1 connect-source loopback 1
 ip msdp originator-id loopback 1
```

All Other Routers

```
ip pim rp-address 10.0.0.1
```

Sparse Mode with Bootstrap Router Example

The following example is a configuration for a candidate BSR, which also happens to be a candidate RP:

```
!
ip multicast-routing
!
interface GigabitEthernet0/0/0
 ip address 172.69.62.35 255.255.255.240
 ip pim sparse-mode
!
interface GigabitEthernet1/0/0
 ip address 172.21.24.18 255.255.255.248
 ip pim sparse-mode
!
interface GigabitEthernet2/0/0
 ip address 172.21.24.12 255.255.255.248
 ip pim sparse-mode
!
ip pim bsr-candidate GigabitEthernet2/0/0 30 10
ip pim rp-candidate GigabitEthernet2/0/0 group-list 5
access-list 5 permit 239.255.2.0 0.0.0.255
```

BSR and RFC 2362 Interoperable Candidate RP Example

When Cisco and non-Cisco routers are being operated in a single PIM domain with PIM Version 2 BSR, care must be taken when configuring candidate RPs because the Cisco implementation of the BSR RP selection is not fully compatible with RFC 2362.

RFC 2362 specifies that the BSR RP be selected as follows (RFC 2362, 3.7):

1. Select the candidate RP with the highest priority (lowest configured priority value).
2. If there is a tie in the priority level, select the candidate RP with the highest hash function value.
3. If there is a tie in the hash function value, select the candidate RP with the highest IP address.

Cisco routers always select the candidate RP based on the longest match on the announced group address prefix before selecting an RP based on priority, hash function, or IP address.

Inconsistent candidate RP selection between Cisco and non-Cisco RFC 2362-compliant routers in the same domain if multiple candidate RPs with partially overlapping group address ranges are configured can occur. Inconsistent candidate RP selection can prevent connectivity between sources and receivers in the PIM domain. A source may register with one candidate RP and a receiver may connect to a different candidate RP even though it is in the same group.

The following example shows a configuration that can cause inconsistent RP selection between a Cisco and a non-Cisco router in a single PIM domain with PIM Version 2 BSR:

```
access-list 10 permit 224.0.0.0  7.255.255.255
ip pim rp-candidate gigabitethernet1/0/0 group-list 10 priority 20
access-list 20 permit 224.0.0.0 15.255.255.255
ip pim rp-candidate gigabitethernet2/0/0 group-list 20 priority 10
```

In this example, a candidate RP on GigabitEthernet interface 1/0/0 announces a longer group prefix of 224.0.0.0/5 with a lower priority of 20. The candidate RP on GigabitEthernet interface 2/0/0 announces a shorter group prefix of 224.0.0.0/4 with a higher priority of 10. For all groups that match both ranges a Cisco router will always select the candidate RP on Ethernet interface 1 because it has the longer announced group prefix. A non-Cisco fully RFC 2362-compliant router will always select the candidate RP on GigabitEthernet interface 2/0/0 because it is configured with a higher priority.

To avoid this interoperability issue, do not configure different candidate RPs to announce partially overlapping group address prefixes. Configure any group prefixes that you want to announce from more than one candidate RP with the same group prefix length.

The following example shows how to configure the previous example so that there is no incompatibility between a Cisco router and a non-Cisco router in a single PIM domain with PIM Version 2 BSR:

```
access-list 10 permit 224.0.0.0  7.255.255.255
ip pim rp-candidate gigabitethernet1/0/0 group-list 10 priority 20
access-list 20 permit 224.0.0.0  7.255.255.255
access-list 20 permit 232.0.0.0  7.255.255.255
ip pim rp-candidate gigabitethernet2/0/0 group-list 20 priority 10
```

In this configuration the candidate RP on Ethernet interface 2 announces group address 224.0.0.0/5 and 232.0.0.0/5 which equal 224.0.0.0/4, but gives the interface the same group prefix length (5) as the candidate RP on Ethernet 1. As a result, both a Cisco router and an RFC 2362-compliant router will select the RP Ethernet interface 2.

Example: Sparse Mode with a Single Static RP

The following example sets the PIM RP address to 192.168.1.1 for all multicast groups and defines all groups to operate in sparse mode:

```
ip multicast-routing
interface gigiabitethernet 1/0/0
```

```
 ip pim sparse-mode
ip pim rp-address 192.168.1.1
```

Note The same RP cannot be used for both bidirectional and sparse mode groups.

The following example sets the PIM RP address to 172.16.1.1 for the multicast group 225.2.2.2 only:

```
access list 1 225.2.2.2 0.0.0.0
 ip pim rp-address 172.17.1.1
```

SSM with IGMPv3 Example

The following example shows how to configure a device (running IGMPv3) for SSM:

```
ip multicast-routing
!
interface GigabitEthernet3/1/0
 ip address 172.21.200.203 255.255.255.0
 description backbone interface
 ip pim sparse-mode
!
interface GigabitEthernet3/2/0
 ip address 131.108.1.2 255.255.255.0
 ip pim sparse-mode
 description ethernet connected to hosts
 ip igmp version 3
!
ip pim ssm default
```

SSM Filtering Example

The following example shows how to configure filtering on legacy RP routers running software releases that do not support SSM routing. This filtering will suppress all unwanted PIM-SM and MSDP traffic in the SSM range. Without this filtering, SSM will still operate, but there may be additional RPT traffic if legacy first hop and last hop routers exist in the network.

```
ip access-list extended  no-ssm-range
  deny   ip any 232.0.0.0 0.255.255.255 ! SSM range
  permit ip any any
! Deny registering in SSM range
ip pim accept-register list no-ssm-range
ip access-list extended msdp-nono-list
  deny   ip any 232.0.0.0 0.255.255.255 ! SSM Range
  ! .
  ! .
  ! .
  ! See ftp://ftpeng.cisco.com/ipmulticast/config-notes/msdp-sa-filter.txt for other SA
  ! messages that typically need to be filtered.
  permit ip any any
! Filter generated SA messages in SSM range. This configuration is only needed if there
! are directly connected sources to this router. The “ip pim accept-register” command
! filters remote sources.
ip msdp redistribute list msdp-nono-list
! Filter received SA messages in SSM range. “Filtered on receipt” means messages are
! neither processed or forwarded. Needs to be configured for each MSDP peer.
ip msdp sa-filter in msdp-peer1 list msdp-nono-list
```

```
! .
! .
! .
ip msdp sa-filter in msdp-peerN list msdp-nono-list
```

Bidir-PIM Example

By default, a bidirectional RP advertises all groups as bidirectional. An access list on the RP can be used to specify a list of groups to be advertised as bidirectional. Groups with the **deny** keyword will operate in dense mode. A different, nonbidirectional RP address is required for groups that operate in sparse mode because a single access list only allows either a **permit** or **deny** keyword.

The following example shows how to configure an RP for both sparse mode and bidirectional mode groups. The groups identified as 224/8 and 227/8 are bidirectional groups, and 226/8 is a sparse mode group. The RP must be configured to use different IP addresses for the sparse mode and bidirectional mode operations. Two loopback interfaces are used to allow this configuration. The addresses of these loopback interfaces must be routed throughout the PIM domain in such a way that the other routers in the PIM domain can communicate with the RP.

```
ip multicast-routing
!
.
.
.
!
interface loopback 0
 description One loopback address for this router's Bidir Mode RP function
 ip address 10.0.1.1 255.255.255.0
!
interface loopback 1
description One loopback address for this router's Sparse Mode RP function
 ip address 10.0.2.1 255.255.255.0
!
.
.
.
!
ip pim bidir-enable
ip pim rp-address 10.0.1.1 45 bidir
ip pim rp-address 10.0.2.1 46
!
access-list 45 permit 224.0.0.0 0.255.255.255
access-list 45 permit 227.0.0.0 0.255.255.255
access-list 46 permit 226.0.0.0 0.255.255.255
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference

Standards and RFCs

Standard/RFC	Title
draft-kouvelas-pim-bidir-new-00.txt	A New Proposal for Bi-directional PIM
RFC 1112	Host Extensions for IP Multicasting
RFC 1918	Address Allocation for Private Internets
RFC 2770	GLOP Addressing in 233/8
RFC 3569	An Overview of Source-Specific Multicast (SSM)

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Basic IP Multicast in IPv4 Networks

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 4

Using MSDP to Interconnect Multiple PIM-SM Domains

This module describes the tasks associated with using Multicast Source Discovery Protocol (MSDP) to interconnect multiple PIM-SM domains. The tasks explain how to configure MSDP peers, mesh groups, and default peers, how to use filters to control and scope MSDP activity, and how to monitor and maintain MSDP. Using MSDP with PIM-SM greatly reduces the complexity of connecting multiple PIM-SM domains.

Information About Using MSDP to Interconnect Multiple PIM-SM Domains

Benefits of Using MSDP to Interconnect Multiple PIM-SM Domains

- Allows a rendezvous point (RP) to dynamically discover active sources outside of its domain.
- Introduces a more manageable approach for building multicast distribution trees between multiple domains.

Use of MSDP to Interconnect Multiple PIM-SM Domains

MSDP is a mechanism to connect multiple PIM-SM domains. The purpose of MSDP is to discover multicast sources in other PIM domains. The main advantage of MSDP is that it reduces the complexity of interconnecting multiple PIM-SM domains by allowing PIM-SM domains to use an interdomain source tree (rather than a common shared tree). When MSDP is configured in a network, RPs exchange source information with RPs in other domains. An RP can join the interdomain source tree for sources that are sending to groups for which it has receivers. The RP can do that because it is the root of the shared tree within its domain, which has branches to all points in the domain where there are active receivers. When a last-hop router learns of a new source outside the PIM-SM domain (through the arrival of a multicast packet from the source down the shared tree), it then can send a join toward the source and join the interdomain source tree.

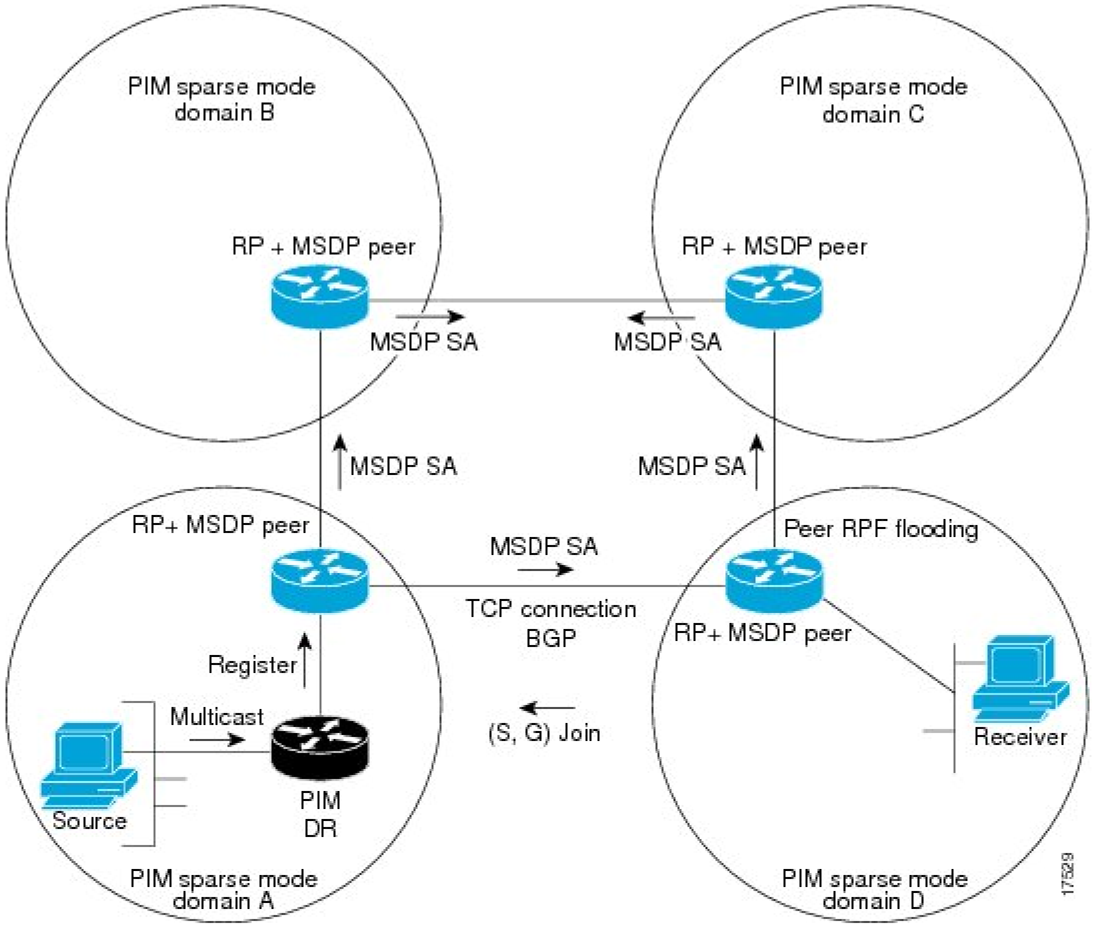
Note If the RP either has no shared tree for a particular group or a shared tree whose outgoing interface list is null, it does not send a join to the source in another domain.

When MSDP is enabled, an RP in a PIM-SM domain maintains MSDP peering relationships with MSDP-enabled routers in other domains. This peering relationship occurs over a TCP connection, where primarily a list of sources sending to multicast groups is exchanged. MSDP uses TCP (port 639) for its peering connections. As with BGP, using point-to-point TCP peering means that each peer must be explicitly configured. The TCP connections between RPs, moreover, are achieved by the underlying routing system. The receiving RP uses the source lists to establish a source path. If the multicast sources are of interest to a domain that has receivers, multicast data is delivered over the normal, source-tree building mechanism provided by PIM-SM. MSDP is also used to announce sources sending to a group. These announcements must originate at the RP of the domain.

Note MSDP depends on BGP or multiprotocol BGP (MBGP) for interdomain operation. We recommended that you run MSDP on RPs sending to global multicast groups.

The figure illustrates MSDP operating between two MSDP peers. PIM uses MSDP as the standard mechanism to register a source with the RP of a domain.

Figure 14: MSDP Running Between RP Peers



When MSDP is implemented, the following sequence of events occurs:

1. When a PIM designated router (DR) registers a source with its RP as illustrated in the figure, the RP sends a Source-Active (SA) message to all of its MSDP peers.

Note The DR sends the encapsulated data to the RP only once per source (when the source goes active). If the source times out, this process happens again when it goes active again. This situation is different from the periodic SA message that contains all sources that are registered to the originating RP. Those SA messages are MSDP control packets, and, thus, do not contain encapsulated data from active sources.

1. The SA message identifies the source address, the group that the source is sending to, and the address or the originator ID of the RP, if configured.
2. Each MSDP peer that receives the SA message floods the SA message to all of its peers downstream from the originator. In some cases (such as the case with the RPs in PIM-SM domains B and C in the figure), an RP may receive a copy of an SA message from more than one MSDP peer. To prevent looping, the RP consults the BGP next-hop database to determine the next hop toward the originator of the SA message. If both MBGP and unicast BGP are configured, MBGP is checked first, and then unicast BGP. That next-hop neighbor is the RPF-peer for the originator. SA messages that are received from the originator

on any interface other than the interface to the RPF peer are dropped. The SA message flooding process, therefore, is referred to as peer-RPF flooding. Because of the peer-RPF flooding mechanism, BGP or MBGP must be running in conjunction with MSDP.

Note (M)BGP is not required in MSDP mesh group scenarios. For more information about MSDP mesh groups, see the Configuring an MSDP Mesh Group, on page 112 section.

Note (M)BGP is not required in default MSDP peer scenarios or in scenarios where only one MSDP peer is configured. For more information, see the Configuring a Default MSDP Peer, on page 111 section.

1. When an RP receives an SA message, it checks to see whether there are any members of the advertised groups in its domain by checking to see whether there are interfaces on the group's (*, G) outgoing interface list. If there are no group members, the RP does nothing. If there are group members, the RP sends an (S, G) join toward the source. As a result, a branch of the interdomain source tree is constructed across autonomous system boundaries to the RP. As multicast packets arrive at the RP, they are then forwarded down its own shared tree to the group members in the RP's domain. The members' DRs then have the option of joining the rendezvous point tree (RPT) to the source using standard PIM-SM procedures.
2. The originating RP continues to send periodic SA messages for the (S, G) state every 60 seconds for as long as the source is sending packets to the group. When an RP receives an SA message, it caches the SA message. Suppose, for example, that an RP receives an SA message for (172.16.5.4, 228.1.2.3) from originating RP 10.5.4.3. The RP consults its mroute table and finds that there are no active members for group 228.1.2.3, so it passes the SA message to its peers downstream of 10.5.4.3. If a host in the domain then sends a join to the RP for group 228.1.2.3, the RP adds the interface toward the host to the outgoing interface list of its (*, 224.1.2.3) entry. Because the RP caches SA messages, the router will have an entry for (172.16.5.4, 228.1.2.3) and can join the source tree as soon as a host requests a join.

Note In all current and supported software releases, caching of MSDP SA messages is mandatory and cannot be manually enabled or disabled. By default, when an MSDP peer is configured, the **ip multicast cache-sa-state** command will automatically be added to the running configuration.

MSDP Message Types

There are four basic MSDP message types, each encoded in their own Type, Length, and Value (TLV) data format.

SA Messages

SA messages are used to advertise active sources in a domain. In addition, these SA messages may contain the initial multicast data packet that was sent by the source.

SA messages contain the IP address of the originating RP and one or more (S, G) pairs being advertised. In addition, the SA message may contain an encapsulated data packet.

Keepalive Messages

Keepalive messages are sent every 60 seconds in order to keep the MSDP session active. If no keepalive messages or SA messages are received for 75 seconds, the MSDP session is reset.

SA Message Origination Receipt and Processing

The section describes SA message origination, receipt, and processing in detail.

SA Message Origination

SA messages are triggered by an RP (assuming MSDP is configured) when any new source goes active within a local PIM-SM domain. A local source is a source that is directly connected to the RP or is the first-hop DR that has registered with it. An RP originates SA messages only for local sources in its PIM-SM domain; that is, for local sources that register with it.

Note A local source is denoted by the A flag being set in the (S, G) mroute entry on the RP (which can be viewed in the output of the **show ip mroute** command). This flag indicates that the source is a candidate for advertisement by the RP to other MSDP peers.

When a source is in the local PIM-SM domain, it causes the creation of (S, G) state in the RP. New sources are detected by the RP either by the receipt of a register message or the arrival of the first (S, G) packet from a directly connected source. The initial multicast packet sent by the source (either encapsulated in the register message or received from a directly connected source) is encapsulated in the initial SA message.

SA Message Receipt

SA messages are only accepted from the MSDP RPF peer that is in the best path back toward the originator. The same SA message arriving from other MSDP peers must be ignored or SA loops can occur. Deterministically selecting the MSDP RPF peer for an arriving SA message requires knowledge of the MSDP topology. However, MSDP does not distribute topology information in the form of routing updates. MSDP infers this information by using (M)BGP routing data as the best approximation of the MSDP topology for the SA RPF check mechanism. An MSDP topology, therefore, must follow the same general topology as the BGP peer topology. Besides a few exceptions (such as default MSDP peers and MSDP peers in MSDP mesh groups), MSDP peers, in general should also be (M)BGP peers.

How RPF Check Rules Are Applied to SA Messages

The rules that apply to RPF checks for SA messages are dependent on the BGP peerings between the MSDP peers:

- Rule 1: Applied when the sending MSDP peer is also an interior (M)BGP peer.
- Rule 2: Applied when the sending MSDP peer is also an exterior (M)BGP peer.
- Rule 3: Applied when the sending MSDP peer is not an (M)BGP peer.

RPF checks are not performed in the following cases:

- If the sending MSDP peer is the only MSDP peer, which would be the case if only a single MSDP peer or a default MSDP peer is configured.

- If the sending MSDP peer is a member of a mesh group.
- If the sending MSDP peer address is the RP address contained in the SA message.

How the Software Determines the Rule to Apply to RPF Checks

The software uses the following logic to determine which RPF rule to apply to RPF checks:

- Find the (M)BGP neighbor that has the same IP address as the sending MSDP peer.
  - If the matching (M)BGP neighbor is an internal BGP (iBGP) peer, apply Rule 1.
  - If the matching (M)BGP neighbor is an external BGP (eBGP) peer, apply Rule 2.
  - If no match is found, apply Rule 3.

The implication of the RPF check rule selection is as follows: The IP address used to configure an MSDP peer on a device must match the IP address used to configure the (M)BGP peer on the same device.

Rule 1 of RPF Checking of SA Messages in MSDP

Rule 1 of RPF checking in MSDP is applied when the sending MSDP peer is also an i(M)BGP peer. When Rule 1 is applied, the RPF check proceeds as follows:

1. The peer searches the BGP Multicast Routing Information Base (MRIB) for the best path to the RP that originated the SA message. If a path is not found in the MRIB, the peer then searches the Unicast Routing Information Base (URIB). If a path is still not found, the RPF check fails.
2. If the previous search succeeds (that is, the best path is found), the peer then determines the address of the BGP neighbor for this best path, which will be the address of the BGP neighbor that sent the peer the path in BGP update messages.

Note The BGP neighbor address is not the same as the next-hop address in the path. Because i(M)BGP peers do not update the next-hop attribute of a path, the next-hop address usually is not the same as the address of the BGP peer that sent us the path.

Note The BGP neighbor address is not necessarily the same as the BGP ID of the peer that sent the peer the path.

1. If the IP address of the sending MSDP peer is the same as the BGP neighbor address (that is, the address of the BGP peer that sent the peer the path), then the RPF check succeeds; otherwise it fails.

Implications of Rule 1 of RPF Checking on MSDP

The MSDP topology must mirror the (M)BGP topology. In general, wherever there is an i(M)BGP peer connection between two devices, an MSDP peer connection should be configured. More specifically, the IP address of the far-end MSDP peer connection must be the same as the far-end i(M)BGP peer connection. The addresses must be the same because the BGP topology between i(M)BGP peers inside an autonomous system is not described by the AS path. If it were always the case that i(M)BGP peers updated the next-hop address in the path when sending an update to another i(M)BGP peer, then the peer could rely on the next-hop address to describe the i(M)BGP topology (and hence the MSDP topology). However, because the default behavior for i(M)BGP peers is to not update the next-hop address, the peer cannot rely on the next-hop address to describe the (M)BGP topology (MSDP topology). Instead, the i(M)BGP peer uses the address of the i(M)BGP peer that sent the path to describe the i(M)BGP topology (MSDP topology) inside the autonomous system.

Tip Care should be taken when configuring the MSDP peer addresses to make sure that the same address is used for both i(M)BGP and MSDP peer addresses.

Rule 2 of RPF Checking of SA Messages in MSDP

Rule 2 of RPF checking in MSDP is applied when the sending MSDP peer is also an e(M)BGP peer. When Rule 2 is applied, the RPF check proceeds as follows:

1. The peer searches the BGP MRIB for the best path to the RP that originated the SA message. If a path is not found in the MRIB, the peer then searches the URIB. If a path is still not found, the RPF check fails.
2. If the previous search succeeds (that is, the best path is found), the peer then examines the path. If the first autonomous system in the best path to the RP is the same as the autonomous system of the e(M)BGP peer (which is also the sending MSDP peer), then the RPF check succeeds; otherwise it fails.

Implications of Rule 2 of RPF Checking on MSDP

The MSDP topology must mirror the (M)BGP topology. In general, wherever there is an e(M)BGP peer connection between two devices, an MSDP peer connection should be configured. As opposed to Rule 1, the IP address of the far-end MSDP peer connection does not have to be the same as the far-end e(M)BGP peer connection.The reason that the addresses do not have to be identical is that BGP topology between two e(M)BGP peers is not described by the AS path.

Rule 3 of RPF Checking of SA Messages in MSDP

Rule 3 of RPF checking is applied when the sending MSDP peer is not a (M)BGP peer at all. When Rule 3 is applied, the RPF check proceeds as follows:

1. The peer searches the BGP MRIB for the best path to the RP that originated the SA message. If a path is not found in the MRIB, the peer then searches the URIB. If a path is still not found, the RPF check fails.
2. If the previous search succeeds (that is, the best path to the RP that originated the SA message is found), the peer then searches the BGP MRIB for the best path to the MSDP peer that sent the SA message. If a path is not found in the MRIB, the peer then searches the URIB. If a path is still not found, the RPF check fails.

Note The autonomous system of the MSDP peer that sent the SA is the origin autonomous system, which is the last autonomous system in the AS path to the MSDP peer.

1. If the first autonomous system in the best path to the RP is the same as the autonomous system of the sending MSDP peer, then the RPF check succeeds; otherwise it fails.

SA Message Processing

The following steps are taken by an MSDP peer whenever it processes an SA message:

1. Using the group address G of the (S, G) pair in the SA message, the peer locates the associated (*, G) entry in the mroute table. If the (*, G) entry is found and its outgoing interface list is not null, then there are active receivers in the PIM-SM domain for the source advertised in the SA message.
2. The MSDP peer then creates an (S, G) entry for the advertised source.
3. If the (S, G) entry did not already exist, the MSDP peer immediately triggers an (S, G) join toward the source in order to join the source tree.
4. The peer then floods the SA message to all other MSDP peers with the exception of:
    - The MSDP peer from which the SA message was received.
    - Any MSDP peers that are in the same MSDP mesh group as this device (if the peer is a member of a mesh group).

Note SA messages are stored locally in the device's SA cache.

MSDP Peers

Like BGP, MSDP establishes neighbor relationships with other MSDP peers. MSDP peers connect using TCP port 639. The lower IP address peer takes the active role of opening the TCP connection. The higher IP address peer waits in LISTEN state for the other to make the connection. MSDP peers send keepalive messages every 60 seconds. The arrival of data performs the same function as the keepalive message and keeps the session from timing out. If no keepalive messages or data is received for 75 seconds, the TCP connection is reset.

MSDP MD5 Password Authentication

The MSDP MD5 password authentication feature is an enhancement to support Message Digest 5 (MD5) signature protection on a TCP connection between two MSDP peers. This feature provides added security by protecting MSDP against the threat of spoofed TCP segments being introduced into the TCP connection stream.

How MSDP MD5 Password Authentication Works

Developed in accordance with RFC 2385, the MSDP MD5 password authentication feature is used to verify each segment sent on the TCP connection between MSDP peers. The **ip msdp password peer** command is used to enable MD5 authentication for TCP connections between two MSDP peers. When MD5 authentication is enabled between two MSDP peers, each segment sent on the TCP connection between the peers is verified. MD5 authentication must be configured with the same password on both MSDP peers; otherwise, the connection between them will not be made. Configuring MD5 authentication causes the Cisco IOS software to generate and verify the MD5 digest of every segment sent on the TCP connection.

Benefits of MSDP MD5 Password Authentication

- Protects MSDP against the threat of spoofed TCP segments being introduced into the TCP connection stream.
- Uses the industry-standard MD5 algorithm for improved reliability and security.

SA Message Limits

The **ip msdp sa-limit** command is used to limit the overall number of SA messages that a device can accept from specified MSDP peers. When the **ip msdp sa-limit** command is configured, the device maintains a per-peer count of SA messages stored in the SA cache and will ignore new messages from a peer if the configured SA message limit for that peer has been reached.

The **ip msdp sa-limit** command was introduced as a means to protect an MSDP-enabled device from denial of service (DoS) attacks. We recommended that you configure SA message limits for all MSDP peerings on the device. An appropriately low SA limit should be configured on peerings with a stub MSDP region (for example, a peer that may have some further downstream peers but that will not act as a transit for SA messages across the rest of the Internet). A high SA limit should be configured for all MSDP peerings that act as transits for SA messages across the Internet.

MSDP Keepalive and Hold-Time Intervals

The **ip msdp keepalive** command is used to adjust the interval at which an MSDP peer will send keepalive messages and the interval at which the MSDP peer will wait for keepalive messages from other peers before declaring them down.

Once an MSDP peering session is established, each side of the connection sends a keepalive message and sets a keepalive timer. If the keepalive timer expires, the local MSDP peer sends a keepalive message and restarts its keepalive timer; this interval is referred to as the keepalive interval. The *keepalive-interval*argument is used to adjust the interval for which keepalive messages will be sent. The keepalive timer is set to the value specified for the *keepalive-interval*argument when the peer comes up. The keepalive timer is reset to the value of the *keepalive-interval* argument whenever an MSDP keepalive message is sent to the peer and reset when the timer expires. The keepalive timer is deleted when an MSDP peering session is closed. By default, the keepalive timer is set to 60 seconds.

Note The value specified for the *keepalive-interval* argument must be less than the value specified for the *holdtime-interval*argument and must be at least one second.

The hold-time timer is initialized to the value of the *hold-time-interval* argument whenever an MSDP peering connection is established, and is reset to the value of the *hold-time-interval*argument whenever an MSDP keepalive message is received. The hold-time timer is deleted whenever an MSDP peering connection is closed. By default, the hold-time interval is set to 75 seconds.

Use the *hold-time-interval* argument to adjust the interval at which the MSDP peer will wait for keepalive messages from other peers before declaring them down.

MSDP Connection-Retry Interval

You can adjust the interval at which all MSDP peers will wait after peering sessions are reset before attempting to reestablish the peering sessions. This interval is referred to as the connection-retry interval. By default, MSDP peers will wait 30 seconds after the session is reset before attempting to reestablish sessions with other peers. The modified configured connection-retry interval applies to all MSDP peering sessions on the device.

MSDP Compliance with IETF RFC 3618

When the MSDP Compliance with IETF RFC 3618 feature is configured, the peer-RPF forwarding rules defined in IETF RFC 3618 are applied to MSDP peers. IETF RFC 3618 provides peer-RPF forwarding rules that are used for forwarding SA messages throughout an MSDP-enabled internet. Unlike the RPF check used when forwarding data packets, which compares a packet's source address against the interface upon which the packet was received, the peer-RPF check compares the RP address carried in the SA message against the MSDP peer from which the message was received. Except when MSDP mesh groups are being used, SA messages from an RP address are accepted from only one MSDP peer to avoid looping SA messages.

Note For more information about the MSDP peer-forwarding rules defined in RFC 3618, see RFC 3618, Multicast Source Discovery Protocol (MSDP) .

Benefits of MSDP Compliance with RFC 3618

- You can use BGP route reflectors (RRs) without running MSDP on them. This capability is useful to service providers that need to reduce the load on RRs.
- You can use an Interior Gateway Protocol (IGP) for the Reverse Path Forwarding (RPF) checks and thereby run peerings without (M)BGP. This capability is useful to enterprise customers that do not run (M)BGP and require larger topologies than mesh groups can provide.

Note IGP peerings must always be between directly connected MSDP peers or else the RPF checks will fail.

- You can have peerings between routers in nondirectly connected autonomous systems (that is, with one or more autonomous systems between them). This capability helps in confederation configurations and for redundancy.

Default MSDP Peers

In most scenarios, an MSDP peer is also a BGP peer. If an autonomous system is a stub or nontransit autonomous system, and particularly if the autonomous system is not multihomed, there is little or no reason to run BGP to its transit autonomous system. A static default route at the stub autonomous system, and a static route pointing to the stub prefixes at the transit autonomous system, is generally sufficient. But if the stub autonomous system is also a multicast domain and its RP must peer with an RP in the neighboring domain, MSDP depends on the BGP next-hop database for its peer-RPF checks. You can disable this dependency on BGP by defining a default peer from which to accept all SA messages without performing the peer-RPF check, using the **ip msdp default-peer** command. A default MSDP peer must be a previously configured MSDP peer.

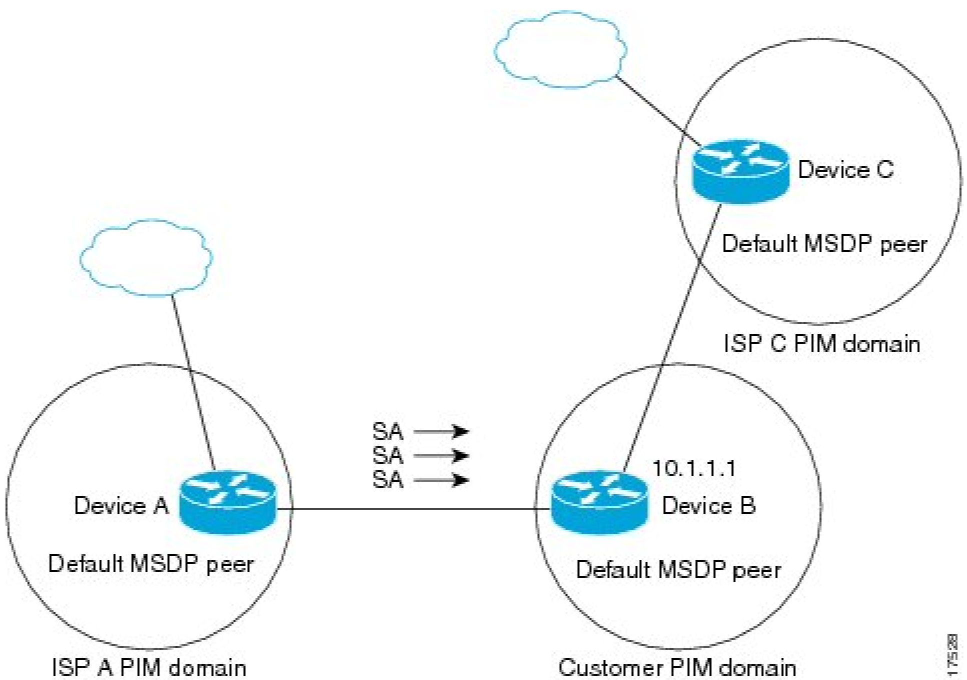
A stub autonomous system also might want to have MSDP peerings with more than one RP for the sake of redundancy. For example, SA messages cannot just be accepted from multiple default peers, because there is no RPF check mechanism. Instead, SA messages are accepted from only one peer. If that peer fails, SA messages are then accepted from the other peer. The underlying assumption here, of course, is that both default peers are sending the same SA messages.

The figure illustrates a scenario where default MSDP peers might be used. In the figure, a customer that owns Router B is connected to the Internet through two Internet service providers (ISPs), one that owns Router A and the other that owns Router C. They are not running BGP or MBGP between them. In order for the customer to learn about sources in the ISP domain or in other domains, Router B identifies Router A as its default MSDP peer. Router B advertises SA messages to both Router A and Router C, but accepts SA messages either from Router A only or Router C only. If Router A is the first default peer in the configuration, it will be used if it is up and running. Only if Router A is not running will Router B accept SA messages from Router C.

The ISP will also likely use a prefix list to define which prefixes it will accept from the customer router. The customer will define multiple default peers, each having one or more prefixes associated with it.

The customer has two ISPs to use. The customer defines both ISPs as default peers. As long as the first default peer identified in the configuration is up and running, it will be the default peer and the customer will accept all SA messages it receives from that peer.

Figure 15: Default MSDP Peer Scenario



Router B advertises SAs to Router A and Router C, but uses only Router A or Router C to accept SA messages. If Router A is first in the configuration, it will be used if it is up and running. Only when Router A is not running will Router B accept SAs from Router C. This is the behavior without a prefix list.

If you specify a prefix list, the peer will be a default peer only for the prefixes in the list. You can have multiple active default peers when you have a prefix list associated with each. When you do not have any prefix lists, you can configure multiple default peers, but only the first one is the active default peer as long as the router has connectivity to this peer and the peer is alive. If the first configured peer goes down or the connectivity to this peer goes down, the second configured peer becomes the active default, and so on.

MSDP Mesh Groups

An MSDP mesh group is a group of MSDP speakers that have fully meshed MSDP connectivity between one another. In other words, each of the MSDP peers in the group must have an MSDP peering relationship (MSDP connection) to every other MSDP peer in the group. When an MSDP mesh group is configured between a group of MSDP peers, SA message flooding is reduced. Because when an MSDP peer in the group receives an SA message from another MSDP peer in the group, it assumes that this SA message was sent to all the other MSDP peers in the group. As a result, it is not necessary for the receiving MSDP peer to flood the SA message to the other MSDP peers in the group.

Benefits of MSDP Mesh Groups

- Optimizes SA flooding--MSDP mesh groups are particularly useful for optimizing SA flooding when two or more peers are in a group.
- Reduces the amount of SA traffic across the Internet--When MSDP mesh groups are used, SA messages are not flooded to other mesh group peers.
- Eliminates RPF checks on arriving SA messages--When an MSDP mesh group is configured, SA messages are always accepted from mesh group peers.

SA Origination Filters

By default, an RP that is configured to run MSDP will originate SA messages for all local sources for which it is the RP. Local sources that register with an RP, therefore, will be advertised in SA messages, which in some cases is not desirable. For example, if sources inside a PIM-SM domain are using private addresses (for example, network 10.0.0.0/8), you should configure an SA origination filter to restrict those addresses from being advertised to other MSDP peers across the Internet.

To control what sources are advertised in SA messages, you can configure SA origination filters on an RP. By creating SA origination filters, you can control the sources advertised in SA messages as follows:

- You can configure an RP to prevent the device from advertising local sources in SA messages. The device will still forward SA messages from other MSDP peers in the normal fashion; it will just not originate any SA messages for local sources.
- You can configure the device to only originate SA messages for local sources sending to specific groups that match (S, G) pairs defined in the extended access list. All other local sources will not be advertised in SA messages.
- You can configure the device to only originate SA messages for local sources sending to specific groups that the match AS paths defined in an AS-path access list. All other local sources will not be advertised in SA messages.
- You can configure the device to only originate SA messages for local sources that match the criteria defined in the route map. All other local sources will not be advertised in SA messages.
- You configure an SA origination filter that includes an extended access list, an AS-path access list, and route map, or a combination thereof. In this case, all conditions must be true before any local sources are advertised in SA messages.

Use of Outgoing Filter Lists in MSDP

By default, an MSDP-enabled device forwards all SA messages it receives to all of its MSDP peers. However, you can prevent SA messages from being forwarded to MSDP peers by creating outgoing filter lists. Outgoing filter lists apply to all SA messages, whether locally originated or received from another MSDP peer, whereas

SA origination filters apply only to locally originated SA messages. For more information about enabling a filter for MSDP SA messages originated by the local device, see the Controlling SA Messages Originated by an RP for Local Sources section.

By creating an outgoing filter list, you can control the SA messages that a device forwards to a peer as follows:

- You can filter all outgoing SA messages forwarded to a specified MSDP peer by configuring the device to stop forwarding its SA messages to the MSDP peer.
- You can filter a subset of outgoing SA messages forwarded to a specified MSDP peer based on (S, G) pairs defined in an extended access list by configuring the device to only forward SA messages to the MSDP peer that match the (S, G) pairs permitted in an extended access list. The forwarding of all other SA messages to the MSDP peer will be stopped.
- You can filter a subset of outgoing SA messages forwarded to a specified MSDP peer based on match criteria defined in a route map by configuring the device to only forward SA messages that match the criteria defined in the route map. The forwarding of all other SA messages to the MSDP peer will be stopped.
- You can filter a subset of outgoing SA messages from a specified peer based on the announcing RP address contained in the SA message by configuring the device to filter outgoing SA messages based on their origin, even after an SA message has been transmitted across one or more MSDP peers. The forwarding of all other SA messages to the MSDP peer will be stopped.
- You can configure an outgoing filter list that includes an extended access list, a route map, and either an RP access list or an RP route map. In this case, all conditions must be true for the MSDP peer to forward the outgoing SA message.

Caution Arbitrary filtering of SA messages can result in downstream MSDP peers being starved of SA messages for legitimate active sources. Care, therefore, should be taken when using these sorts of filters. Normally, outgoing filter lists are used only to reject undesirable sources, such as sources using private addresses.

Use of Incoming Filter Lists in MSDP

By default, an MSDP-enabled device receives all SA messages sent to it from its MSDP peers. However, you can control the source information that a device receives from its MSDP peers by creating incoming filter lists.

By creating incoming filter lists, you can control the incoming SA messages that a device receives from its peers as follows:

- You can filter all incoming SA messages from a specified MSDP peer by configuring the device to ignore all SA messages sent to it from the specified MSDP peer.
- You can filter a subset of incoming SA messages from a specified peer based on (S, G) pairs defined in an extended access list by configuring the device to only receive SA messages from the MSDP peer that match the (S, G) pairs defined in the extended access list. All other incoming SA messages from the MSDP peer will be ignored.
- You can filter a subset of incoming SA messages from a specified peer based on both (S, G) pairs defined in an extended access list and on match criteria defined in a route map by configuring the device to only receive incoming SA messages that both match the (S, G) pairs defined in the extended access list and

match the criteria defined in the route map. All other incoming SA messages from the MSDP peer will be ignored.

- You can filter a subset of incoming SA messages from a specified peer based on the announcing RP address contained in the SA message by configuring the device to filter incoming SA messages based on their origin, even after the SA message may have already been transmitted across one or more MSDP peers.
- You can configure an incoming filter list that includes an extended access list, a route map, and either an RP access list or an RP route map. In this case, all conditions must be true for the MSDP peer to receive the incoming SA message.

Caution Arbitrary filtering of SA messages can result in downstream MSDP peers being starved of SA messages for legitimate active sources. Care, therefore, should be taken when using these sorts of filters. Normally, incoming filter lists are used only to reject undesirable sources, such as sources using private addresses.

TTL Thresholds in MSDP

The time-to-live (TTL) value provides a means to limit the number of hops a packet can take before being dropped. The **ip multicast ttl-threshold** command is used to specify a TTL for data-encapsulated SA messages sent to specified MSDP peers. By default, multicast data packets in SA messages are sent to an MSDP peer, provided the TTL value of the packet is greater than 0, which is standard TTL behavior.

In general, a TTL-threshold problem can be introduced by the encapsulation of a source’s initial multicast packet in an SA message. Because the multicast packet is encapsulated inside of the unicast SA message (whose TTL is 255), its TTL is not decremented as the SA message travels to the MSDP peer. Furthermore, the total number of hops that the SA message traverses can be drastically different than a normal multicast packet because multicast and unicast traffic may follow completely different paths to the MSDP peer and hence the remote PIM-SM domain. As a result, encapsulated packets can end up violating TTL thresholds. The solution to this problem is to configure a TTL threshold that is associated with any multicast packet that is encapsulated in an SA message sent to a particular MSDP peer using the **ip multicast ttl-threshold** command. The **ip msdp ttl-threshold** command prevents any multicast packet whose TTL in the IP header is less than the TTL value specified for the *ttl-value*argument from being encapsulated in SA messages sent to that peer.

MSDP MIB

The MSDP MIB describes managed objects that can be used to remotely monitor MSDP speakers using SNMP. The MSDP MIB module contains four scalar objects and three tables. The tables are the Requests table, the Peer table, and the Source-Active (SA) Cache table. The Cisco implementation supports the Peer table and SA Cache table only. However, the MSDP implementation used in Cisco IOS software does not associate sending SA requests to peers with group addresses (or group address masks).

Note The MSDP-MIB.my file can be downloaded from the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml .

How to Use MSDP to Interconnect Multiple PIM-SM Domains

The first task is required; all other tasks are optional.

Configuring an MSDP Peer

Note By enabling an MSDP peer, you implicitly enable MSDP.

Before you begin

- IP multicast routing must be enabled and PIM-SM must be configured.
- With the exception of a single MSDP peer, default MSDP peer, and MSDP mesh group scenarios, all MSDP peers must be configured to run BGP prior to being configured for MSDP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp peer** {*peer-name*| *peer-address*} [*connect-source type number*] [**remote-as** *as-number*]
4. **ip msdp description** {*peer-name*| *peer-address*} *text*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp peer** {*peer-name*| *peer-address*} [*connect-source type number*] [**remote-as** *as-number*] **Example:** `Device(config)# ip msdp peer 192.168.1.2 connect-source loopback0`	Enables MSDP and configures an MSDP peer as specified by the DNS name or IP address. **Note** The device that is selected to be configured as an MSDP peer is also usually a BGP neighbor. If it is not, see the Configuring a Default MSDP Peer, on page 111 section or the Configuring an MSDP Mesh Group, on page 112 section.

	Command or Action	Purpose
		• If you specify the **connect-source**keyword, the primary address of the specified local interface *type* and *number* values are used as the source IP address for the TCP connection. The **connect-source**keyword is recommended, especially for MSDP peers on a border that peer with a device inside of a remote domain.
Step 4	**ip msdp description** {*peer-name*| *peer-address*} *text* **Example:** `Device(config)# ip msdp description 192.168.1.2 router at customer a`	(Optional) Configures a description for a specified peer to make it easier to identify in a configuration or in **show** command output.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Shutting Down an MSDP Peer

Perform this optional task to shut down an MSDP peer.

If you are configuring several MSDP peers and you do not want any of the peers to go active until you have finished configuring all of them, you can shut down each peer, configure each peer, and later bring each peer up. You might also want to shut down an MSDP session without losing the configuration for that MSDP peer.

> **Note** When an MSDP peer is shut down, the TCP connection is terminated and not restarted until the peer is brought back up using the **no ip msdp shutdown** command (for the specified peer).

Before you begin

MSDP is running and the MSDP peers must be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp shutdown** {*peer-name* | *peer-address*}
4. Repeat Step 3 to shut down additional MSDP peers.
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp shutdown** {*peer-name* | *peer-address*} **Example:** `(config)# ip msdp shutdown 192.168.1.3`	Administratively shuts down the specified MSDP peer.
Step 4	Repeat Step 3 to shut down additional MSDP peers.	--
Step 5	**end** **Example:** `(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring MSDP MD5 Password Authentication Between MSDP Peers

Perform this optional task to configure MSDP MD5 password authentication between MSDP peers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp password peer** {*peer-name* | *peer-address*} [*encryption-type*] *string*
4. **exit**
5. **show ip msdp peer** [peer-address | peer-name]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip msdp password peer** {*peer-name* | *peer-address*} [*encryption-type*] *string* **Example:** `Device(config)# ip msdp password peer 10.32.43.144 0 test`	Enables MD5 password encryption for a TCP connection between two MSDP peers. **Note** MD5 authentication must be configured with the same password on both MSDP peers; otherwise, the connection between them will not be made. • If you configure or change the password or key, which is used for MD5 authentication between two MSDP peers, the local device does not disconnect the existing session after you configure the password. You must manually disconnect the session to activate the new or changed password.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip msdp peer** [peer-address | peer-name] **Example:** `Device# show ip msdp peer`	(Optional) Displays detailed information about MSDP peers. **Note** Use this command to verify whether MD5 password authentication is enabled on an MSDP peer.

Troubleshooting Tips

If a device has a password configured for an MSDP peer but the MSDP peer does not, a message such as the following will appear on the console while the devices attempt to establish an MSDP session between them:

```
%TCP-6-BADAUTH: No MD5 digest from [peer's IP address]:11003 to [local router's
IP address]:179
```

Similarly, if the two devices have different passwords configured, a message such as the following will appear on the console:

```
%TCP-6-BADAUTH: Invalid MD5 digest from [peer's IP address]:11004 to [local router's
IP address]:179
```

The **debug ip tcp transactions** command is used to display information on significant TCP transactions such as state changes, retransmissions, and duplicate packets. In the context of monitoring or troubleshooting MSDP MD5 password authentication, use the **debug ip tcp transactions** command to verify that the MD5 password is enabled and that the keepalive message is received by the MSDP peer.

Preventing DoS Attacks by Limiting the Number of SA Messages Allowed in the SA Cache from Specified MSDP Peers

Perform this optional (but highly recommended) task to limit the overall number of SA messages that the device can accept from specified MSDP peers. Performing this task protects an MSDP-enabled device from distributed denial-of-service (DoS) attacks.

> **Note** We recommend that you perform this task for all MSDP peerings on the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp sa-limit** {*peer-address* | *peer-name*} *sa-limit*
4. Repeat Step 3 to configure SA limits for additional MSDP peers.
5. **exit**
6. **show ip msdp count** [*as-number*]
7. **show ip msdp peer** [*peer-address* | *peer-name*]
8. **show ip msdp summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp sa-limit** {*peer-address* | *peer-name*} *sa-limit* **Example:** `Device(config)# ip msdp sa-limit 192.168.10.1 100`	Limits the number of SA messages allowed in the SA cache from the specified MSDP.
Step 4	Repeat Step 3 to configure SA limits for additional MSDP peers.	--
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show ip msdp count** [*as-number*] **Example:** `Device# show ip msdp count`	(Optional) Displays the number of sources and groups originated in MSDP SA messages and the number of SA messages from an MSDP peer in the SA cache.
Step 7	**show ip msdp peer** [*peer-address* | *peer-name*] **Example:** `Device# show ip msdp peer`	(Optional) Displays detailed information about MSDP peers. **Note** The output of this command displays the number of SA messages received from MSDP peers that are stored in the cache.
Step 8	**show ip msdp summary** **Example:** `Device# show ip msdp summary`	(Optional) Displays MSDP peer status. **Note** The output of this command displays a per-peer "SA Count" field that displays the number of SAs stored in the cache.

Adjusting the MSDP Keepalive and Hold-Time Intervals

Perform this optional task to adjust the interval at which an MSDP peer will send keepalive messages and the interval at which the MSDP peer will wait for keepalive messages from other peers before declaring them down. By default, it may take as long as 75 seconds for an MSDP peer to detect that a peering session with another MSDP peer has gone down. In network environments with redundant MSDP peers, decreasing the hold-time interval can expedite the reconvergence time of MSDP peers in the event that an MSDP peer fails.

Note We recommend that you do not change the command defaults for the **ip msdp keepalive** command, because the command defaults are in accordance with RFC 3618, *Multicast Source Discovery Protocol*. If your network environment requires that you modify the defaults, you must configure the same time values for the *keepalive-interval* and *hold-time-interval* arguments on both ends of the MSDP peering session.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp keepalive** {*peer-address* | *peer-name*} *keepalive-interval hold-time-interval*
4. Repeat Step 3 to adjust the keepalive message interval for additional MSDP peers.
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp keepalive** {*peer-address* | *peer-name*} *keepalive-interval hold-time-interval* **Example:** `Device(config)# ip msdp keepalive 10.1.1.3 40 55`	Configures the interval at which an MSDP peer will send keepalive messages and the interval at which the MSDP peer will wait for keepalive messages from other peers before declaring them down.
Step 4	Repeat Step 3 to adjust the keepalive message interval for additional MSDP peers.	--
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Adjusting the MSDP Connection-Retry Interval

Perform this optional task to adjust the interval at which MSDP peers will wait after peering sessions are reset before attempting to reestablish the peering sessions. In network environments where fast recovery of SA messages is required, such as in trading floor network environments, you may want to decrease the connection-retry interval to a time value less than the default value of 30 seconds.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp timer** *connection-retry-interval*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip msdp timer** *connection-retry-interval* **Example:** `Device# ip msdp timer 45`	Configures the interval at which MSDP peers will wait after peering sessions are reset before attempting to reestablish the peering sessions.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring MSDP Compliance with IETF RFC 3618

Perform this optional task to configure MSDP peers to be compliant with Internet Engineering Task Force (IETF) RFC 3618 specifications for MSDP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp rpf rfc3618**
4. **end**
5. **show ip msdp rpf-peer** *rp-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp rpf rfc3618** **Example:** `Router(config)# ip msdp rpf rfc3618`	Enables compliance with the peer-RPF forwarding rules specified in IETF RFC 3618.
Step 4	**end** **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config)# end`	
Step 5	**show ip msdp rpf-peer** *rp-address* **Example:** `Router# show ip msdp rpf-peer 192.168.1.5`	(Optional) Displays the unique MSDP peer information from which a router will accept SA messages originating from the specified RP.

Configuring a Default MSDP Peer

Perform this optional task to configure a default MSDP peer.

Before you begin

An MSDP default peer must be a previously configured MSDP peer. Before configuring a default MSDP peer, you must first configure an MSDP peer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp default-peer** {*peer-address* | *peer-name*} [**prefix-list** *list*]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp default-peer** {*peer-address* | *peer-name*} [**prefix-list** *list*] **Example:** `Device(config)# ip msdp default-peer 192.168.1.3`	Configures a default peer from which to accept all MSDP SA messages
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring an MSDP Mesh Group

Perform this optional task to configure an MSDP mesh group.

Note You can configure multiple mesh groups per device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp mesh-group** *mesh-name* {*peer-address* | *peer-name*}
4. Repeat Step 3 to add MSDP peers as members of the mesh group.
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp mesh-group** *mesh-name* {*peer-address* | *peer-name*} **Example:** `(config)# ip msdp mesh-group peermesh`	Configures an MSDP mesh group and indicates that an MSDP peer belongs to that mesh group. **Note** All MSDP peers on a device that participate in a mesh group must be fully meshed with all other MSDP peers in the group. Each MSDP peer on each device must be configured as a peer using the **ip msdp peer** command and also as a member of the mesh group using the **ip msdp mesh-group** command.
Step 4	Repeat Step 3 to add MSDP peers as members of the mesh group.	--
Step 5	**exit** **Example:** `(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Controlling SA Messages Originated by an RP for Local Sources

Perform this task to control SA messages originated by an RP by enabling a filter to restrict which registered sources are advertised in SA messages.

Note For best practice information related to configuring MSDP SA message filters, see the Multicast Source Discovery Protocol SA Filter Recommendations tech note.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp redistribute** [**list** *access-list*] [**asn** *as-access-list*] [**route-map** *map-name*]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp redistribute** [**list** *access-list*] [**asn** *as-access-list*] [**route-map** *map-name*] **Example:** `Device(config)# ip msdp redistribute route-map customer-sources`	Enables a filter for MSDP SA messages originated by the local device. **Note** The **ip msdp redistribute** command can also be used to advertise sources that are known to the RP but not registered. However, it is strongly recommended that you not originate advertisements for sources that have not registered with the RP.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Controlling the Forwarding of SA Messages to MSDP Peers Using Outgoing Filter Lists

Perform this optional task to control the forwarding of SA messages to MSDP peers by configuring outgoing filter lists.

Note For best practice information related to configuring MSDP SA message filters, see the Multicast Source Discovery Protocol SA Filter Recommendations tech note.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp sa-filter out** {*peer-address* | *peer-name*} [**list** *access-list*] [**route-map** *map-name*] [**rp-list** *access-list* | **rp-route-map** *map-name*]
4. Repeat Step 3 to configure outgoing filter lists for additional MSDP peers.
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp sa-filter out** {*peer-address* | *peer-name*} [**list** *access-list*] [**route-map** *map-name*] [**rp-list** *access-list* | **rp-route-map** *map-name*] **Example:** `Device(config)# ip msdp sa-filter out 192.168.1.5 peerone`	Enables a filter for outgoing MSDP messages.
Step 4	Repeat Step 3 to configure outgoing filter lists for additional MSDP peers.	--
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Controlling the Receipt of SA Messages from MSDP Peers Using Incoming Filter Lists

Perform this optional task to control the receipt of incoming SA messages from MSDP peers.

Note For best practice information related to configuring MSDP SA message filters, see the Multicast Source Discovery Protocol SA Filter Recommendations tech note.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp sa-filter in** {*peer-address* | *peer-name*} [**list** *access-list*] [**route-map** *map-name*] [**rp-list** *access-list* | **rp-route-map** *map-name*]
4. Repeat Step 3 to configure incoming filter lists for additional MSDP peers.
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp sa-filter in** {*peer-address* | *peer-name*} [**list** *access-list*] [**route-map** *map-name*] [**rp-list** *access-list* | **rp-route-map** *map-name*] **Example:** `Device(config)# ip msdp sa-filter in 192.168.1.3`	Enables a filter for incoming MSDP SA messages.
Step 4	Repeat Step 3 to configure incoming filter lists for additional MSDP peers.	--
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Using TTL Thresholds to Limit the Multicast Data Sent in SA Messages

Perform this optional task to establish a time to live (TTL) threshold to limit the multicast data sent in SA messages.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp ttl-threshold** {*peer-address* | *peer-name*} *ttl-value*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp ttl-threshold** {*peer-address* | *peer-name*} *ttl-value* **Example:** **Example:** `Device(config)# ip msdp ttl-threshold 192.168.1.5 8`	Sets a TTL value for MSDP messages originated by the local device. • By default, multicast data packets in SA messages are sent to an MSDP peer, provided the TTL value of the packet is greater than 0, which is standard TTL behavior.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Including a Bordering PIM Dense Mode Region in MSDP

Perform this optional task to configure a border device to send SA messages for sources active in a PIM dense mode (PIM-DM) region.

You can have a device that borders a PIM-SM region and a PIM-DM region. By default, sources in the PIM-DM domain are not included in MSDP. You can configure this border device to send SA messages for sources active in the PIM-DM domain. If you do so, it is very important to also configure the **ip msdp redistribute** command to control what local sources from the PIM-DM domain are advertised. Not configuring this command can result in the (S, G) state remaining long after a source in the PIM-DM domain has stopped sending. For configuration information, see the Controlling SA Messages Originated by an RP for Local Sources, on page 113 section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp border sa-address** *type number*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp border sa-address** *type number* **Example:** `Device(config)# ip msdp border sa-address gigabitethernet0/0/0`	Configures the device on the border between a PIM-SM and PIM-DM domain to originate SA messages for active sources in the PIM-DM domain. • The IP address of the interface is used as the originator ID, which is the RP field in the SA message.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring an Originating Address Other Than the RP Address

Perform this optional task to allow an MSDP speaker that originates an SA message to use the IP address of its interface as the RP address in the SA message.

You can also change the originator ID for any one of the following reasons:

- If you configure multiple devices in an MSDP mesh group for Anycast RP.
- If you have a device that borders a PIM-SM domain and a PIM-DM domain. If a device borders a PIM-SM domain and a PIM-DM domain and you want to advertise active sources within the PIM-DM domain, configure the RP address in SA messages to be the address of the originating device's interface.

Before you begin

MSDP is enabled and the MSDP peers are configured. For more information about configuring MSDP peers, see the Configuring an MSDP Peer, on page 103 section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp originator-id** *type number*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp originator-id** *type number* **Example:** `Device(config)# ip msdp originator-id ethernet 1`	Configures the RP address in SA messages to be the address of the originating device's interface.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Monitoring MSDP

Perform this optional task to monitor MSDP SA messages, peers, state, and peer status.

SUMMARY STEPS

1. **enable**
2. **debug ip msdp** [*peer-address* | *peer-name*] [**detail**] [**routes**]
3. **debug ip msdp resets**
4. **show ip msdp count** [*as-number*]
5. **show ip msdp peer** [*peer-address* | *peer-name*]
6. **show ip msdp sa-cache** [*group-address* | *source-address* | *group-name* | *source-name*] [*as-number*]
7. **show ip msdp summary**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device# enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **debug ip msdp** [*peer-address* | *peer-name*] [**detail**] [**routes**]

Use this command to debug MSDP activity.

Use the optional *peer-address* or *peer-name* argument to specify for which peer debug events are logged.

The following is sample output from the **debug ip msdp** command:

Example:

```
Device# debug ip msdp
MSDP debugging is on
Device#
MSDP: 224.150.44.254: Received 1388-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1388, ec: 115, RP: 172.31.3.92
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.92, used EMBGP peer
MSDP: 224.150.44.250: Forward 1388-byte SA to peer
MSDP: 224.150.44.254: Received 1028-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1028, ec: 85, RP: 172.31.3.92
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.92, used EMBGP peer
MSDP: 224.150.44.250: Forward 1028-byte SA to peer
MSDP: 224.150.44.254: Received 1388-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 1388, ec: 115, RP: 172.31.3.111
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.111, used EMBGP peer
MSDP: 224.150.44.250: Forward 1388-byte SA to peer
MSDP: 224.150.44.250: Received 56-byte message from peer
MSDP: 224.150.44.250: SA TLV, len: 56, ec: 4, RP: 192.168.76.241
MSDP: 224.150.44.250: Peer RPF check passed for 192.168.76.241, used EMBGP peer
MSDP: 224.150.44.254: Forward 56-byte SA to peer
MSDP: 224.150.44.254: Received 116-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 116, ec: 9, RP: 172.31.3.111
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.111, used EMBGP peer
MSDP: 224.150.44.250: Forward 116-byte SA to peer
MSDP: 224.150.44.254: Received 32-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 32, ec: 2, RP: 172.31.3.78
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.78, used EMBGP peer
MSDP: 224.150.44.250: Forward 32-byte SA to peer
```

Step 3 **debug ip msdp resets**

Use this command to debug MSDP peer reset reasons.

Example:

```
Device# debug ip msdp resets
```

Step 4 **show ip msdp count** [*as-number*]

Use this command to display the number of sources and groups originated in MSDP SA messages and the number of SA messages from an MSDP peer in the SA cache. The **ip msdp cache-sa-state** command must be configured for this command to produce any output.

The following is sample output from the **show ip msdp count**command:

Example:

```
Device# show ip msdp count
SA State per Peer Counters, <Peer>: <# SA learned>
    192.168.4.4: 8
SA State per ASN Counters, <asn>: <# sources>/<# groups>
    Total entries: 8
    ?: 8/8
```

Step 5 **show ip msdp peer** [*peer-address* | *peer-name*]

Use this command to display detailed information about MSDP peers.

Use the optional *peer-address* or *peer-name* argument to display information about a particular peer.

The following is sample output from the **show ip msdp peer**command:

Example:

```
Device# show ip msdp peer 192.168.4.4
MSDP Peer 192.168.4.4 (?), AS 64512 (configured AS)
  Connection status:
    State: Up, Resets: 0, Connection source: Loopback0 (2.2.2.2)
    Uptime(Downtime): 00:07:55, Messages sent/received: 8/18
    Output messages discarded: 0
    Connection and counters cleared 00:08:55 ago
  SA Filtering:
    Input (S,G) filter: none, route-map: none
    Input RP filter: none, route-map: none
    Output (S,G) filter: none, route-map: none
    Output RP filter: none, route-map: none
  SA-Requests:
    Input filter: none
  Peer ttl threshold: 0
  SAs learned from this peer: 8
  Input queue size: 0, Output queue size: 0
  MD5 signature protection on MSDP TCP connection: not enabled
```

Step 6 **show ip msdp sa-cache** [*group-address* | *source-address* | *group-name* | *source-name*] [*as-number*]

Use this command to display the (S, G) state learned from MSDP peers.

The following is sample output from the **show ip msdp sa-cache**command:

Example:

```
Device# show ip msdp sa-cache
MSDP Source-Active Cache - 8 entries
(10.44.44.5, 239.232.1.0), RP 192.168.4.4, BGP/AS 64512, 00:01:20/00:05:32, Peer 192.168.4.4
(10.44.44.5, 239.232.1.1), RP 192.168.4.4, BGP/AS 64512, 00:01:20/00:05:32, Peer 192.168.4.4
(10.44.44.5, 239.232.1.2), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer 192.168.4.4
(10.44.44.5, 239.232.1.3), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer 192.168.4.4
(10.44.44.5, 239.232.1.4), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer 192.168.4.4
(10.44.44.5, 239.232.1.5), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer 192.168.4.4
(10.44.44.5, 239.232.1.6), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer 192.168.4.4
(10.44.44.5, 239.232.1.7), RP 192.168.4.4, BGP/AS 64512, 00:01:19/00:05:32, Peer 192.168.4.4
```

Step 7 **show ip msdp summary**

Use this command to display MSDP peer status.

The following is sample output from the **show ip msdp summary** command:

Example:

```
Device# show ip msdp summary
MSDP Peer Status Summary
Peer Address     AS    State    Uptime/  Reset SA    Peer Name
                                Downtime Count Count
192.168.4.4      4     Up       00:08:05 0     8     ?
```

Clearing MSDP Connections Statistics and SA Cache Entries

Perform this optional task to clear MSDP connections, statistics, and SA cache entries.

SUMMARY STEPS

1. **enable**
2. **clear ip msdp peer** [*peer-address* | *peer-name*]
3. **clear ip msdp statistics** [*peer-address* | peer-name]
4. **clear ip msdp sa-cache** [*group-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip msdp peer** [*peer-address* | *peer-name*] **Example:** `Device# clear ip msdp peer`	Clears the TCP connection to the specified MSDP peer and resets all MSDP message counters.
Step 3	**clear ip msdp statistics** [*peer-address* | peer-name] **Example:** `Device# clear ip msdp statistics`	Clears the statistics counters for the specified MSDP peer and resets all MSDP message counters.
Step 4	**clear ip msdp sa-cache** [*group-address*] **Example:** `Device# clear ip msdp sa-cache`	Clears SA cache entries. • If the **clear ip msdp sa-cache** is specified with the optional *group-address* argument or *source-address*argument, all SA cache entries are cleared.

	Command or Action	Purpose
		• Use the optional *group-address* argument to clear all SA cache entries associated with a specific group.

Enabling SNMP Monitoring of MSDP

Perform this optional task to enable Simple Network Management Protocol (SNMP) monitoring of MSDP.

Before you begin

- SNMP and MSDP is configured on your devices.
- In each PIM-SM domain there should be a device that is configured as the MSDP speaker. This device must have SNMP and the MSDP MIB enabled.

Note
- All MSDP-MIB objects are implemented as read-only.
- The Requests table is not supported in Cisco's implementation of the MSDP MIB.
- The msdpEstablished notification is not supported in Cisco's implementation of the MSDP MIB.

SUMMARY STEPS

1. **enable**
2. **snmp-server enable traps msdp**
3. **snmp-server host** *host* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth**| **priv** | **noauth**]}] *community-string* [**udp-port** *port-number*] **msdp**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**snmp-server enable traps msdp** **Example:** `Device# snmp-server enable traps msdp`	Enables the sending of MSDP notifications for use with SNMP. **Note** The **snmp-server enable traps msdp** command enables both traps and informs.
Step 3	**snmp-server host** *host* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth**| **priv** | **noauth**]}] *community-string* [**udp-port** *port-number*] **msdp** **Example:**	Specifies the recipient (host) for MSDP traps or informs.

	Command or Action	Purpose
	`Device# snmp-server host examplehost msdp`	
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

You can compare the results of MSDP MIB notifications to the output from the software by using the **show ip msdp summary** and **show ip msdp peer** commands on the appropriate device. You can also compare the results of these commands to the results from SNMP Get operations. You can verify SA cache table entries using the **show ip msdp sa-cache** command. Additional troubleshooting information, such as the local address of the connection, the local port, and the remote port, can be obtained using the output from the **debug ip msdp** command.

Configuration Examples for Using MSDP to Interconnect Multiple PIM-SM Domains

Example: Configuring an MSDP Peer

The following example shows how to establish MSDP peering connections between three MSDP peers:

Device A

```
!
interface Loopback 0
 ip address 10.220.8.1 255.255.255.255
!
ip msdp peer 10.220.16.1 connect-source Loopback0
ip msdp peer 10.220.32.1 connect-source Loopback0
!
```

Device B

```
!
interface Loopback 0
 ip address 10.220.16.1 255.255.255.255
!
ip msdp peer 10.220.8.1 connect connect-source Loopback0
ip msdp peer 10.220.32.1 connect connect-source Loopback0
!
```

Device C

```
!
```

```
interface Loopback 0
 ip address 10.220.32.1 255.255.255.255
!
ip msdp peer 10.220.8.1 connect 10.220.8.1 connect-source Loopback0
ip msdp peer 10.220.16.1 connect 10.220.16.1 connect-source Loopback0
!
```

Example: Configuring MSDP MD5 Password Authentication

The following example shows how to enable MD5 password authentication for TCP connections between two MSDP peers:

Device A

```
!
ip msdp peer 10.3.32.154
ip msdp password peer 10.3.32.154 0 test
!
```

Device B

```
!
ip msdp peer 10.3.32.153
ip msdp password peer 10.3.32.153 0 test
!
```

Configuring MSDP Compliance with IETF RFC 3618 Example

The following example shows how to configure the MSDP peers at 10.10.2.4 and 10.20.1.2 to be compliant with peer-RPF forwarding rules specified in IETF RFC 3618:

```
ip msdp peer 10.10.2.4
ip msdp peer 10.20.1.2
ip msdp rpf rfc3618
```

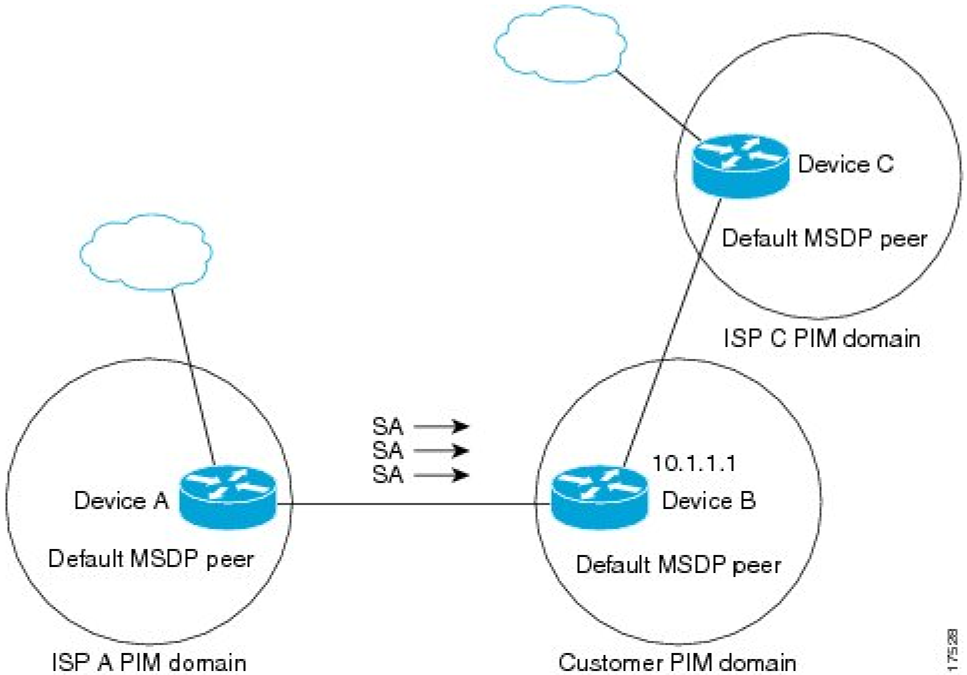
Configuring a Default MSDP Peer Example

The figure illustrates a scenario where default MSDP peers might be used. In the figure, a customer that owns Router B is connected to the internet through two ISPs, one that owns Router A and the other that owns Router C. They are not running (M)BGP between them. In order for the customer to learn about sources in the ISP domain or in other domains, Router B identifies Router A as its default MSDP peer. Router B advertises SA messages to both Router A and Router C, but accepts SA messages either from Router A only or Router C only. If Router A is the first default peer in the configuration, it will be used if it is up and running. Only if Router A is not running will Router B accept SA messages from Router C.

The ISP will also likely use a prefix list to define which prefixes it will accept from the customer router. The customer will define multiple default peers, each having one or more prefixes associated with it.

The customer has two ISPs to use. The customer defines both ISPs as default peers. As long as the first default peer identified in the configuration is up and running, it will be the default peer and the customer will accept all SA messages it receives from that peer.

Figure 16: Default MSDP Peer Scenario



Router B advertises SAs to Router A and Router C, but uses only Router A or Router C to accept SA messages. If Router A is first in the configuration file, it will be used if it is up and running. Only when Router A is not running will Router B accept SAs from Router C. This is the behavior without a prefix list.

If you specify a prefix list, the peer will be a default peer only for the prefixes in the list. You can have multiple active default peers when you have a prefix list associated with each. When you do not have any prefix lists, you can configure multiple default peers, but only the first one is the active default peer as long as the router has connectivity to this peer and the peer is alive. If the first configured peer goes down or the connectivity to this peer goes down, the second configured peer becomes the active default, and so on.

The following example shows a partial configuration of Router A and Router C in the figure. Each of these ISPs may have more than one customer using default peering, like the customer in the figure. In that case, they may have similar configurations. That is, they will only accept SAs from a default peer if the SA is permitted by the corresponding prefix list.

Router A Configuration

```
ip msdp default-peer 10.1.1.1
ip msdp default-peer 10.1.1.1 prefix-list site-b ge 32
ip prefix-list site-b permit 10.0.0.0/8
```

Router C Configuration

```
ip msdp default-peer 10.1.1.1 prefix-list site-b ge 32
ip prefix-list site-b permit 10.0.0.0/8
```

Example: Configuring MSDP Mesh Groups

The following example shows how to configure three devices to be fully meshed members of an MSDP mesh group:

Device A Configuration

```
ip msdp peer 10.2.2.2
ip msdp peer 10.3.3.3
ip msdp mesh-group test-mesh-group 10.2.2.2
ip msdp mesh-group test-mesh-group 10.3.3.3
```

Device B Configuration

```
ip msdp peer 10.1.1.1
ip msdp peer 10.3.3.3
ip msdp mesh-group test-mesh-group 10.1.1.1
ip msdp mesh-group test-mesh-group 10.3.3.3
```

Device C Configuration

```
ip msdp peer 10.1.1.1
ip msdp peer 10.2.2.2
ip msdp mesh-group test-mesh-group 10.1.1.1
ip msdp mesh-group test-mesh-group 10.2.2.2
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference

Standards and RFC

Standard/RFC	Title
RFC 2385	Protection of BGP Sessions via the TCP MD5 Signature Option
RFC 2858	Multiprotocol Extensions for BGP-4
RFC 3618	Multicast Source Discovery Protocol

MIBs

MIB	MIBs Link
MSDP-MIB.my	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Using MSDP to Interconnect Multiple PIM-SM Domains

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 5

PIM Allow RP

This module describes how to configure the PIM Allow RP feature in IPv4 or IPv6 networks for inter-connecting Protocol Independent Multicast (PIM) Sparse Mode (SM) domains with different rendezvous points (RPs). PIM Allow RP enables the receiving device to use its own RP to create state and build shared trees when an incoming (*, G) Join is processed and a different RP is identified. This allows the receiving device to accept the (*, G) Join from the different RP.

Restrictions for PIM Allow RP

- PIM Allow RP only supports connecting PIM SM domains.
- PIM Allow RP is applicable for downstream traffic only, that is, it is only applicable for building the shared tree.
- PIM Allow RP does not work with Auto-RP or Boot Strap Router (BSR). Only static configuration is supported. However, it does allow the embedded RP in the consumer network to be different than the one configured statically in the service provider network.

Information About PIM Allow RP

Rendezvous Points

A rendezvous point (RP) is a role that a router performs when operating in PIM-SM or bidirectional PIM. An RP is required only in networks running PIM-SM or bidirectional PIM. In PIM-SM, only network segments with active receivers that have explicitly requested multicast data will be forwarded the traffic.

An RP acts as the meeting place for sources and receivers of multicast data. In a PIM-SM network, first hop designated routers with directly connected sources initially send their traffic to the RP. This traffic is then forwarded to receivers down a shared distribution tree. By default, when the last hop router with a directly connected receiver receives traffic from the shared tree, it immediately performs a shortest path tree switchover and sends a Join message towards the source, creating a source-based distribution tree between the source and the receiver.

PIM Allow RP

There are three types of networks: publisher, consumer, and transport. Many publisher networks can originate content and many consumer networks can be interested in the content. The transport network, owned and operated by a service provider, connects the publisher and the consumer networks.

The consumer and the transport networks are connected as follows:

For a specific group range, or all-groups range (similar to a default route), the service provider defines a particular rendezvous point (RP), such as RP-A. Reverse path forwarding of RP-A from a consumer device will cause a (*,G) Join to be sent towards the transport network.

For the same group, the service provider may define a different RP, such as RP-B, that is used to build the shared tree within the transport network for G. RP-A and RP-B are typically different RPs and each RP is defined for different group ranges.

RFC 4601 dictates that if a device receives a (*, G) Join and the RP that is specified in the (*, G) Join is different than what the receiving device expects (unknown RPs), the incoming (*, G) Join must be ignored. The PIM Allow RP feature enables the receiving device to use its own RP to create state and build shared trees when an incoming (*, G) Join is processed and a different RP is identified. This allows the receiving device to accept the (*, G) Join from the different RP.

PIM Allow RP is only applicable for downstream traffic, for building the shared tree. It does not work with Auto-RP or BSR. Only static configuration is supported. However, PIM Allow RP does compensate for the embedded RP in the consumer network to be different than the one configured statically in the transport network.

How to Configure PIM Allow RP

Configuring RPs for PIM-SM

Before you begin

All access lists should be configured prior to beginning the configuration task. For information about how to configure an access list, see the “Creating an IP Access List and Applying It to an Interface” module of the *Security Configuration Guide: Access Control Lists* guide.

For IPv6 network devices, you must first enable IPv6 unicast routing on all interfaces of the device on which you want to enable IPv6 multicast routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3.
   - **ip multicast-routing** [**vrf** *vrf-name*] **distributed**
   - **ipv6 multicast-routing** [**vrf** *vrf-name*]

4. **interface** *type number*
5.
    - **ip pim sparse-mode**
    - **ipv6 pim enable**
6. **ipv6 address** {*ipv6-address* | *prefix-length* | *prefix-name* *sub-bits* | *prefix-length*}
7. **no shut**
8. **exit**
9. Repeat Steps 4 through 8 on every interface that uses IP multicast.
10.
    - **ip pim** [**vrf** *vrf-name*] **rp-address** *rp-address* [*access-list*] [**override**]
    - **ipv6 pim** [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-address-list*]
11. **exit**
12. **show ip pim rp** [**mapping**] [*rp-address*]
13. **show ip mroute**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	• **ip multicast-routing** [**vrf** *vrf-name*] **distributed**
	• **ipv6 multicast-routing** [**vrf** *vrf-name*]
	Example:
	`Device(config)# ip multicast-routing`
	`Device(config)# ipv6 multicast-routing`
		• For IPv4: Enables multicast routing on all interfaces of the device. In Cisco IOS XE Release 3.2S and earlier releases, the **distributed** keyword is optional.
		• For IPv6: Enables multicast routing on all interfaces of the device and also enables multicast forwarding for PIM and MLD on all multicast-enabled interfaces of the device.
		Note IPv6 multicast routing is disabled by default when IPv6 unicast routing is enabled. On certain devices, the IPv6 multicast routing must also be enabled in order to use IPv6 unicast routing.

Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	• **ip pim sparse-mode** • **ipv6 pim enable** **Example:**	• For IPv4: Enables PIM. You must use sparse mode. • For IPv6: Enables IPv6 and by default, IPv6 PIM.

	Command or Action	Purpose
	`Device(config-if)# ip pim sparse-mode` `Device(config-if)# ipv6 pim enable`	
Step 6	**ipv6 address** {*ipv6-address* | *prefix-length* | *prefix-name sub-bits* | *prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:DB8::4:4/64`	For IPv6 only: Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 7	**no shut** **Example:** `Device(config-if)# no shut`	Enables an interface.
Step 8	**exit** **Example:** `Device(config-if)# exit`	— Returns to global configuration mode.
Step 9	Repeat Steps 4 through 8 on every interface that uses IP multicast.	

Step 10	• **ip pim** [**vrf** *vrf-name*] **rp-address** *rp-address* [*access-list*] [**override**]
	• **ipv6 pim** [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-address-list*]
	Example:
	`Device(config)# ip pim rp-address 192.0.2.1 acl-sparse`
	`Device(config)# ipv6 pim rp-address 2001:DB8::1:1 acl_sparse1`
		• For IPv4: Configures the address of a PIM RP. If no access list is specified, the RP address is applied to all multicast groups, 224/4.
		• For IPv6: Configures the address of a PIM RP. If no group-address list is specified, the RP address is applied to the entire routable IPv6 multicast group range, excluding SSM, which ranges from FFX[3-f]::/8 to FF3X::/96.

Step 11	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 12	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `Device# show ip pim rp mapping`	(Optional) Displays RPs known in the network and shows how the router learned about each RP.
Step 13	**show ip mroute** **Example:** `Device# show ip mroute`	(Optional) Displays the contents of the IP mroute table.

Enabling PIM Allow RP

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3.
   - **ip pim allow-rp** [**group-list** *access-list* | **rp-list** *access-list* [**group-list** *access-list*]]
   - **ipv6 pim allow-rp** [**group-list** *access-list* | **rp-list** *access-list* [**group-list** *access-list*]]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	• **ip pim allow-rp** [**group-list** *access-list* | **rp-list** *access-list* [**group-list** *access-list*]]
	• **ipv6 pim allow-rp** [**group-list** *access-list* | **rp-list** *access-list* [**group-list** *access-list*]]
	Example:
	`Device(config)# ip pim allow-rp`
	`Device(config)# ipv6 pim allow-rp`
		Enables PIM Allow RP.

Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.

Displaying Information About PIM-SM and RPs

SUMMARY STEPS

1. **enable**
2.
   - **show ip pim** [**vrf** *vrf-name*] **rp** [**metric**] [*rp-address*]
   - **show ipv6 pim** [**vrf** *vrf-name*] **interface** [**state-on**] [**state-off**] [*type number*]
3.
   - **show ip pim** [**vrf** *vrf-name*] **rp mapping** [*rp-address*]
   - **show ipv6 pim** [**vrf** *vrf-name*] **group-map** [*group-name* | *group-address*] | [*group-range* | *group-mask*] [**info-source** {**bsr** | **default** | **embedded-rp** | **static**}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.

Step 2	• **show ip pim** [**vrf** *vrf-name*] **rp** [**metric**] [*rp-address*]
	• **show ipv6 pim** [**vrf** *vrf-name*] **interface** [**state-on**] [**state-off**] [*type number*]
	Example:
	`Device# show ip pim interface`
	`Device# show ipv6 pim interface`
		Displays information about interfaces configured for PIM.

Step 3	• **show ip pim** [**vrf** *vrf-name*] **rp mapping** [*rp-address*]
	• **show ipv6 pim** [**vrf** *vrf-name*] **group-map** [*group-name* | *group-address*] | [*group-range* | *group-mask*] [**info-source** {**bsr** | **default** | **embedded-rp** | **static**}]
	Example:
	`Device# show ipv6 pim rp mapping`
	`Device# show ipv6 pim group-map static`
		Displays the mappings for the PIM group to the active rendezvous points.

Configuration Examples for PIM Allow RP

Example: IPv4 PIM Allow RP

In the following example:

1. The downstream device loopback (Loopback100) creates a static (*,239.1.2.3) Join to a nonexistent RP (11.30.3.3).
2. The static route makes the device think that this RP can be reached through the upstream device via 11.10.2.1, causing the downstream device to send a (*,239.1.2.3) PIM Join with an RP address (11.30.3.3) to the upstream router.
3. When the upstream device receives the (*,239.1.2.3) PIM Join, it realizes that the RP address in the Join (11.30.3.3) is different from the known (configured) interface-to-RP address (11.10.3.3).
4. The PIM allow RP configuration on the upstream device permits the (*,239.1.2.3) to be processed and creates a (*,239.1.2.3) join to the RP (11.10.3.3).

Note If the **pim allow-rp** command is not configured on the upstream device, the upstream device must ignore Joins with different RPs.

```
##################
#   Downstream
```

```
##################
!
hostname downstream-router
!
!
ip multicast-routing distributed
!
!
interface Loopback100
ip address 101.10.1.2 255.255.255.0
ip igmp static-group 239.1.2.3
ip pim sparse-dense-mode
no shut
!
interface Ethernet1/2
ip address 11.10.2.2 255.255.255.0
ip pim sparse-dense-mode
no shut
!
router ospf 200
network 11.0.0.0 0.255.255.255 area 1
network 101.0.0.0 0.255.255.255 area 1
!
ip pim rp-address 11.30.3.3
ip mroute 11.30.3.3 255.255.255.255 11.10.2.1
!
end


##################
#   Upstream
##################
!
hostname Upstream-router
!
!
ip multicast-routing distributed
!
!
interface FastEthernet0/0/2
ip address 11.10.2.1 255.255.255.0
ip pim sparse-dense-mode
no shut
!
interface FastEthernet0/0/4
! interface to RP (11.10.3.3)
ip address 10.10.4.1 255.255.255.0
ip pim sparse-dense-mode
no shut
!
router ospf 200
network 10.0.0.0 0.255.255.255 area 1
network 11.0.0.0 0.255.255.255 area 1
!
ip pim rp-address 11.10.3.3
ip pim allow-rp
!
end
```

Example: IPv6 PIM Allow RP

In the following example:

1. The downstream device loopback creates an static (*,FF03::1) Join to a non-existent RP (80::1:1:3).
2. The static route makes the device think that this RP can be reach via the upstream device via 10::1:1:1 and causes the downstream device to send a (*,FF03::1) PIM Join with an RP address (80::1:1:3) to the upstream device.
3. When the upstream device receives the (*,FF03::1) PIM Join, it realizes that the RP address in the Join (80::1:1:3) is different from the (known) configured address (20::1:1:3).
4. The PIM allow RP configuration on the upstream device permits the (*,FF03::1) to be processed, and creates a (*,FF03::1) Join to the RP (20::1:1:3).

Note If the **pim allow-rp** command is not configured on the upstream device, the upstream device must ignore Joins with different RPs .

```
##################
# Downstream
##################

!
hostname downstream-router
!
!
ipv6 unicast-routing
ipv6 multicast-routing
!
!
interface Loopback100
ipv6 address FE80::50:1:2 link-local
ipv6 address 50::1:1:2/64
ipv6 enable
ipv6 ospf 1 area 0
ipv6 mld join-group FF03::1
!
interface Ethernet1/2
ipv6 address FE80::10:1:2 link-local
ipv6 address 10::1:1:2/64
ipv6 enable
ipv6 ospf 1 area 0
no keepalive
!
!
ipv6 pim rp-address 80::1:1:3
ipv6 route 80::1:1:3/128 10::1:1:1 multicast
!
ipv6 router ospf 1
router-id 205.2.0.2
!
!
end
```

```
###################
#  Upstream
###################
!
hostname Upstream-router
!
!
ipv6 unicast-routing
ipv6 multicast-routing
!
!
interface FastEthernet0/0/2
ipv6 address FE80::10:1:1 link-local
ipv6 address 10::1:1:1/64
ipv6 enable
ipv6 ospf 1 area 0
!
interface FastEthernet0/0/3
! interface to the RP (20::1:1:3)
ipv6 address FE80::20:1:1 link-local
ipv6 address 20::1:1:1/64
ipv6 enable
ipv6 ospf 1 area 0
!
!
ipv6 pim rp-address 20::1:1:3
ipv6 pim allow-rp
!
ipv6 router ospf 1
router-id 205.1.0.1
!
!
end
```

Additional References for PIM Allow RP

Standards and RFCs

Standard/RFC	Title
RFC 4601	*Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for PIM Allow RP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 6

Configuring Source Specific Multicast

This module describes how to configure Source Specific Multicast (SSM). The Source Specific Multicast feature is an extension of IP multicast where datagram traffic is forwarded to receivers from only those multicast sources to which the receivers have explicitly joined. For multicast groups configured for SSM, only source-specific multicast distribution trees (no shared trees) are created.

Restrictions for Source Specific Multicast

Legacy Applications Within the SSM Range Restrictions

Existing applications in a network predating SSM will not work within the SSM range unless they are modified to support (S, G) channel subscriptions or are enabled through URL Rendezvous Directory (URD). Therefore, enabling SSM in a network may cause problems for existing applications if they use addresses within the designated SSM range.

IGMP v3lite and URD Require a Cisco Last Hop Router

SSM and IGMPv3 are solutions that are being standardized in the IETF. However, IGMP v3lite and URD are Cisco-developed solutions. For IGMP v3lite and URD to operate properly for a host, the last hop router toward that host must be a Cisco router with IGMP v3lite or URD enabled.

Note This limitation does not apply to an application using the Host Side IGMP Library (HSIL) if the host has kernel support for IGMPv3, because then the HSIL will use the kernel IGMPv3 instead of IGMP v3lite.

Address Management Restrictions

Address management is still necessary to some degree when SSM is used with Layer 2 switching mechanisms. Cisco Group Management Protocol (CGMP), IGMP snooping, or Router-Port Group Management Protocol

(RGMP) currently support only group-specific filtering, not (S, G) channel-specific filtering. If different receivers in a switched network request different (S, G) channels sharing the same group, then they will not benefit from these existing mechanisms. Instead, both receivers will receive all (S, G) channel traffic (and filter out the unwanted traffic on input). Because of the ability of SSM to reuse the group addresses in the SSM range for many independent applications, this situation can lead to less than expected traffic filtering in a switched network. For this reason it is important to follow the recommendations set forth in the IETF drafts for SSM to use random IP addresses out of the SSM range for an application to minimize the chance for reuse of a single address within the SSM range between different applications. For example, an application service providing a set of television channels should, even with SSM, use a different group for each television (S, G) channel. This setup will guarantee that multiple receivers to different channels within the same application service will never experience traffic aliasing in networks that include Layer 2 switches.

IGMP Snooping and CGMP Limitations

IGMPv3 uses new membership report messages that may not be recognized correctly by older IGMP Snooping switches, in which case hosts will not properly receive traffic. This situation is not an issue if URD or IGMP v3lite is used with hosts where the operating system is not upgraded for IGMPv3, because IGMP v3lite and URD rely only on IGMPv1 or IGMPv2 membership reports.

URD Intercept URL Limitations

A URD intercept URL string must be fewer than 256 bytes in length, starting from the */path* argument. In the HTTP/TCP connection, this string must also be contained within a single TCP/IP packet. For example, for a 256-byte string, a link maximum transmission unit (MTU) of 128 bytes between the host and intercepting router would cause incorrect operation of URD.

State Maintenance Limitations

In PIM-SSM, the last hop router will continue to periodically send (S, G) join messages if appropriate (S, G) subscriptions are on the interfaces. Therefore, as long as receivers send (S, G) subscriptions, the shortest path tree (SPT) state from the receivers to the source will be maintained, even if the source is not sending traffic for longer periods of time (or even never).

This case is opposite to PIM-SM, where (S, G) state is maintained only if the source is sending traffic and receivers are joining the group. If a source stops sending traffic for more than 3 minutes in PIM-SM, the (S, G) state will be deleted and only reestablished after packets from the source arrive again through the RPT. Because no mechanism in PIM-SSM notifies a receiver that a source is active, the network must maintain the (S, G) state in PIM-SSM as long as receivers are requesting receipt of that channel.

HSIL Limitations

As explained in the IGMP v3lite Host Signalling, on page 144 concept, the HSIL tries to determine if the host operating system supports IGMPv3. This check is made so that a single application can be used both on hosts where the operating system has been upgraded to IGMPv3 and on hosts where the operating system only supports IGMPv1 or IGMPv2.

Checking for the availability of IGMPv3 in the host operating system can only be made by the HSIL if IGMPv3 kernel support exists for at least one version of this operating system at the time when the HSIL was provided. If such an IGMPv3 kernel implementation has become available only recently, then users may need to also upgrade the HSIL on their hosts so that applications compiled with the HSIL will then dynamically bind to the newest version of the HSIL, which should support the check for IGMPv3 in the operating system kernel. Upgrading the HSIL can be done independently of upgrading the application itself.

Information About Source Specific Multicast

SSM Overview

Source Specific Multicast (SSM). SSM is an extension of IP multicast where datagram traffic is forwarded to receivers from only those multicast sources that the receivers have explicitly joined. For multicast groups configured for SSM, only source-specific multicast distribution trees (not shared trees) are created.

SSM Components

SSM is a datagram delivery model that best supports one-to-many applications, also known as broadcast applications.

SSM is a core networking technology for Cisco's implementation of IP multicast solutions targeted for audio and video broadcast application environments and is described in RFC 3569. The following components together support the implementation of SSM:

- Protocol Independent Multicast source-specific mode (PIM-SSM)
- Internet Group Management Protocol Version 3 (IGMPv3)

Protocol Independent Multicast (PIM) SSM, or PIM-SSM, is the routing protocol that supports the implementation of SSM and is derived from PIM sparse mode (PIM-SM). IGMP is the Internet Engineering Task Force (IETF) standards track protocol used for hosts to signal multicast group membership to routers. IGMP Version 3 supports source filtering, which is required for SSM. IGMP For SSM to run with IGMPv3, SSM must be supported in the router, the host where the application is running, and the application itself.

How SSM Differs from Internet Standard Multicast

The standard IP multicast infrastructure in the Internet and many enterprise intranets is based on the PIM-SM protocol and Multicast Source Discovery Protocol (MSDP). These protocols have proved to be reliable, extensive, and efficient. However, they are bound to the complexity and functionality limitations of the Internet Standard Multicast (ISM) service model. For example, with ISM, the network must maintain knowledge about which hosts in the network are actively sending multicast traffic. With SSM, this information is provided by receivers through the source addresses relayed to the last-hop devices by IGMPv3. SSM is an incremental response to the issues associated with ISM and is intended to coexist in the network with the protocols developed for ISM. In general, SSM provides IP multicast service for applications that utilize SSM.

ISM service is described in RFC 1112. This service consists of the delivery of IP datagrams from any source to a group of receivers called the multicast host group. The datagram traffic for the multicast host group consists of datagrams with an arbitrary IP unicast source address S and the multicast group address G as the IP destination address. Systems will receive this traffic by becoming members of the host group. Membership in a host group simply requires signaling the host group through IGMP Version 1, 2, or 3.

In SSM, delivery of datagrams is based on (S, G) channels. Traffic for one (S, G) channel consists of datagrams with an IP unicast source address S and the multicast group address G as the IP destination address. Systems will receive this traffic by becoming members of the (S, G) channel. In both SSM and ISM, no signaling is required to become a source. However, in SSM, receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources. In other words, receivers can receive traffic only from (S, G) channels to which they are subscribed, whereas in ISM, receivers need not know the IP addresses of sources from which they receive their traffic. The proposed standard approach for channel subscription signaling utilizes IGMP INCLUDE mode membership reports, which are supported only in IGMP Version 3.

SSM can coexist with the ISM service by applying the SSM delivery model to a configured subset of the IP multicast group address range. The Internet Assigned Numbers Authority (IANA) has reserved the address range from 232.0.0.0 through 232.255.255.255 for SSM applications and protocols. The software allows SSM configuration for an arbitrary subset of the IP multicast address range from 224.0.0.0 through 239.255.255.255. When an SSM range is defined, an existing IP multicast receiver application will not receive any traffic when it tries to use addresses in the SSM range unless the application is modified to use explicit (S, G) channel subscription or is SSM-enabled through a URL Rendezvous Directory (URD).

SSM Operations

An established network in which IP multicast service is based on PIM-SM can support SSM services. SSM can also be deployed alone in a network without the full range of protocols that are required for interdomain PIM-SM. That is, SSM does not require an RP, so there is no need for an RP mechanism such as Auto-RP, MSDP, or bootstrap router (BSR).

If SSM is deployed in a network that is already configured for PIM-SM, then only the last-hop routers must be upgraded to a software image that supports SSM. Routers that are not directly connected to receivers do not have to upgrade to a software image that supports SSM. In general, these non-last-hop routers must only run PIM-SM in the SSM range. They may need additional access control configuration to suppress MSDP signaling, registering, or PIM-SM shared-tree operations from occurring within the SSM range.

The SSM mode of operation is enabled by configuring the SSM range using the **ip pim ssm** global configuration command. This configuration has the following effects:

- For groups within the SSM range, (S, G) channel subscriptions are accepted through IGMPv3 INCLUDE mode membership reports.
- PIM operations within the SSM range of addresses change to PIM-SSM, a mode derived from PIM-SM. In this mode, only PIM (S, G) Join and Prune messages are generated by the router. Incoming messages related to rendezvous point tree (RPT) operations are ignored or rejected, and incoming PIM register messages are immediately answered with Register-Stop messages. PIM-SSM is backward-compatible with PIM-SM unless a router is a last-hop router. Therefore, routers that are not last-hop routers can run PIM-SM for SSM groups (for example, if they do not yet support SSM).
- For groups within the SSM range, no MSDP Source-Active (SA) messages within the SSM range will be accepted, generated, or forwarded.

IGMPv3 Host Signaling

IGMPv3 is the third version of the IETF standards track protocol in which hosts signal membership to last-hop routers of multicast groups. IGMPv3 introduces the ability for hosts to signal group membership that allows filtering capabilities with respect to sources. A host can signal either that it wants to receive traffic from all sources sending to a group except for some specific sources (a mode called EXCLUDE) or that it wants to receive traffic only from some specific sources sending to the group (a mode called INCLUDE).

IGMPv3 can operate with both ISM and SSM. In ISM, both EXCLUDE and INCLUDE mode reports are accepted by the last-hop router. In SSM, only INCLUDE mode reports are accepted by the last-hop router.

Benefits of

IP Multicast Address Management Not Required

In the ISM service, applications must acquire a unique IP multicast group address because traffic distribution is based only on the IP multicast group address used. If two applications with different sources and receivers use the same IP multicast group address, then receivers of both applications will receive traffic from the senders of both applications. Even though the receivers, if programmed appropriately, can filter out the unwanted traffic, this situation would cause generally unacceptable levels of unwanted traffic.

Allocating a unique IP multicast group address for an application is still a problem. Most short-lived applications use mechanisms like Session Description Protocol (SDP) and Session Announcement Protocol (SAP) to get a random address, a solution that does not work well with a rising number of applications in the Internet. The best current solution for long-lived applications is described in RFC 2770, but this solution suffers from the restriction that each autonomous system is limited to only 255 usable IP multicast addresses.

In SSM, traffic from each source is forwarded between routers in the network independent of traffic from other sources. Thus different sources can reuse multicast group addresses in the SSM range.

Denial of Service Attacks from Unwanted Sources Inhibited

In SSM, multicast traffic from each individual source will be transported across the network only if it was requested (through IGMPv3, IGMP v3lite, or URD memberships) from a receiver. In contrast, ISM forwards traffic from any active source sending to a multicast group to all receivers requesting that multicast group. In Internet broadcast applications, this ISM behavior is highly undesirable because it allows unwanted sources to easily disturb the actual Internet broadcast source by simply sending traffic to the same multicast group. This situation depletes bandwidth at the receiver side with unwanted traffic and thus disrupts the undisturbed reception of the Internet broadcast. In SSM, this type of denial of service (DoS) attack cannot be made by simply sending traffic to a multicast group.

Easy to Install and Manage

SSM is easy to install and provision in a network because it does not require the network to maintain which active sources are sending to multicast groups. This requirement exists in ISM (with IGMPv1, IGMPv2, or IGMPv3).

The current standard solutions for ISM service are PIM-SM and MSDP. Rendezvous point (RP) management in PIM-SM (including the necessity for Auto-RP or BSR) and MSDP is required only for the network to learn about active sources. This management is not necessary in SSM, which makes SSM easier than ISM to install and manage, and therefore easier than ISM to operationally scale in deployment. Another factor that contributes to the ease of installation of SSM is the fact that it can leverage preexisting PIM-SM networks and requires only the upgrade of last hop routers to support IGMPv3, IGMP v3lite, or URD.

Ideal for Internet Broadcast Applications

The three benefits previously described make SSM ideal for Internet broadcast-style applications for the following reasons:

- The ability to provide Internet broadcast services through SSM without the need for unique IP multicast addresses allows content providers to easily offer their service (IP multicast address allocation has been a serious problem for content providers in the past).
- The prevention against DoS attacks is an important factor for Internet broadcast services because, with their exposure to a large number of receivers, they are the most common targets for such attacks.

- The ease of installation and operation of SSM makes it ideal for network operators, especially in those cases where content needs to be forwarded between multiple independent PIM domains (because there is no need to manage MSDP for SSM between PIM domains).

IGMP v3lite Host Signalling

IGMP v3lite is a Cisco-developed transitional solution for application developers to immediately start programming SSM applications. It allows you to write and run SSM applications on hosts that do not yet support IGMPv3 in their operating system kernel.

Applications must be compiled with the Host Side IGMP Library (HSIL) for IGMP v3lite. This software provides applications with a subset of the IGMPv3 applications programming interface (API) that is required to write SSM applications. HSIL was developed for Cisco by Talarian and is available from the following web page:

http://www.talarianmulticast.com/cgi-bin/igmpdownld

One part of the HSIL is a client library linked to the SSM application. It provides the SSM subset of the IGMPv3 API to the SSM application. If possible, the library checks whether the operating system kernel supports IGMPv3. If it does, then the API calls simply are passed through to the kernel. If the kernel does not support IGMPv3, then the library uses the IGMP v3lite mechanism.

When using the IGMP v3lite mechanism, the library tells the operating system kernel to join to the whole multicast group, because joining to the whole group is the only method for the application to receive traffic for that multicast group (if the operating system kernel only supports IGMPv1 or IGMPv2). In addition, the library signals the (S, G) channel subscriptions to an IGMP v3lite server process, which is also part of the HSIL. A server process is needed because multiple SSM applications may be on the same host. This server process will then send IGMP v3lite-specific (S, G) channel subscriptions to the last hop Cisco IOS router, which needs to be enabled for IGMP v3lite. This router will then “see” both the IGMPv1 or IGMPv2 group membership report from the operating system kernel and the (S, G) channel subscription from the HSIL daemon. If the router sees both of these messages, it will interpret them as an SSM (S, G) channel subscription and join to the channel through PIM-SSM. We recommend referring to the documentation accompanying the HSIL software for further information on how to utilize IGMP v3lite with your application.

IGMP v3lite is supported by Cisco only through the API provided by the HSIL, not as a function of the router independent of the HSIL. By default, IGMP v3lite is disabled. When IGMP v3lite is configured through the **ip igmp v3lite** interface configuration command on an interface, it will be active only for IP multicast addresses in the SSM range.

URD Host Signalling

URD is a Cisco-developed transitional solution that allows existing IP multicast receiver applications to be used with SSM without the need to modify the application and change or add any software on the receiver host running the application. URD is a content provider solution in which the receiver applications can be started or controlled through a web browser.

URD operates by passing a special URL from the web browser to the last hop router. This URL is called a URD intercept URL. A URD intercept URL is encoded with the (S, G) channel subscription and has a format that allows the last hop router to easily intercept it.

As soon as the last hop router intercepts both an (S, G) channel subscription encoded in a URD intercept URL and sees an IGMP group membership report for the same multicast group from the receiver application, the last hop router will use PIM-SSM to join toward the (S, G) channel as long as the application maintains the membership for the multicast group G. The URD intercept URL is thus only needed initially to provide the last hop router with the address of the sources to join to.

A URD intercept URL has the following syntax:

```
http://
webserver
:465/
path
?group=
group
&source=
source1
&...source=
sourceN
&
```

The *webserver* string is the name or IP address to which the URL is targeted. This target need not be the IP address of an existing web server, except for situations where the web server wants to recognize that the last hop router failed to support the URD mechanism. The number 465 indicates the URD port. Port 465 is reserved for Cisco by the IANA for the URD mechanism so that no other applications can use this port.

When the browser of a host encounters a URD intercept URL, it will try to open a TCP connection to the web server on port 465. If the last hop router is enabled for URD on the interface where the router receives the TCP packets from the host, it will intercept all packets for TCP connections destined to port 465 independent of the actual destination address of the TCP connection (independent of the address of the web server). Once intercepted, the last hop router will "speak" a very simple subset of HTTP on this TCP connection, emulating a web server. The only HTTP request that the last hop router will understand and reply to is the following GET request:

```
GET
argument
 HTTP/1.0
argument
 = /
path
?group=
group
&source=
source1
&...source=
sourceN
&
```

When it receives a GET command, the router tries to parse the argument according to this syntax to derive one or more (S, G) channel memberships. The *path* string of the argument is anything up to, but not including, the first question mark, and is ignored. The *group* and *source1* through *sourceN* strings are the IP addresses or fully qualified domain names of the channels for which this argument is a subscription request. If the argument matches the syntax shown, the router interprets the argument to be subscriptions for the channels (*source1* , *group*) through (*sourceN* , *group*).

The router will accept the channel subscriptions if the following conditions are met:

- The IP address of the multicast group is within the SSM range.
- The IP address of the host that originated the TCP connection is directly connected to the router.

If the channel subscription is accepted, the router will respond to the TCP connection with the following HTML page format:

```
HTTP/1.1 200 OK
Server:cisco IOS
Content-Type:text/html
<html>
<body>
Retrieved URL string successfully
</body>
</html>
```

If an error condition occurs, the <body> part of the returned HTML page will carry an appropriate error message. The HTML page is a by-product of the URD mechanism. This returned text may, depending on how the web pages carrying a URD intercept URL are designed, be displayed to the user or be sized so that the actual returned HTML page is invisible.

The primary effect of the URD mechanism is that the router will remember received channel subscriptions and will match them against IGMP group membership reports received by the host. The router will "remember" a URD (S, G) channel subscription for up to 3 minutes without a matching IGMP group membership report. As soon as the router sees that it has received both an IGMP group membership report for a multicast group G and a URD (S, G) channel subscription for the same group G, it will join the (S, G) channel through PIM-SSM. The router will then continue to join to the (S, G) channel based only on the presence of a continuing IGMP membership from the host. Thus, one initial URD channel subscription is all that is needed to be added through a web page to enable SSM with URD.

If the last hop router from the receiver host is not enabled for URD, then it will not intercept the HTTP connection toward the web server on port 465. This situation will result in a TCP connection to port 465 on the web server. If no further provisions on the web server are taken, then the user may see a notice (for example, "Connection refused") in the area of the web page reserved for displaying the URD intercept URL (if the web page was designed to show this output). It is also possible to let the web server "listen" to requests on port 465 and install a Common Gateway Interface (CGI) script that would allow the web server to know if a channel subscription failed (for example, to subsequently return more complex error descriptions to the user).

Because the router returns a Content-Type of text and HTML, the best way to include the URD intercept URL into a web page is to use a frame. By defining the size of the frame, you can also hide the URD intercept URL on the displayed page.

By default, URD is disabled on all interfaces. When URD is configured through the **ip urd** interface configuration command on an interface, it will be active only for IP multicast addresses in the SSM range.

How to Configure Source Specific Multicast

Configuring SSM

To configure SSM, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **ip pim ssm** [**default** | **range***access-list*]
2. Router(config)# **interface** type number
3. Router(config-if)# **ip pim** {**sparse-mode** | **sparse-dense-mode**}
4. Do one of the following:
   - Router(config-if)# **ip igmp version 3**

-
-
-
- Router(config-if)# **ip igmp v3lite**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **ip pim ssm** [**default** | **range***access-list*]	Defines the SSM range of IP multicast addresses.
Step 2	Router(config)# **interface** type number	Selects an interface that is connected to hosts on which IGMPv3, IGMP v3lite, and URD can be enabled.
Step 3	Router(config-if)# **ip pim** {**sparse-mode** | **sparse-dense-mode**}	Enables PIM on an interface. You must use either sparse mode or sparse-dense mode.
Step 4	Do one of the following: • Router(config-if)# **ip igmp version 3** • • • • Router(config-if)# **ip igmp v3lite** **Example:** **Example:** **Example:** **Example:** `Router(config-if)# ip urd`	Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2. or Enables the acceptance and processing of IGMP v3lite membership reports on an interface. or Enables interception of TCP packets sent to the reserved URD port 465 on an interface and processing of URD channel subscription reports.

Configuration Examples of Source Specific Multicast

tbd

SSM with IGMPv3 Example

The following example shows how to configure a router (running IGMPv3) for SSM:

```
ip multicast-routing
!
interface GigabitEthernet3/1/0
 ip address 172.21.200.203 255.255.255.0
```

```
 description backbone interface
 ip pim sparse-mode
!
interface GigabitEthernet3/2/0
 ip address 131.108.1.2 255.255.255.0
 ip pim sparse-mode
 description ethernet connected to hosts
 ip igmp version 3
!
ip pim ssm default
```

SSM with IGMP v3lite and URD Example

The following example shows how to configure IGMP v3lite and URD on interfaces connected to hosts for SSM. Configuring IGMP v3lite and URD is not required or recommended on backbone interfaces.

```
interface gigabitethernet 3/1/1
 ip address 172.21.200.203 255.255.255.0
 ip pim sparse-dense-mode
 description gigabitethernet connected to hosts
!
interface gigabitethernet 1/1/1
description gigabitethernet connected to hosts
 ip address 131.108.1.2 255.255.255.0
 ip pim sparse-dense-mode
 ip urd
 ip igmp v3lite
```

SSM Filtering Example

The following example shows how to configure filtering on legacy RP routers running software releases that do not support SSM routing. This filtering will suppress all unwanted PIM-SM and MSDP traffic in the SSM range. Without this filtering, SSM will still operate, but there may be additional RPT traffic if legacy first hop and last hop routers exist in the network.

```
ip access-list extended  no-ssm-range
  deny   ip any 232.0.0.0 0.255.255.255 ! SSM range
  permit ip any any
! Deny registering in SSM range
ip pim accept-register list no-ssm-range
ip access-list extended msdp-nono-list
  deny   ip any 232.0.0.0 0.255.255.255 ! SSM Range
  ! .
  ! .
  ! .
  ! See ftp://ftpeng.cisco.com/ipmulticast/config-notes/msdp-sa-filter.txt for other SA
  ! messages that typically need to be filtered.
  permit ip any any
! Filter generated SA messages in SSM range. This configuration is only needed if there
! are directly connected sources to this router. The “ip pim accept-register” command
! filters remote sources.
ip msdp redistribute list msdp-nono-list
! Filter received SA messages in SSM range. “Filtered on receipt” means messages are
! neither processed or forwarded. Needs to be configured for each MSDP peer.
ip msdp sa-filter in msdp-peer1 list msdp-nono-list
! .
! .
```

```
! .
ip msdp sa-filter in msdp-peerN list msdp-nono-list
```

Additional References

The following sections provide references related to Source Specific Multicast.

Related Documents

Related Topic	Document Title
PIM-SM and SSM concepts and configuration examples	"Configuring Basic IP Multicast " module
IP multicast commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing standards has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Source Specific Multicast

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 7

Tunneling to Connect Non-IP Multicast Areas

This module describes how to configure a Generic Route Encapsulation (GRE) tunnel to tunnel IP multicast packets between non-IP multicast areas. The benefit is that IP multicast traffic can be sent from a source to a multicast group, over an area where IP multicast is not supported.

Prerequisites for Tunneling to Connect Non-IP Multicast Areas

This module assumes you understand the concepts in the "IP Multicast Technology Overview" module.

Information About Tunneling to Connect Non-IP Multicast Areas

Benefits of Tunneling to Connect Non-IP Multicast Areas

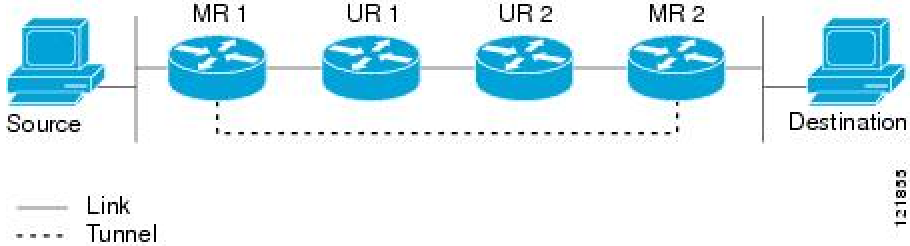
- If the path between a source and a group member (destination) does not support IP multicast, a tunnel between them can transport IP multicast packets.
- Per packet load balancing can be used. Load balancing in IP multicast is normally per (S,G). Therefore, (S1, G) can go over Link X and (S2, G) can go over Link Y, where X and Y are parallel links. If you create a tunnel between the routers, you can get per packet load balancing because the load balancing is done on the tunnel unicast packets.

IP Multicast Static Route

IP multicast static routes (mroutes) allow you to have multicast paths diverge from the unicast paths. When using Protocol Independent Multicast (PIM), the router expects to receive packets on the same interface where it sends unicast packets back to the source. This expectation is beneficial if your multicast and unicast topologies are congruent. However, you might want unicast packets to take one path and multicast packets to take another.

The most common reason for using separate unicast and multicast paths is tunneling. When a path between a source and a destination does not support multicast routing, a solution is to configure two routers with a GRE tunnel between them. In the figure, each unicast router (UR) supports unicast packets only; each multicast router (MR) supports multicast packets.

Figure 17: Tunnel for Multicast Packets



In the figure, Source delivers multicast packets to Destination by using MR 1 and MR 2. MR 2 accepts the multicast packet only if it believes it can reach Source over the tunnel. If this situation is true, when Destination sends unicast packets to Source, MR 2 sends them over the tunnel. The check that MR2 can reach Source over the tunnel is a Reverse Path Forwarding (RPF) check, and the static mroute allows the check to be successful when the interface that the multicast packet arrives on is not the unicast path back to the source. Sending the packet over the tunnel could be slower than natively sending it through UR 2, UR 1, and MR 1.

A multicast static route allows you to use the configuration in the figure by configuring a static multicast source. The system uses the configuration information instead of the unicast routing table to route the traffic. Therefore, multicast packets can use the tunnel without having unicast packets use the tunnel. Static mroutes are local to the router they are configured on and not advertised or redistributed in any way to any other router.

How to Connect Non-IP Multicast Areas

Configuring a Tunnel to Connect Non-IP Multicast Areas

Configure a multicast static route if you want your multicast paths to differ from your unicast paths. For example, you might have a tunnel between two routers because the unicast path between a source and destination does not support multicast routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **ip pim sparse-mode**
6. **tunnel source** {*ip-address* | *type number*}
7. **tunnel destination** {*hostname* | *ip-address*}
8. Repeat Steps 1 through 7 on the router at the opposite end of the tunnel, reversing the tunnel source and destination addresses.
9. **end**
10. **ip mroute** *source-address mask* **tunnel** *number* [*distance*]

11. **ip mroute** *source-address mask* **tunnel** *number* [*distance*]
12. **end**
13. **show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*]
14. **show ip rpf** {*source-address* | *source-name*} [**metric**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 0`	Configures a tunnel interface.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered gigabitethernet 0/0/0`	Enables IP processing without assigning an IP address to the interface.
Step 5	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables PIM sparse mode on the tunnel interface.
Step 6	**tunnel source** {*ip-address*	*type number*} **Example:** `Router(config-if)# tunnel source 100.1.1.1`	Configures the tunnel source.
Step 7	**tunnel destination** {*hostname*	*ip-address*} **Example:** `Router(config-if)# tunnel destination 100.1.5.3`	Configures the tunnel destination.
Step 8	Repeat Steps 1 through 7 on the router at the opposite end of the tunnel, reversing the tunnel source and destination addresses.	Router A’s tunnel source address will match Router B’s tunnel destination address. Router A’s tunnel destination address will match Router B’s tunnel source address.

	Command or Action	Purpose
Step 9	**end** **Example:** `Router(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 10	**ip mroute** *source-address mask* **tunnel** *number* [*distance*] **Example:** `Router(config)# ip mroute 0.0.0.0 0.0.0.0 tunnel 0`	Configures a static multicast route over which to reverse path forward to the other end of the tunnel. • Because the use of the tunnel makes the multicast topology incongruent with the unicast topology, and only multicast traffic traverses the tunnel, you must configure the routers to reverse path forward correctly over the tunnel. • When a source range is specified, the mroute applies only to those sources. • In the example, the *source-address* and *mask* of 0.0.0.0 0.0.0.0 indicate any address. • The shorter distance is preferred. • The default distance is 0.
Step 11	**ip mroute** *source-address mask* **tunnel** *number* [*distance*] **Example:** `Router(config)# ip mroute 0.0.0.0 0.0.0.0 tunnel 0`	Configures a static route over which to reverse path forward from the access router to the other end of the tunnel.
Step 12	**end** **Example:** `Router(config)# end`	(Optional) Ends the current configuration session and returns to privileged EXEC mode.
Step 13	**show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*] **Example:** `Router# show ip mroute`	(Optional) Displays the contents of the IP multicast routing (mroute) table.
Step 14	**show ip rpf** {*source-address* | *source-name*} [**metric**] **Example:** `Router# show ip rpf 10.2.3.4`	(Optional) Displays how IP multicast routing does RPF.

Configuration Examples for Tunneling to Connect Non-IP Multicast Areas

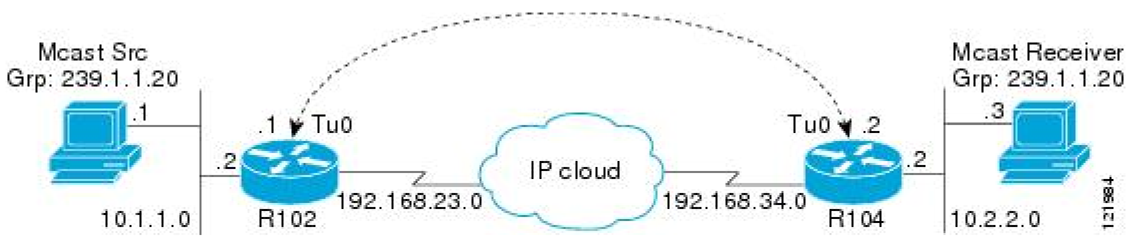
Tunneling to Connect Non-IP Multicast Areas Example

The following example also appears online at:

http://www.cisco.com/en/US/tech/tk828/tk363/technologies_configuration_example09186a00801a5aa2.shtml

In the figure below, the multicast source (10.1.1.1) is connected to R102 and is configured for multicast group 239.1.1.20. The multicast receiver (10.2.2.3) is connected to R104 and is configured to receive multicast packets for group 239.1.1.20. Separating R102 and R104 is an IP cloud, which is not configured for multicast routing.

Figure 18: Tunnel Connecting Non-IP Multicast Areas



A tunnel is configured between R102 to R104 sourced with their loopback interfaces. The **ip pim sparse-dense-mode** command is configured on tunnel interfaces and multicast-routing is enabled on R102 and R104. Sparse-dense mode configuration on the tunnel interfaces allows sparse-mode or dense-mode packets to be forwarded over the tunnel depending on rendezvous point (RP) configuration for the group.

Note For dense mode--With PIM dense mode configured over the tunnel, an **ip mroute 10.1.1.0 255.255.255.0 tunnel 0** command is configured on R104 to ensure a successful RPF for multicast source address 10.1.1.1. Incoming (10.1.1.1, 239.1.1.20) multicast packets over Tunnel0 (Tu0) are checked for Reverse Path Forwarding (RPF) using this mroute statement. After a successful check, the multicast packets are forwarded to outgoing interface list (OIL) interfaces.

Note For sparse mode--With PIM sparse mode configured over the tunnel, ensure that the following points are addressed:

- For a successful RPF verification of multicast traffic flowing over the shared tree (*,G) from RP, an **ip mroute rp-address nexthop** command needs to be configured for the RP address, pointing to the tunnel interface.

Assuming R102 to be the RP (RP address 2.2.2.2) in this case, the mroute would be the **ip mroute 2.2.2.2 255.255.255.255 tunnel 0**command, which ensures a successful RPF check for traffic flowing over the shared tree.

- For a successful RPF verification of multicast (S,G) traffic flowing over the Shortest Path Tree (SPT), an **ip mroute source-address nexthop** command needs to be configured for the multicast source, pointing to the tunnel interface.

In this case, when SPT traffic is flowing over tunnel interface an **ip mroute 10.1.1.0 255.255.255.0 tunnel 0** command is configured on R104 to ensure a successful RPF verification for incoming (10.1.1.1, 239.1.1.20) multicast packets over the Tunnel 0 interface.

R102#

```
version 12.2
hostname r102
ip subnet-zero
no ip domain-lookup
!--- It stops IP domain lookup, which improves the show command response time.
!
ip multicast-routing
!--- Enables IP multicast routing.
!
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!--- Tunnel Source interface.
!
interface Tunnel0
!--- Tunnel interface configured for PIM and carrying multicast packets to R104.
 ip address 192.168.24.1 255.255.255.252
 ip pim sparse-dense-mode
 tunnel source Loopback0
 tunnel destination 4.4.4.4
!
interface Ethernet0/0
!--- Interface connected to Source.
ip address 10.1.1.2 255.255.255.0
 ip pim sparse-dense-mode
!
interface Serial8/0
 ip address 192.168.23.1 255.255.255.252
!--- Note IP PIM sparse-dense mode is not configured on Serial interface.
!
router ospf 1
 log-adjacency-changes
 network 2.2.2.2 0.0.0.0 area 0
 network 10.1.1.0 0.0.0.255 area 0
 network 192.168.23.0 0.0.0.255 area 0
!
ip classless
ip pim bidir-enable
!
line con 0
line aux 0
line vty 0 4
 login
!
end
```

R104#

```
version 12.2
!
hostname r104
```

```
!
ip subnet-zero
no ip domain-lookup
!--- It stops IP domain lookup, which improves the show command response time.
!
ip multicast-routing
!--- Enables IP multicast routing.
!
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
!--- Tunnel Source interface.
!
interface Tunnel0
 ip address 192.168.24.2 255.255.255.252
!--- Tunnel interface configured for PIM and carrying multicast packets.
ip pim sparse-dense-mode
 tunnel source Loopback0
 tunnel destination 2.2.2.2
!
interface Ethernet0/0
 ip address 10.2.2.2 255.255.255.0
 ip pim sparse-dense-mode
!
interface Serial9/0
 ip address 192.168.34.1 255.255.255.252
!--- Note IP PIM sparse-dense mode is not configured on Serial interface.
!
!
router ospf 1
 log-adjacency-changes
 network 4.4.4.4 0.0.0.0 area 0
 network 10.2.2.0 0.0.0.255 area 0
 network 192.168.34.0 0.0.0.255 area 0
!
ip classless
no ip http server
ip pim bidir-enable
ip mroute 10.1.1.0 255.255.255.0 Tunnel0
!--- This Mroute ensures a successful RPF check for packets flowing from the source.
!--- 10.1.1.1 over Shared tree in case of Dense more and SPT in case of Sparse mode.
!
ip mroute 2.2.2.2 255.255.255.255 tunnel 0
!--- This Mroute is required for RPF check when Sparse mode multicast traffic is
!--- flowing from RP (assuming R102 with 2.2.2.2 as RP)  towards receiver via tunnel
!--- before the SPT switchover.
line con 0
line aux 0
line vty 0 4
 login
!
end
```

Additional References

Related Documents

Related Topic	Document Title
IP multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing standards has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Tunneling to Connect Non-IP Multicast Areas

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 8

Automatic Multicast Tunneling

Automatic Multicast Tunneling (AMT) provides a method to tunnel multicast data over a unicast network. The tunneling is performed between AMT relays and AMT gateways, using User Datagram Protocol (UDP) encapsulation. AMT enables service providers and their customers to participate in delivering multicast traffic even in the absence of end-to-end multicast connectivity.

Restrictions for Automatic Multicast Tunneling

- AMT tunnel only support PIM passive mode on AMT Gateway side.
- AMT only support SSM modes for PIM signaling.
- AMT only support ipv4 transport for tunnel.
- For AMT, auto-rp is invalid between relay and gateway.

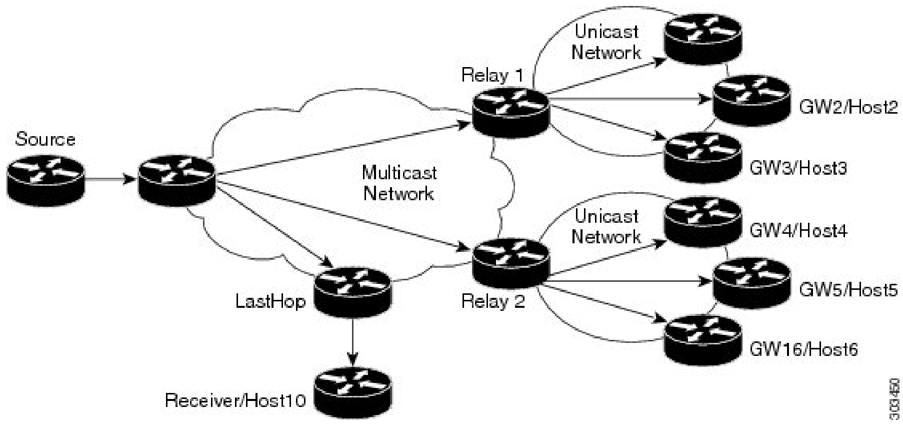
Information About Automatic Multicast Tunneling

Overview

The multicast source sends traffic to the first-hop. Multicast traffic flows through the network until it reaches the last-hop (receivers) or AMT relays. AMT Relay is a multicast router configured to support transit routing between nonmulticast capable internetwork and the native multicast infrastructure.

The following diagram provides a sample AMT network where Relay1 and Relay2 are two AMT relays, which encapsulate the traffic into AMT tunnels, and send one copy to each of the AMT gateways.

Figure 19: Automatic Multicast Tunneling (AMT)



Automatic Multicast Tunneling Message Exchanges

The AMT protocol defines seven message types for control and encapsulation. The message exchanges happen in the following sequence:

1. Relay Discovery—Gateway sends an AMT discovery message to an anycast address that represents the AMT relay.
2. Relay Advertisement—Relay responds with an advertisement message, which includes the relay's unique IP address.
3. Relay Request—Gateway sends an AMT Request message to the relay using the unique IP address as the destination, along with a nonce to be used for security.
4. Membership Query—Relay responds with an AMT query that includes the nonce from the AMT request and an opaque security code.
5. Membership Update—Gateway responds with a membership update that includes an encapsulated IGMPv3/MLDv2 packet.
6. Teardown—Gateway sends a message to stop the delivery of multicast data messages requested in an earlier membership update message.
7. After validation the Relay establishes the AMT Tunnel and starts sending multicast traffic [Type 6]. Any further (S,G) uses the same Request/Query/Update - three-way handshake because the tunnel is already established.

AMT Tunnel and Traffic Types

The multicast traffic carried in the AMT tunnel may be IPv4 or IPv6. The AMT tunnel may be setup with IPv4 or IPv6 endpoints, thereby providing the following possibilities.

- IPv4-in-IPv4—IPv4 multicast traffic carried over an IPv4 tunnel
- IPv6-in-IPv4—IPv6 multicast traffic carried over an IPv4 tunnel
- IPv6-in-IPv6—IPv6 multicast traffic carried over an IPv6 tunnel
- IPv4-in-IPv6—IPv4 multicast traffic carried over an IPv6 tunnel

Note In Cisco IOS XE Release 3.15S, AMT supports IPv4-in-IPv4 and IPv6-in-IPv4 only.

Advantages of Automatic Multicast Tunneling

- Simplicity—Instead of incurring the overhead of manually provisioning, establishing and maintaining GRE tunnels between two locations, the receiving network simply sends AMT advertisements to a well-known any-cast prefix. The rest of the tunnel establishment process is done automatically without the need for additional configuration.
- Resiliency—Because the relay discovery uses an any-cast address, gateways automatically find the closest relay. If that relay becomes unavailable or unreachable, the routing table reconverges on the next closest relay.
- Efficiency—AMT allows transit routers to perform flow-based load balancing for more efficient link utilization.

Automatic Multicast Tunneling supports IPv4 transport for tunnel and also support for IPv4 multicast traffic & IPv6 multicast traffic. You can also configure AMT relay-only, gateway-only and relay-gateway coexisting modes.

Prerequisites for AMT

- AMT relay and gateway tunnel requires an interface IP address. However the interface IP addresses do not need to be unique. You can configure the same addresses for the tunnel source address using the **ip unnumbered** command.
- You must configure IGMP version 3 on the tunnel interfaces for the SSM to work across tunnels.
- The tunnel source port and tunnel destination port must be configured to achieve a valid AMT configuration.

Configuration Recommendations for AMT

The tunnel source interface address should be an interface that is set up to be reachable from receivers under all instances. You can use a physical interface address, but it can cause the tunnel to go down if that physical interface is down. A loopback interface is recommended to sustain availability. You do not need a separate loopback address, you can reuse the loopback interface that you usually would have to carry router and router-ID IP address (usually Loopback 0).

IP PIM passive is the recommended and supported mode on AMT interfaces to enable IP multicast routing for AMT tunnel interfaces. This is recommended since no PIM messages (only AMT/IGMP messages) will be sent or received via the AMT tunnel.

On an AMT relay, you need only one tunnel interface to which all gateways can connect. Therefore the interface is a multipoint interface. Every gateway interface can only connect to one relay, but you can configure multiple tunnel interfaces.

Even though AMT tunnels (relay and gateway) only support IPv4 tunnels, the IPv4 tunnels can carry both IPv4 and IPv6 simultaneously. You can configure the AMT for the version(s) of IP that you need.

If you want to set up a redundant AMT relay, your AMT relay address configured on the gateway should be any IP address that you set up as an anycast address on your relays. If you do not plan to use redundant (Anycast) Relay, you can use the relays tunnel source address on the gateways.

If you shut down the anycast Loopback interface, the AMT relay would not accept new AMT gateway tunnel requests. IP routing for the anycast address would only point to any other relays active with the same address and new requests would therefore go to those relays. Any AMT gateways already connected to this relay would stay on this relay, because they already are using the relays interface address.

How to Configure Automatic Multicast Tunneling

Enabling and Configuring Automatic Multicast Tunneling on a Relay

Note Switch-over from active relay to backup relay can take more than 5 minutes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip address** *ip-address mask*
5. **no ip redirects**
6. **ip pim sparse-mode**
7. **ip igmp version** *version number*
8. **ipv6 enable**
9. **tunnel source** *interface-type interface-number*
10. **tunnel mode udp multipoint**
11. **tunnel dst-port dynamic**
12. **tunnel src-port dynamic**
13. **amt relay traffic** {**ip** | **ipv6**}
14. **exit**
15. **ip multicast-routing distributed**
16. **ipv6 multicast-routing**
17. **ip pim ssm** {*default* | *range access-list*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 11`	Specifies the interface tunnel number and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# 11.1.1.1 255.255.255.0`	Configures the IP address on the interface.
Step 5	**no ip redirects** **Example:** `Device(config-if)# no ip redirects`	Disables sending ICMP Redirect messages.
Step 6	**ip pim sparse-mode** **Example:** `Device(config-if)# ip pim sparse-mode`	Enable PIM sparse-mode operation.
Step 7	**ip igmp version** *version number* **Example:** `Device(config-if)# ip igmp version 3`	Specifies IGMP version.
Step 8	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 on interface.
Step 9	**tunnel source** *interface-type interface-number* **Example:** `Device(config-if)# tunnel source loopback 0`	Configures tunnel source as a loopback interface.
Step 10	**tunnel mode udp multipoint** **Example:** `Device(config-if)# tunnel mode udp multipoint`	Specifies the UDP encapsulation protocol.
Step 11	**tunnel dst-port dynamic** **Example:** `Device(config-if)# tunnel dst-port dynamic`	Specifies the tunnel destination port.
Step 12	**tunnel src-port dynamic** **Example:** `Device(config-if)# tunnel src-port dynamic`	Specifies the tunnel source port.

	Command or Action	Purpose
Step 13	**amt relay traffic** {**ip** | **ipv6**} **Example:** `Device(config-if)# amt relay traffic ipv6`	Enables IPv4 or IPv6 traffic on AMT relay interface.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 15	**ip multicast-routing distributed** **Example:** `Device(config)# ip multicast-routing distributed`	Enables Multicast Distributed Switching (MDS).
Step 16	**ipv6 multicast-routing** **Example:** `Device(config)# ipv6 multicast-routing`	Enables multicast routing using Protocol Independent Multicast (PIM) and Multicast Listener Discovery (MLD).
Step 17	**ip pim ssm** {*default* | *range access-list*} **Example:** `Device(config)# ip pim ssm default`	Specifies the Source Specific Multicast (SSM) range of IP multicast addresses.

Enabling and Configuring Automatic Multicast Tunneling on Gateway

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip address** *ip-address mask*
5. **ip pim passive**
6. **ip igmp version** *version number*
7. **ipv6 enable**
8. **ipv6 pim passive**
9. **tunnel source** *interface-type interface-number*
10. **tunnel mode udp ip**
11. **tunnel destination dynamic**
12. **tunnel dst-port dynamic**
13. **tunnel src-port dynamic**
14. **amt gateway traffic** {**ip** | **ipv6**}
15. **amt gateway relay-address** *IP address*
16. **exit**
17. **ip multicast-routing distributed**
18. **ipv6 multicast-routing**
19. **ip pim ssm** {*default* | *range access-list*}

20. **ipv6 multicast pim-passive-enable**
21. **ip route** *ip-address interface-type interface-number* [**multicast**]
22. **ipv6 route** *ipv6-prefix/prefix-length interface-type interface-number* [**multicast**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 11`	Specifies the interface tunnel number and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# 11.1.1.1 255.255.255.0`	Configures the IP address on the interface.
Step 5	**ip pim passive** **Example:** `Device(config-if)# ip pim passive`	Enables IP PIM passive mode operation.
Step 6	**ip igmp version** *version number* **Example:** `Device(config-if)# ip igmp version 3`	Specifies IGMP version.
Step 7	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 on interface.
Step 8	**ipv6 pim passive** **Example:** `Device(config-if)# ipv6 pim passive`	Enables IPv6 PIM passive mode operation.
Step 9	**tunnel source** *interface-type interface-number* **Example:** `Device(config-if)# tunnel source loopback 0`	Configures tunnel source as a loopback interface.
Step 10	**tunnel mode udp ip** **Example:** `Device(config-if)# tunnel mode udp ip`	Specifies the UDP encapsulation protocol.

	Command or Action	Purpose
Step 11	**tunnel destination dynamic** **Example:** `Device(config-if)# tunnel destination dynamic`	Applies the tunnel destination address dynamically to the tunnel interface.
Step 12	**tunnel dst-port dynamic** **Example:** `Device(config-if)# tunnel dst-port dynamic`	Applies the tunnel destination port dynamically.
Step 13	**tunnel src-port dynamic** **Example:** `Device(config-if)# tunnel src-port dynamic`	Applies the tunnel source port dynamically.
Step 14	**amt gateway traffic** {**ip** | **ipv6**} **Example:** `Device(config-if)# amt gateway traffic ipv6`	Enables IPv4 or IPv6 traffic on AMT gateway interface.
Step 15	**amt gateway relay-address** *IP address* **Example:** `Device(config-if)# amt gateway relay-address 172.16.0.0`	Specifies the destination IP address of AMT discovery.
Step 16	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 17	**ip multicast-routing distributed** **Example:** `Device(config)# ip multicast-routing distributed`	Enables Multicast Distributed Switching (MDS).
Step 18	**ipv6 multicast-routing** **Example:** `Device(config)# ipv6 multicast-routing`	Enables multicast routing using Protocol Independent Multicast (PIM) and Multicast Listener Discovery (MLD).
Step 19	**ip pim ssm** {*default* | *range access-list*} **Example:** `Device(config)# ip pim ssm default`	Specifies the Source Specific Multicast (SSM) range of IP multicast addresses.
Step 20	**ipv6 multicast pim-passive-enable** **Example:** `Device(config)# ipv6 multicast pim-passive-enable`	Enables passive PIM operation.
Step 21	**ip route** *ip-address interface-type interface-number* [**multicast**] **Example:**	Specifies the IP address along with interface type, interface number, and multicast route.

	Command or Action	Purpose
	`Device(config)# ip route 101.0.0.2 255.255.255.255 tunnel10 multicast`	
Step 22	**ipv6 route** *ipv6-prefix/prefix-length interface-type interface-number* [**multicast**] **Example:** `Device(config)# ipv6 route 2011::101:0:0:2/128 Tunnel10 multicast`	Specifies the IPv6 prefix and length along with interface type, interface number, and multicast route (route only usable by multicast).

Displaying and Verifying AMT Configuration

SUMMARY STEPS

1. **ping** *3.3.3.3* **source** *5.5.5.5*
2. **show ip igmp membership**
3. **show ip mroute**
4. **show ip rpf***10.3.3.1*

DETAILED STEPS

Step 1 **ping** *3.3.3.3* **source** *5.5.5.5*

Example:

```
Device# ping 3.3.3.3 source 5.5.5.5
   Type escape sequence to abort.
   Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
   Packet sent with a source address of 5.5.5.5
   !!!!!
   Success rate is 100 percent (5/5), round-trip min/avg/max = 6/6/8 ms
```

Ping the relay tunnel source address from the gateway, using the gateways tunnel source address and the gateway address (if both are different as the example).

Step 2 **show ip igmp membership**

Example:

```
Device# show ip igmp membership
  ...
   Channel/Group                  Reporter        Uptime   Exp.  Flags  Interface
  /*,232.2.3.4                    0.0.0.0         00:03:02 stop  2MA    Lo0
   10.3.3.1,232.2.3.4                             00:03:02 stop  A      Lo0
```

Note The command is executed at destination gateway.

You can use the **ip igmp static-group** command, you create an IGMP membership request on the interface. You can do this on any physical interface configured for IP multicast, but if there are multiple routers attached to the interface, then the AMT gateway router may not become the PIM-DR and therefore not join to the multicast traffic. It can therefore be easier to put the join on an existing loopback interface and also enable it for IP multicast (IP PIM passive).

Step 3 **show ip mroute**

Example:

```
Device# show ip mroute
  ...
  (10.3.3.1, 232.2.3.4), 00:01:53/00:01:08, flags: sTI
    Incoming interface: Tunnel1, RPF nbr 0.0.0.0, Mroute
    Outgoing interface list:
      Loopback0, Forward/Sparse-Dense, 00:01:51/00:01:08
```

Note The command is executed at destination gateway.

The AMT gateway will join the multicast traffic towards the AMT relay if the mroute shows the incoming interface as the Tunnel interface and when it is joined for example, when there are one or more outgoing interfaces.

When the command is executed on the source gateway, the mroute state on the relay will show the relay interface as an adjacency in the outgoing interface list. Because there is one interface for all joining relays, each outgoing copy across the AMT relay interface is identified by the AMT relay interface, the IP address of the gateway and the UDP port number of the relay as seen by the relay. If there are multiple gateways in a home behind a NAT/PAT (single IP address to the internet), there would be multiple adjacencies shown with the same gateway IP address, but different UDP ports. If state is not built as expected, you can use **debug ip igmp** to troubleshoot it. There are no additional AMT debugs for the AMT/IGMP joins, instead those are all part of IGMP debugs.

Step 4 **show ip rpf***10.3.3.1*

Example:

```
Device#
RPF information for ? (10.3.3.1)
  RPF interface: Tunnel1
  RPF neighbor: ? (0.0.0.0)
  RPF route/mask: 10.3.3.0/24
  RPF type: multicast (static)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base
```

Note The command is executed at destination gateway.

Verifies Multicast RPF.

If the output is not showing the desired AMT gateway tunnel interface as the RPF interface, then your routing configuration for IP multicast into the AMT tunnel is not set up correctly. If you have only one AMT gateway interface, you can use a static mroute. If you have multiple AMT tunnel for different relays, you need to configure the source prefixes (and RP-addresses when you use PIM-SM) towards their respective AMT tunnel.

For IP multicast tree to flow correctly via the AMT tunnels, you must first check if the IP multicast state is correctly built from gateway to relay. This primarily means that the (S,G) join for a source in PIM-SSM/PIM-SM or the (*,G) join to the RP address in PIM-SM need to RPF towards the desired AMT gateway tunnel.

Displaying and Verifying AMT Relay Configuration

SUMMARY STEPS

1. **enable**
2. **show ip amt tunnel**
3. **show ip mroute section** [**group-address**]

4. **show ipv6 mroute section** [**group-address**]
5. **show ip mfib section** [**group-address**]
6. **show ipv6 mfib section** [**group-address**]
7. **show platform software ip rp active mfib section** [**group-address**]
8. **show platform software ipv6 rp active mfib section** [**group-address**]
9. **show platform software mlist rp active index** *multicast-index-number*
10. **show platform software adjacency rp active index** *platform-allocated-index-value*
11. **show ip interface brief**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ip amt tunnel**

Example:

```
Device# show ip amt tunnel

AMT Relay tunnel:
    Local address       UDP port
      11.11.11.11       2268 (0x8DC )
    Remote address                      Expire time
      33.33.33.33       59464(0xE848)   00:03:07
    Connected to 1 Gateway

  Total active Gateways: 1
```

Displays AMT relay configuration.

Step 3 **show ip mroute section** [**group-address**]

Example:

```
Device# show ip mroute section 232.1.1.1

(101.0.0.2, 232.1.1.1), 2d00h/00:02:18, flags: sTI
  Incoming interface: GigabitEthernet0/0/4, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel10, 33.33.33.33, UDP port 59464, Forward/Sparse, 2d00h/00:02:18
```

Displays information about sparse mode routes in the IP multicast routing (mroute) table for the specified multicast group.

Step 4 **show ipv6 mroute section** [**group-address**]

Example:

```
Device# show ipv6 mroute section 232.1.1.1

(2011::101:0:0:2, FF3F::232:1:1:1), 2d00h/never, flags: sTI
  Incoming interface: GigabitEthernet0/0/4
  RPF nbr: 2011::101:0:0:2
```

```
  Immediate Outgoing interface list:
    Tunnel10, AMT NH 33.33.33.33, UDP Port 59464, Forward, 2d00h/never
```

Displays information about sparse mode routes in the IPv6 mroute table for the specified multicast group.

Step 5 **show ip mfib section** [**group-address**]

Example:

```
Device# show ip mfib section 232.1.1.1

(101.0.0.2,232.1.1.1) Flags: HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   NA/NA/NA/NA, Other: NA/NA/NA
   GigabitEthernet0/0/4 Flags: A
   Tunnel10, AMT Encap 33.33.33.33, UDP Port:59464 Flags: F NS
     Pkts: 0/0
```

Displays the status of entries and interfaces in the IPv4 Multicast Forwarding Information Base (MFIB) for the specified multicast group.

Step 6 **show ipv6 mfib section** [**group-address**]

Example:

```
Device# show ipv6 mfib section 232.1.1.1

(2011::101:0:0:2,FF3F::232:1:1:1) Flags: HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   NA/NA/NA/NA, Other: NA/NA/NA
   GigabitEthernet0/0/4 Flags: A
   Tunnel10, AMT Encap 33.33.33.33, UDP Port:59464 Flags: F NS
     Pkts: 0/0
```

Displays the status of entries and interfaces in the IPv6 Multicast Forwarding Information Base (MFIB) for the specified multicast group.

Step 7 **show platform software ip rp active mfib section** [**group-address**]

Example:

```
Device# show platform software ip rp active mfib section 232.1.1.1

232.1.1.1, 101.0.0.2/64 --> OBJ_INTF_LIST (0x5a)
                Obj id: 0x5a, Flags:
                OM handle: 0x421712c4
```

Displays platform software IPv4 MFIB route processor information for the specified multicast group.

Step 8 **show platform software ipv6 rp active mfib section** [**group-address**]

Example:

```
Device# show platform software ipv6 rp active mfib section 232.1.1.1

ff3f::232:1:1:1, 2011::101:0:0:2/256 --> OBJ_INTF_LIST (0x5b)
                Obj id: 0x5b, Flags:
                OM handle: 0x42171d24
```

Displays platform software IPv6 MFIB route processor information for the specified multicast group.

Step 9 **show platform software mlist rp active index** *multicast-index-number*

Example:

```
Device# show platform software mlist rp active index 0x5a

OCE              Type              OCE Flags        Interface
--------------------------------------------------------------------------------------
```

```
0x57            OBJ_ADJACENCY    NS, F            Tunnel10
0xf80000c1      OBJ_ADJACENCY    A                GigabitEthernet0/0/4
```

Displays platform software route processor information for the specified multicast list index.

Step 10 **show platform software adjacency rp active index** *platform-allocated-index-value*

Example:

```
Device# show platform software adjacency rp active index 0x57

Number of adjacency objects: 22

Adjacency id: 0x57 (87)
  Interface: Tunnel10, IF index: 20, Link Type: MCP_LINK_IP
  Encap: 45:0:0:0:0:0:0:0:ff:11:63:95:b:b:b:b:21:21:21:21:8:dc:e8:48:0:0:0:0:6:0
  Encap Length: 30, Encap Type: MCP_ET_TUNNEL, MTU: 1470
  Flags: no-l3-inject
  Incomplete behavior type: None
  Fixup: gre
  Fixup_Flags_2: pmip-udp
  Nexthop addr: 33.33.33.33
  IP FRR MCP_ADJ_IPFRR_NONE 0
  OM handle: 0x4217013c
```

Displays platform software adjacency route processor information for the specified platform allocated index.

Step 11 **show ip interface brief**

Example:

```
Device(source gateway)# show ip interface brief

  Interface              IP-Address      OK? Method Status                Protocol
    Tunnel1                3.3.3.3         YES TFTP   up                    up
    ....
```

The above output indicates that the relay has no connection from a gateway. The AMT relay interface will be up after it is correctly configured. If the output displays Protocol "down", you must check your configuration. If the interface command shows **administratively down**, you have not configured the "no shut" in the interface.

Displays errors in AMT gateway configuration.

Displaying and Verifying AMT Gateway Configuration

SUMMARY STEPS

1. **enable**
2. **show ip amt tunnel**
3. **show ip mroute section** [**group-address**]
4. **show ipv6 mroute section** [**group-address**]
5. **show ip mfib section** [**group-address**]
6. **show ipv6 mfib section** [**group-address**]
7. **show platform software adjacency rp active index** *platform-allocated-index-value*
8. **show ip interface brief**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ip amt tunnel**

Example:

```
Device(destination gateway)# show ip amt tunnel

AMT Gateway tunnel Tunnel1:
       Local address       UDP port
            5.5.5.5           54358(0xD456)
       Remote address
            3.3.3.3           2268 (0x8DC )

  Total active Relays: 1
```

```
Device(source gateway)# show ip amt tunnel
AMT Relay tunnel:
      Local address       UDP port
        3.3.3.3           2268 (0x8DC )
      Remote address                        Expire time
        5.5.5.5           54358(0xD456)   00:03:53
      Connected to 1 Gateway
```

The above output is displayed if both relay and gateway is working correctly.

Displays AMT gateway configuration.

Step 3 **show ip mroute section** [**group-address**]

Example:

```
Device# show ip mroute section 232.1.1.1

(101.0.0.2, 232.1.1.1), 2d00h/00:02:54, flags: sTI
  Incoming interface: Tunnel10, RPF nbr 0.0.0.0, Mroute
  Outgoing interface list:
    GigabitEthernet0/0/4, Forward/Sparse, 2d00h/00:02:54
```

Displays information about the IP multicast routing (mroute) table for the specified multicast group.

Step 4 **show ipv6 mroute section** [**group-address**]

Example:

```
Device# show ipv6 mroute section 232.1.1.1

(2011::101:0:0:2, FF3F::232:1:1:1), 2d00h/never, flags: sTI
  Incoming interface: Tunnel10
  RPF nbr: ::
  Immediate Outgoing interface list:
    GigabitEthernet0/0/4, Forward, 2d00h/never
```

Displays information about the IPv6 mroute table for the specified multicast group.

Step 5 **show ip mfib section** [**group-address**]

Example:

```
Device# show ip mfib section 232.1.1.1

(101.0.0.2,232.1.1.1) Flags: HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   NA/NA/NA/NA, Other: NA/NA/NA
   Tunnel10 Flags: A
   GigabitEthernet0/0/4 Flags: F NS
     Pkts: 0/0
```

Displays the entries and interfaces in the IPv4 Multicast Forwarding Information Base (MFIB) for the specified multicast group.

Step 6 **show ipv6 mfib section** [**group-address**]

Example:

```
Device# show ipv6 mfib section 232.1.1.1

(2011::101:0:0:2,FF3F::232:1:1:1) Flags: HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   NA/NA/NA/NA, Other: NA/NA/NA
   Tunnel10 Flags: A
   GigabitEthernet0/0/4 Flags: F NS
     Pkts: 0/0
```

Displays the entries and interfaces in the IPv6 Multicast Forwarding Information Base (MFIB) for the specified multicast group.

Step 7 **show platform software adjacency rp active index** *platform-allocated-index-value*

Example:

```
Device# show platform software adjacency rp active index 0x57

Number of adjacency objects: 19
Adjacency id: 0xf8000126 (4160749862)
  Interface: Tunnel10, IF index: 18, Link Type: MCP_LINK_IP
  Encap: 45:0:0:0:0:0:0:0:ff:11:63:95:21:21:21:21:b:b:b:b:e8:48:8:dc:0:0:0:0:6:0
  Encap Length: 30, Encap Type: MCP_ET_TUNNEL, MTU: 1470
  Incomplete behavior type: None
  Fixup: gre
  Fixup_Flags_2: pmip-udp
  IP FRR MCP_ADJ_IPFRR_NONE 0
  OM handle: 0x4216aaf4
```

Displays platform software adjacency route processor information for the specified platform allocated index.

Step 8 **show ip interface brief**

Example:

```
Device(destination gateway)# show ip interface brief

  Interface              IP-Address      OK? Method Status                Protocol
  Tunnel1                5.5.5.5         YES TFTP   up                    down
```

The Protocol: down displayed in the above output indicates that the gateway has no connection to the relay. If the interface command shows **administratively down**, you have not configured the "no shut" in the interface. If the command displays UNKNOWN but Protocol shows "up", then the gateway did have a connection to the relay hat was terminated.

Note It takes a few minutes before a gateway will consider a relay to be unreachable after not receiving packets from it. The gateway will then revert trying to reach the anycast address to find a new relay (in case the existing relay is down).

Displays errors in AMT gateway configuration.

Configuration Examples for Automatic Multicast Tunneling

Example: AMT Relay Configuration

The following example shows how to configure AMT relay:

```
enable
configure terminal
interface Tunnel10
 ip address 11.1.1.1 255.255.255.0
 no ip redirects
 ip pim sparse-mode
 ip igmp version 3
 ipv6 enable
 tunnel source Loopback0
 tunnel mode udp multipoint
 tunnel dst-port dynamic
 tunnel src-port dynamic
 amt relay traffic ip
 amt relay traffic ipv6
exit

ip multicast-routing distributed
ipv6 multicast-routing
ip pim ssm default
end
```

Example: AMT Gateway Configuration

The following example shows how to configure AMT gateway:

```
enable
configure terminal
interface Tunnel10
 ip address 33.1.1.1 255.255.255.0
 ip pim passive
 ip igmp version 3
 ipv6 enable
 ipv6 pim passive
 tunnel source Loopback0
 tunnel mode udp ip
 tunnel destination dynamic
 tunnel dst-port dynamic
 tunnel src-port dynamic
 amt gateway traffic ip
 amt gateway traffic ipv6
 amt gateway relay-address 167.3.0.1
exit

ip multicast-routing distributed
ipv6 multicast-routing
ip pim ssm default
```

```
ipv6 multicast pim-passive-enable
ip route 101.0.0.2 255.255.255.255 Tunnel10 multicast
ipv6 route 2011::101:0:0:2/128 Tunnel10 multicast
end
```

Additional References for Automatic Multicast Tunneling

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP Multicast commands	Cisco IOS IP Multicast Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Automatic Multicast Tunneling

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 9

BFD Support for Multicast (PIM)

This module contains information for enabling the Bidirectional Forwarding Detection (BFD) detection protocol on Protocol Independent Multicast (PIM) interfaces in your IP v4 and IPv6 network. Enabling PIM BFD enables PIM to use BFD's quicker adjacent system failure detection and avoid the slower query-interval in its own detection mechanisms.

Restrictions for BFD Support for Multicast (PIM)

- This feature is not supported for Multicast VPN (MVPN).
- This feature is supported only on interfaces on which both PIM and BFD are supported.

Information About BFD Support for Multicast (PIM)

PIM BFD

Bidirectional Forwarding Detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols and independent of the higher layer protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning is easier and reconvergence time is consistent and predictable.

Protocol Independent Multicast (PIM) uses a hello mechanism for discovering new neighbors and for detecting failures between adjacent nodes. The minimum failure detection time in PIM is 3 times the PIM Query-Interval. To enable faster failure detection, the rate at which a PIM Hello message is transmitted on an interface is configurable. However, lower intervals increase the load on the protocol and can increase CPU and memory utilization and cause a system-wide negative impact on performance. Lower intervals can also cause PIM neighbors to expire frequently as the neighbor expiry can occur before the hello messages received from those neighbors are processed.

The BFD Support for Multicast (PIM) feature, also known as PIM BFD, registers PIM as a client of BFD. PIM can then utilize BFD to initiate a session with an adjacent PIM node to support BFD's fast adjacency failure detection in the protocol layer. PIM registers just once for both PIM and IPv6 PIM.

At PIMs request (as a BFD client), BFD establishes and maintains a session with an adjacent node for maintaining liveness and detecting forwarding path failure to the adjacent node. PIM hellos will continue to be exchanged between the neighbors even after BFD establishes and maintains a BFD session with the neighbor. The behavior of the PIM hello mechanism is not altered due to the introduction of this feature.

Although PIM depends on the Interior Gateway Protocol (IGP) and BFD is supported in IGP, PIM BFD is independent of IGP's BFD.

How to Configure BFD Support for Multicast (PIM)

Enabling BFD PIM on an Interface

Before you begin

- For IPv4 networks, IP multicast must be enabled and Protocol Independent Multicast (PIM)must be configured on the interface. For information, see the “Configuring Basic IP Multicast in IPv4 Networks” module of the *IP Multicast: PIM Configuration Guide*.
- For IPv6 networks, IPv6 multicast must be enabled and Protocol Independent Multicast (PIM) must be configured on the interface. For information, see the *IP Multicast: PIM Configuration Guide*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface***type number*
4. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*
5. Use one of the following:
   - **ip pim bfd**
   - **ipv6 pim bfd**
6. Repeat the preceding steps for each interface to be enabled for BFD PIM.
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface***type number* **Example:** `Device(config)# interface fastethernet 1/6`	Enters interface configuration mode for the specified interface.
Step 4	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **Example:** `Device(config-if)# bfd interval 500 min_rx 500 multiplier 5`	Enables BFD on the interface.

Step 5	Use one of the following:
	• **ip pim bfd**
	• **ipv6 pim bfd**
	Example:
	`Device(config-if)# ip pim bfd`
	`Device(config-if)# ipv6 pim bfd`
		Enables PIM BFD on the interface.

Step 6	Repeat the preceding steps for each interface to be enabled for BFD PIM.	
Step 7	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode .

Configuration Examples for BFD Support for Multicast (PIM)

Additional References for BFD Support for Multicast (PIM)

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference

Related Topic	Document Title
Bidirectional Forwarding Detection (BFD) detection protocol	*IP Routing BFD Configuration Guide*
Protocol Independent Multicast (PIM) in an IPv4 network	"Configuring Basic IP Multicast in IPv4 Networks" module of the *IP Multicast: PIM Configuration Guide*
PIM in an IPv6 network	*IP Multicast: PIM Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BFD Support for Multicast (PIM)

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 10

HSRP Aware PIM

This module describes how to configure the HSRP Aware PIM feature for enabling multicast traffic to be forwarded through the Hot Standby Router Protocol (HSRP) active router (AR), allowing Protocol Independent Multicast (PIM) to leverage HSRP redundancy, avoid potential duplicate traffic, and enable failover.

Restrictions for HSRP Aware PIM

- HSRP IPv6 is not supported.
- Stateful failover is not supported. During PIM stateless failover, the HSRP group's virtual IP address transfers to the standby router but no mrouting sate information is transferred. PIM listens and responds to state change events and creates mroute states upon failover.
- The maximum number of HSRP groups that can be tracked by PIM on each interface is 16.
- The redundancy priority for a PIM DR must be greater than the configured or default value (1) of the PIM DR priority on any device for which the same HSRP group is enabled or the HSRP Active will fail to win the DR election.
- Dense mode is not supported.
- HSRP address as PIM RP is not supported. HSRP aware PIM is for coordinating PIM DR election and HSRP primary election.

Information About HSRP Aware PIM

HSRP

Hot Standby Router Protocol (HSRP) is a Cisco proprietary redundancy protocol for establishing a fault-tolerant default gateway.

The protocol establishes a framework between network devices in order to achieve default gateway failover if the primary gateway becomes inaccessible. By sharing an IP address and a MAC (Layer 2) address, two or more devices can act as a single virtual router. The members of a virtual router group continually exchange status messages and one device can assume the routing responsibility of another, should it go out of commission for either planned or unplanned reasons. Hosts continue to forward IP packets to a consistent IP and MAC addres,s and the changeover of devices doing the routing is transparent.

HSRP is useful for hosts that do not support a router discovery protocol and cannot switch to a new device when their selected device reloads or loses power. Because existing TCP sessions can survive the failover, this protocol also provides a more transparent recovery for hosts that dynamically choose a next hop for routing IP traffic.

When HSRP is configured on a network segment, it provides a virtual MAC address and an IP address that is shared among a group of devices running HSRP. The address of this HSRP group is referred to as the virtual IP address. One of these devices is selected by the protocol to be the active router (AR). The AR receives and routes packets destined for the MAC address of the group.

HSRP uses a priority mechanism to determine which HSRP configured device is to be the default AR. To configure a device as the AR, you assign it a priority that is higher than the priority of all the other HSRP-configured devices. The default priority is 100, so if you configure just one device to have a higher priority, that device will be the default AR.

Devices that are running HSRP send and receive multicast User Datagram Protocol (UDP)-based hello messages to detect device failure and to designate active and standby devices. When the AR fails to send a hello message within a configurable period of time, the standby device with the highest priority becomes the AR. The transition of packet forwarding functions between devices is completely transparent to all hosts on the network.

You can configure multiple Hot Standby groups on an interface, thereby making fuller use of redundant devices and load sharing.

HSRP is not a routing protocol as it does not advertise IP routes or affect the routing table in any way.

HSRP has the ability to trigger a failover if one or more interfaces on the device fail. This can be useful for dual branch devices each with a single serial link back to the head end. If the serial link of the primary device goes down, the backup device takes over the primary functionality and thus retains connectivity to the head end.

HSRP Aware PIM

Protocol Independent Multicast (PIM) has no inherent redundancy capabilities and its operation is completely independent of Hot Standby Router Protocol (HSRP) group states. As a result, IP multicast traffic is forwarded not necessarily by the same device as is elected by HSRP. The HSRP Aware PIM feature provides consistent IP multicast forwarding in a redundant network with virtual routing groups enabled.

HSRP Aware PIM enables multicast traffic to be forwarded through the HSRP active router (AR), allowing PIM to leverage HSRP redundancy, avoid potential duplicate traffic, and enable failover, depending on the HSRP states in the device. The PIM designated router (DR) runs on the same gateway as the HSRP AR and maintains mroute states.

In a multiaccess segment (such as LAN), PIM DR election is unaware of the redundancy configuration, and the elected DR and HSRP AR may not be the same router. In order to ensure that the PIM DR is always able to forward PIM Join/Prune message towards RP or FHR, the HSRP AR becomes the PIM DR (if there is only one HSRP group). PIM is responsible for adjusting DR priority based on the group state. When a failover occurs, multicast states are created on the new AR elected by the HSRP group and the AR assumes responsibility for the routing and forwarding of all the traffic addressed to the HSRP virtual IP address.

With HSRP Aware PIM enabled, PIM sends an additional PIM Hello message using the HSRP virtual IP addresses as the source address for each active HSRP group when a device becomes HSRP Active. The PIM Hello will carry a new GenID in order to trigger other routers to respond to the failover. When a downstream device receives this PIM Hello, it will add the virtual address to its PIM neighbor list. The new GenID carried in the PIM Hello will trigger downstream routers to resend PIM Join messages towards the virtual address. Upstream routers will process PIM Join/Prunes (J/P) based on HSRP group state.

If the J/P destination matches the HSRP group virtual address and if the destination device is in HSRP active state, the new AR processes the PIM Join because it is now the acting PIM DR. This allows all PIM Join/Prunes to reach the HSRP group virtual address and minimizes changes and configurations at the downstream routers side.

The IP routing service utilizes the existing virtual routing protocol to provide basic stateless failover services to client applications, such as PIM. Changes in the local HSRP group state and standby router responsibility are communicated to interested client applications. Client applications may build on top of IRS to provide stateful or stateless failover. PIM, as an HSRP client, listens to the state change notifications from HSRP and automatically adjusts the priority of the PIM DR based on the HSRP state. The PIM client also triggers communication between upstream and downstream devices upon failover in order to create an mroute state on the new AR.

How to Configure HSRP Aware PIM

Configuring an HSRP Group on an Interface

Before you begin

- IP multicast must already be configured on the device.
- PIM must already be configured on the interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip address** *ip-address mask*
5. **standby** [*group-number*] **ip** [*ip-address* [*secondary*]]
6. **standby** [*group-number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime*

7. **standby** [*group-number*] **priority** *priority*
8. **standby** [*group-number*] **name** *group-name*
9. **end**
10. **show standby** [*type number* [*group*]] [**all** | **brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Device(config)# interface ethernet 0/0`	Specifies an interface to be configured and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.0.0.2 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 5	**standby** [*group-number*] **ip** [*ip-address* [*secondary*]] **Example:** `Device(config-if)# standby 1 ip 192.0.2.99`	Activates HSRP and defines an HRSP group.
Step 6	**standby** [*group-number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime* **Example:** `Device(config-if)# standby 1 timers 5 15`	(Optional) Configures the time between hello packets and the time before other devices declare an HSRP active or standby router to be down.
Step 7	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 120`	(Optional) Assigns the HSRP priority to be used to help select the HSRP active and standby routers.
Step 8	**standby** [*group-number*] **name** *group-name* **Example:** `Device(config-if)# standby 1 name HSRP1`	(Optional) Defines a name for the HSRP group. **Note** We recommend that you always configure the **standby ip name** command when configuring an HSRP group to be used for HSRP Aware PIM.
Step 9	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 10	**show standby** [*type number* [*group*]] [**all** | **brief**] **Example:** `Device# show standby`	Displays HSRP group information for verifying the configuration.

Configuring PIM Redundancy

Before you begin

The HSRP group must already be configured on the interface. See the "Configuring an HSRP Group on an Interface" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip address** *ip-address mask*
5. **ip pim redundancy** *group* **dr-priority** *priority*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Device(config)# interface ethernet 0/0`	Specifies an interface to be configured and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.0.0.2 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 5	**ip pim redundancy** *group* **dr-priority** *priority* **Example:** `Device(config-if)# ip pim redundancy HSRP1 dr-priority 60`	Enables PIM redundancy and assigns a redundancy priority value to the active PIM designated router (DR). • Because HSRP group names are case sensitive, the value of the *group* argument must match the group

	Command or Action	Purpose
		name configured by using the **standby ip name** command. • The redundancy priority for a PIM DR must be greater than the configured or default value (1) of the PIM DR priority on any device for which the same HSRP group is enabled.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for HSRP Aware PIM

Example: Configuring an HSRP Group on an Interface

```
interface ethernet 0/0
 ip address 10.0.0.2 255.255.255.0
 standby 1 ip 192.0.2.99
 standby 1 timers 5 15
 standby 1 priority 120
 standby 1 name HSRP1
 !
!
```

Example: Configuring PIM Redundancy

```
interface ethernet 0/0
 ip address 10.0.0.2 255.255.255.0
 ip pim redundancy HSRP1 dr-priority 60
 !
!
```

Additional References for HSRP Aware PIM

Related Documents

Related Topic	Document Title
IP multicast commands	Cisco IOS IP Multicast Command Reference
HSRP commands	First Hop Redundancy Protocol Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2281	*Cisco Hot Standby Router Protocol (HSRP)*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for HSRP Aware PIM

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 11

VRRP Aware PIM

The Virtual Router Redundancy Protocol (VRRP) eliminates the single point of failure inherent in the static default routed environment. VRRP is an election protocol that dynamically assigns responsibility for one or more virtual routers to the VRRP routers on a LAN, allowing several routers on a multi access link to utilize the same virtual IP address.

VRRP Aware PIM is a redundancy mechanism for the Protocol Independent Multicast (PIM) to interoperate with VRRP. It allows PIM to track VRRP state and to preserve multicast traffic upon fail over in a redundant network with virtual routing groups enabled.

This module explains how to configure VRRP Aware PIM in a network.

Restrictions for VRRP Aware PIM

- Only PIM sparse mode (SM) and source specific multicast (SSM) modes are supported. Bidirectional (BiDir) PIM is not supported.
- PIM interoperability with Hot Standby Router Protocol (HSRP) IPv6 is not supported.
- PIM tracks only one virtual group, either Virtual Router Redundancy Protocol (VRRP) or HSRP, per interface.
- VRRP Aware PIM is not supported on a Transit network. PIM redundancy enabled interface does not support the PIM joining the network from down stream.

Information About VRRP Aware PIM

Overview of VRRP Aware PIM

Virtual Router Redundancy Protocol (VRRP) is a redundancy protocol for establishing a fault-tolerant default gateway. The protocol establishes a framework between network devices in order to achieve default gateway failover if the primary gateway becomes inaccessible.

Protocol Independent Multicast (PIM) has no inherent redundancy capabilities and its operation is completely independent of VRRP group states. As a result, IP multicast traffic is forwarded not necessarily by the same device as is elected by VRRP. The VRRP Aware PIM feature provides consistent IP multicast forwarding in a redundant network with virtual routing groups enabled.

In a multi-access segment (such as LAN), PIM designated router (DR) election is unaware of the redundancy configuration, and the elected DR and VRRP primary router (MR) may not be the same router. In order to ensure that the PIM DR is always able to forward PIM Join/Prune message towards RP or FHR, the VRRP MR becomes the PIM DR (if there is only one VRRP group). PIM is responsible for adjusting DR priority based on the group state. When a fail over occurs, multicast states are created on the new MR elected by the VRRP group and the MR assumes responsibility for the routing and forwarding of all the traffic addressed to the VRRP virtual IP address. This ensures the PIM DR runs on the same gateway as the VRRP MR and maintains mroute states. It enables multicast traffic to be forwarded through the VRRP MR, allowing PIM to leverage VRRP redundancy, avoid potential duplicate traffic, and enable fail over, depending on the VRRP states in the device.

Virtual Router Redundancy Service (VRRS) provides public APIs for a client to communicate with VRRP. VRRP Aware PIM is a feature of VRRS that supports VRRPv3 (unified VRRP) in both IPv4 and IPv6.

PIM, as a VRRS client, uses the VRRS client API to obtain generic First Hop Redundancy Protocol (FHRP) state and configuration information in order to provide multicast redundancy functionalities.

PIM performs the following as a VRRS client:

- Listens to state change and update notification from VRRS server (i.e., VRRP).
- Automatically adjust PIM DR priority based on VRRP state.
- Upon VRRP fail over, PIM receives state change notification from VRRS for the tracked VRRP group and ensures traffic is forwarded through VRRP MR.

How to Configure VRRP Aware PIM

Configuring VRRP Aware PIM

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **fhrp version vrrp** *version*
4. **interface** *type number*

5. **ip address** *address* {*primary* |*secondary*}
6. **vrrp** *group id* **address-family ipv4**
7. **vrrs leader** *group name*
8. **vrrp** *group id* **ip** *ip address*{*primary* |*secondary*}
9. **exit**
10. **interface** *type number*
11. **ip pim redundancy** *group name* **vrrp dr-priority** *priority-value*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**fhrp version vrrp** *version* **Example:** `Device(config)# fhrp version vrrp v3`	Enables the ability to configure VRRPv3 and VRRS.
Step 4	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Specifies an interface to be configured and enters interface configuration mode.
Step 5	**ip address** *address* {*primary* |*secondary*} **Example:** `Device(config-if)# ip address 192.0.2.2`	Specifies a primary or secondary address for the VRRP group.
Step 6	**vrrp** *group id* **address-family ipv4** **Example:** `Device(config-if)# vrrp 1 address-family ipv4`	Creates a VRRP group and enters VRRP configuration mode.
Step 7	**vrrs leader** *group name* **Example:** `Device(config-if-vrrp)# vrrs leader VRRP1`	Enables community and (or) extended community exchange with the specified neighbor.

	Command or Action	Purpose
Step 8	**vrrp** *group id* **ip** *ip address*{*primary* |*secondary*} **Example:** `Device(config-if-vrrp)# vrrp 1 ip 10.1.6.1`	Exits address family configuration mode and returns to router configuration mode.
Step 9	**exit** **Example:** `Device(config-if-vrrp)# exit`	Exits VRRP configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Specifies an interface to be configured and enters interface configuration mode.
Step 11	**ip pim redundancy** *group name* **vrrp dr-priority** *priority-value* **Example:** `Device(config-if)# ip pim redundancy VRRP1 vrrp dr-priority 90`	sets the priority for which a router is elected as the designated router (DR). • The redundancy dr-priority value should be same on all routers that are enabled with VRRP Aware PIM feature.
Step 12	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for VRRP Aware PIM

Example: VRRP Aware PIM

```
conf terminal
 fhrp version vrrp v3
interface Ethernet0/0
 ip  address 192.0.2.2
 vrrp 1 address-family ipv4

 vrrp 1 ip 10.1.6.1

 vrrs leader VRRP1
interface Ethernet0/0
 ip pim redundancy VRRP1 vrrp  dr-priority 90
 !
```

Additional References for VRRP Aware PIM

Related Documents

Related Topic	Document Title
IP multicast commands	Cisco IOS IP Multicast Command Reference
Configuring VRRP	First Hop Redundancy Protocols Configuration Guide
IP multicast PIM	IP Multicast: PIM Configuration Guide

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for VRRP Aware PIM

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 12

Verifying IP Multicast Operation

This module describes how to verify IP multicast operation in a network after Protocol Independent Multicast (PIM) sparse mode (PIM-SM) or Source Specific Multicast (PIM-SSM) has been implemented. The tasks in this module can be used to test IP multicast reachability and to confirm that receivers and sources are operating as expected in an IP multicast network.

Prerequisites for Verifying IP Multicast Operation

- Before performing the tasks in this module, you should be familiar with the concepts described in the " IP Multicast Technology Overview " module.
- The tasks in this module assume that IP multicast has been enabled and that PIM-SM or SSM has been configured using the relevant tasks described in the " Configuring Basic IP Multicast " module.

Restrictions for Verifying IP Multicast Operation

- For PIM-SM, this module assumes that the shortest path tree (SPT) threshold for PIM-enabled routers is set to the value of zero (the default) and not infinity. For more information about setting the SPT threshold, see the **ip pim spt-threshold** command page in the *Cisco IOS IP Multicast Command Reference*.
- Verifying IP multicast operation in a bidirectional PIM (bidir-PIM) network or a PIM-SM network with a finite or infinite SPT threshold is outside the scope of this module.

Information About Verifying IP Multicast Operation

Guidelines for Verifying IP Multicast Operation in a PIM-SM and PIM-SSM Network Environment

When you verify the operation of IP multicast in a PIM-SM network environment or in an PIM-SSM network environment, a useful approach is to begin the verification process on the last hop router, and then continue the verification process on the routers along the SPT until the first hop router has been reached. The goal of the verification is to ensure that IP multicast traffic is being routed properly through an IP multicast network.

Common Commands Used to Verify IP Multicast Operation on the Last Hop Router for PIM-SM and PIM-SSM

The table describes the common commands used to verify IP multicast operation on the last hop router in PIM-SM and PIM-SSM network environments.

Table 2: Common IP Multicast Verification Commands (Last Hop Router)

Command	Description and Purpose
show ip igmp groups	Displays the multicast groups with receivers that are directly connected to the router and that were learned through the Internet Group Management Protocol (IGMP). • Use this command to confirm that the IGMP cache is being properly populated on the last hop router for the groups that receivers on the LAN have joined.
show ip pim rp mapping	Displays all group-to-RP mappings of which the router is aware (either configured or learned from Auto-RP or BSR). • Use this command to confirm that the group-to-RP mappings are being populated correctly on the last hop router. **Note** The **show ip pim rp mapping**command does not work with routers in a PIM-SSM network because PIM-SSM does not use rendezvous points (RPs).
show ip mroute	Displays the contents of the multicast routing (mroute) table. • Use this command to verify that the mroute table is being populated properly on the last hop router.
show ip interface	Displays information and statistics about configured interfaces. • Use this command to verify that IP multicast fast switching is enabled on the outgoing interface on the last hop router.

Command	Description and Purpose
show ip mfib	Displays the forwarding entries and interfaces in the IP Multicast Forwarding Information Base (MFIB).
show ip pim interface count	Displays statistics related to the number of multicast packets received by and sent out a PIM-enabled interface. • Use this command on the last hop router to confirm that multicast traffic is being forwarded on the last hop router.
show ip mroute active	Displays the rate that active sources are sending to multicast groups, in kilobits per second (kb/s). • Use this command to display information about the multicast packet rate for active sources sending to groups on the last hop router.
show ip mroute count	Displays statistics related to mroutes in the mroute table. • Use this command on the last hop router to confirm that multicast traffic is flowing on the last hop router.

Common Commands Used to Verify IP Multicast Operation on Routers Along the SPT for PIM-SM and PIM-SSM

The table describes the common commands used to verify IP multicast operation on routers along the SPT in PIM-SM and PIM-SSM network environments.

Table 3: Common IP Multicast Verification Commands (Routers Along SPT)

Command	Description and Purpose
show ip mroute	Displays the contents of the mroute table. • Use this command to confirm that the Reverse Path Forwarding (RPF) neighbor toward the source is the expected RPF neighbor for each router along the SPT.
show ip mroute active	Displays the rate that active sources are sending to multicast groups, in kb/s. • Use this command to display information about the multicast packet rate for active sources sending to groups on routers along the SPT.

Common Commands Used to Verify IP Multicast Operation on the First Hop Router for PIM-SM and PIM-SSM

The table describes the common commands used to verify IP multicast operation on the first hop router in PIM-SM and PIM-SSM network environments.

Table 4: Common IP Multicast Verification Commands (First Hop Router)

Command	Description and Purpose
show ip mroute	Displays the contents of the mroute table. • Use this command to confirm that the F flag is set for the mroutes on the first hop router.
show ip mroute active	Displays the rate that active sources are sending to multicast groups, in kb/s. • Use this command to display information about the multicast packet rate for active sources sending to groups on the first hop router.

How to Verify IP Multicast Operation

Using PIM-Enabled Routers to Test IP Multicast Reachability

If all the PIM-enabled routers and access servers that you administer are members of a multicast group, pinging that group causes all routers to respond, which can be a useful administrative and debugging tool.

To use PIM-enabled routers to test IP multicast reachability, perform the following tasks:

Configuring Routers to Respond to Multicast Pings

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp join-group** *group-address*
5. Repeat Step 3 and Step 4 for each interface on the router participating in the multicast network.
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**interface** *type number*	Enters interface configuration mode. For the *type* and *number* arguments, specify an interface that is directly connected to hosts or is facing hosts.
Step 4	**ip igmp join-group** *group-address*	(Optional) Configures an interface on the router to join the specified group.

	Command or Action	Purpose
		For the purpose of this task, configure the same group address for the *group-address* argument on all interfaces on the router participating in the multicast network. **Note** With this method, the router accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the router from fast switching.
Step 5	Repeat Step 3 and Step 4 for each interface on the router participating in the multicast network.	--
Step 6	**end**	Ends the current configuration session and returns to privileged EXEC mode.

Pinging Routers Configured to Respond to Multicast Pings

on a router to initiate a ping test to the routers configured to respond to multicast pings. This task is used to test IP multicast reachability in a network.

SUMMARY STEPS

1. **enable**
2. **ping** *group-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**ping** *group-address*	Pings an IP multicast group address. A successful response indicates that the group address is functioning.

Verifying IP Multicast Operation in a PIM-SM or a PIM-SSM Network

Perform the following optional tasks to verify IP multicast operation in a PIM-SM or a PIM-SSM network. The steps in these tasks help to locate a faulty hop when sources and receivers are not operating as expected.

Note If packets are not reaching their expected destinations, you might want consider disabling IP multicast fast switching, which would place the router in process switching mode. If packets begin reaching their proper destinations after IP multicast fast switching has been disabled, then the issue most likely was related to IP multicast fast switching.

Verifying IP Multicast Operation on the Last Hop Router

Perform the following task to verify the operation of IP multicast on the last hop router.

Note If you are verifying a last hop router in a PIM-SSM network, ignore Step 3.

SUMMARY STEPS

1. **enable**
2. **show ip igmp groups**
3. **show ip pim rp mapping**
4. **show ip mroute**
5. **show ip interface** [*type number*]
6. **show ip mfib**
7. **show ip pim interface count**
8. **show ip mroute count**
9. **show ip mroute active** [*kb/s*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **show ip igmp groups**

Use this command to verify IGMP memberships on the last hop router. This information will confirm the multicast groups with receivers that are directly connected to the last hop router and that are learned through IGMP.

The following is sample output from the **show ip igmp groups** command:

Example:

```
Router# show ip igmp groups
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
239.1.2.3        GigabitEthernet1/0/0     00:05:14  00:02:14  10.1.0.6
224.0.1.39       GigabitEthernet0/0/0     00:09:11  00:02:08  172.31.100.1
```

Step 3 **show ip pim rp mapping**

Use this command to confirm that the group-to-RP mappings are being populated correctly on the last hop router.

Note Ignore this step if you are verifying a last hop router in a PIM-SSM network. The **show ip pim rp mapping**command does not work with routers in a PIM-SSM network because PIM-SSM does not use RPs. In addition, if configured correctly, PIM-SSM groups should not appear in the output of the **show ip pim rp mapping**command.

The following is sample output from the **show ip pim rp mapping**command:

Example:

```
Router# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 172.16.0.1 (?), v2v1
    Info source: 172.16.0.1 (?), elected via Auto-RP
         Uptime: 00:09:11, expires: 00:02:47
```

Step 4 **show ip mroute**

Use this command to verify that the mroute table is being populated properly on the last hop router.

The following is sample output from the **show ip mroute** command:

Example:

```
Router# show ip mroute
(*, 239.1.2.3), 00:05:14/00:03:04, RP 172.16.0.1, flags: SJC
  Incoming interface: GigabitEthernet0/0/0, RPF nbr 172.31.100.1
  Outgoing interface list:
  GigabitEthernet1/0, Forward/Sparse-Dense, 00:05:10/00:03:04

(10.0.0.1, 239.1.2.3), 00:02:49/00:03:29, flags: T
  Incoming interface: GigabitEthernet0/0/0, RPF nbr 172.31.100.1
  Outgoing interface list:
  GigabitEthernet1/0, Forward/Sparse-Dense, 00:02:49/00:03:04

(*, 224.0.1.39), 00:10:05/stopped, RP 0.0.0.0, flags: DC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse-Dense, 00:05:15/00:00:00
    GigabitEthernet0/0, Forward/Sparse-Dense, 00:10:05/00:00:00

(172.16.0.1, 224.0.1.39), 00:02:00/00:01:33, flags: PTX
  Incoming interface: GigabitEthernet0/0/0, RPF nbr 172.31.100.1
```

Step 5 **show ip interface** [*type number*]

Use this command to verify that multicast fast switching is enabled for optimal performance on the outgoing interface on the last hop router.

Note Using the **no ip mroute-cache** interface command disables IP multicast fast-switching. When IP multicast fast switching is disabled, packets are forwarded through the process-switched path.

The following is sample output from the **show ip interface**command for a particular interface:

Example:

```
Router# show ip interface GigabitEthernet 0/0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 172.31.100.2/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.1 224.0.0.22 224.0.0.13
      224.0.0.5 224.0.0.6
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
```

```
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is disabled
  IP Fast switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect inbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
```

Step 6 **show ip mfib**

Use this command to display the forwarding entries and interfaces in the IP Multicast Forwarding Information Base (MFIB).

Example:

Step 7 **show ip pim interface count**

Use this command to confirm that multicast traffic is being forwarded on the last hop router.

The following is sample output from the **show ip pim interface** command with the **count** keyword:

Example:

```
Router# show ip pim interface count

State: * - Fast Switched, D - Distributed Fast Switched
       H - Hardware Switching Enabled
Address          Interface             FS Mpackets In/Out
172.31.100.2     GigabitEthernet0/0/0  *   4122/0
10.1.0.1         GigabitEthernet1/0/0  *   0/3193
```

Step 8 **show ip mroute count**

Use this command to confirm that multicast traffic is being forwarded on the last hop router.

The following is sample output from the **show ip mroute**command with the **count** keyword:

Example:

```
Router# show ip mroute count
IP Multicast Statistics
6 routes using 4008 bytes of memory
3 groups, 1.00 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)
```

```
Group: 239.1.2.3, Source count: 1, Packets forwarded: 3165, Packets received: 3165
  RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0
  Source: 10.0.0.1/32, Forwarding: 3165/20/28/4, Other: 0/0/0

Group: 224.0.1.39, Source count: 1, Packets forwarded: 21, Packets received: 120
  Source: 172.16.0.1/32, Forwarding: 21/1/48/0, Other: 120/0/99

Group: 224.0.1.40, Source count: 1, Packets forwarded: 10, Packets received: 10
  Source: 172.16.0.1/32, Forwarding: 10/1/48/0, Other: 10/0/0
```

Step 9 **show ip mroute active** [*kb/s*]

Use this command on the last hop router to display information about active multicast sources sending traffic to groups on the last hop router. The output of this command provides information about the multicast packet rate for active sources.

Note By default, the output of the **show ip mroute** command with the **active** keyword displays information about active sources sending traffic to groups at a rate greater than or equal to 4 kb/s. To display information about active sources sending low-rate traffic to groups (that is, traffic less than 4 kb/s), specify a value of 1 for the *kb/s* argument. Specifying a value of 1 for this argument displays information about active sources sending traffic to groups at a rate equal to or greater than 1 kb/s, which effectively displays information about all possible active source traffic.

The following is sample output from the **show ip mroute**command with the **active** keyword:

Example:

```
Router# show ip mroute active
Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 50 secs), 4 kbps(life avg)
```

Verifying IP Multicast on Routers Along the SPT

Perform the following task to verify the operation of IP multicast on routers along the SPT in a PIM-SM or PIM-SSM network.

SUMMARY STEPS

1. **enable**
2. **show ip mroute** [*group-address*]
3. **show ip mroute active**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **show ip mroute** [*group-address*]

Use this command on routers along the SPT to confirm the RPF neighbor toward the source for a particular group or groups.

The following is sample output from the **show ip mroute**command for a particular group:

Example:

```
Router# show ip mroute 239.1.2.3
(*, 239.1.2.3), 00:17:56/00:03:02, RP 172.16.0.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet0/0/0, Forward/Sparse-Dense, 00:17:56/00:03:02

(10.0.0.1, 239.1.2.3), 00:15:34/00:03:28, flags: T
  Incoming interface: Serial1/0, RPF nbr 172.31.200.1
  Outgoing interface list:
    GigabitEthernet0/0/0, Forward/Sparse-Dense, 00:15:34/00:03:02
```

Step 3 **show ip mroute active**

Use this command on routers along the SPT to display information about active multicast sources sending to groups. The output of this command provides information about the multicast packet rate for active sources.

Note By default, the output of the **show ip mroute** command with the **active** keyword displays information about active sources sending traffic to groups at a rate greater than or equal to 4 kb/s. To display information about active sources sending low-rate traffic to groups (that is, traffic less than 4 kb/s), specify a value of 1 for the *kb/s* argument. Specifying a value of 1 for this argument displays information about active sources sending traffic to groups at a rate equal to or greater than 1 kb/s, which effectively displays information about all possible active source traffic.

The following is sample output from the **show ip mroute** command with the **active** keyword:

Example:

```
Router# show ip mroute active
Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)
```

Verifying IP Multicast on the First Hop Router

Perform the following task to verify the operation of IP multicast on the first hop router.

SUMMARY STEPS

1. **enable**
2. **show ip mroute** [*group-address*]
3. **show ip mroute active** [*kb/s*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

```
Router> enable
```

Step 2 **show ip mroute** [*group-address*]

Use this command on the first hop router to confirm the F flag has been set for mroutes on the first hop router.

The following is sample output from the **show ip mroute**for a particular group:

Example:

```
Router# show ip mroute 239.1.2.3
(*, 239.1.2.3), 00:18:10/stopped, RP 172.16.0.1, flags: SPF
  Incoming interface: Serial1/0, RPF nbr 172.31.200.2
  Outgoing interface list: Null

(10.0.0.1, 239.1.2.3), 00:18:10/00:03:22, flags: FT
  Incoming interface: GigabitEthernet0/0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/0, Forward/Sparse-Dense, 00:18:10/00:03:19
```

Step 3 **show ip mroute active** [*kb/s*]

Use this command on the first hop router to display information about active multicast sources sending to groups. The output of this command provides information about the multicast packet rate for active sources.

Note By default, the output of the **show ip mroute** command with the **active** keyword displays information about active sources sending traffic to groups at a rate greater than or equal to 4 kb/s. To display information about active sources sending low-rate traffic to groups (that is, traffic less than 4 kb/s), specify a value of 1 for the *kb/s* argument. Specifying a value of 1 for this argument displays information about active sources sending traffic to groups at a rate equal to or greater than 1 kb/s, which effectively displays information about all possible active source traffic.

The following is sample output from the **show ip mroute** command with the **active** keyword:

Example:

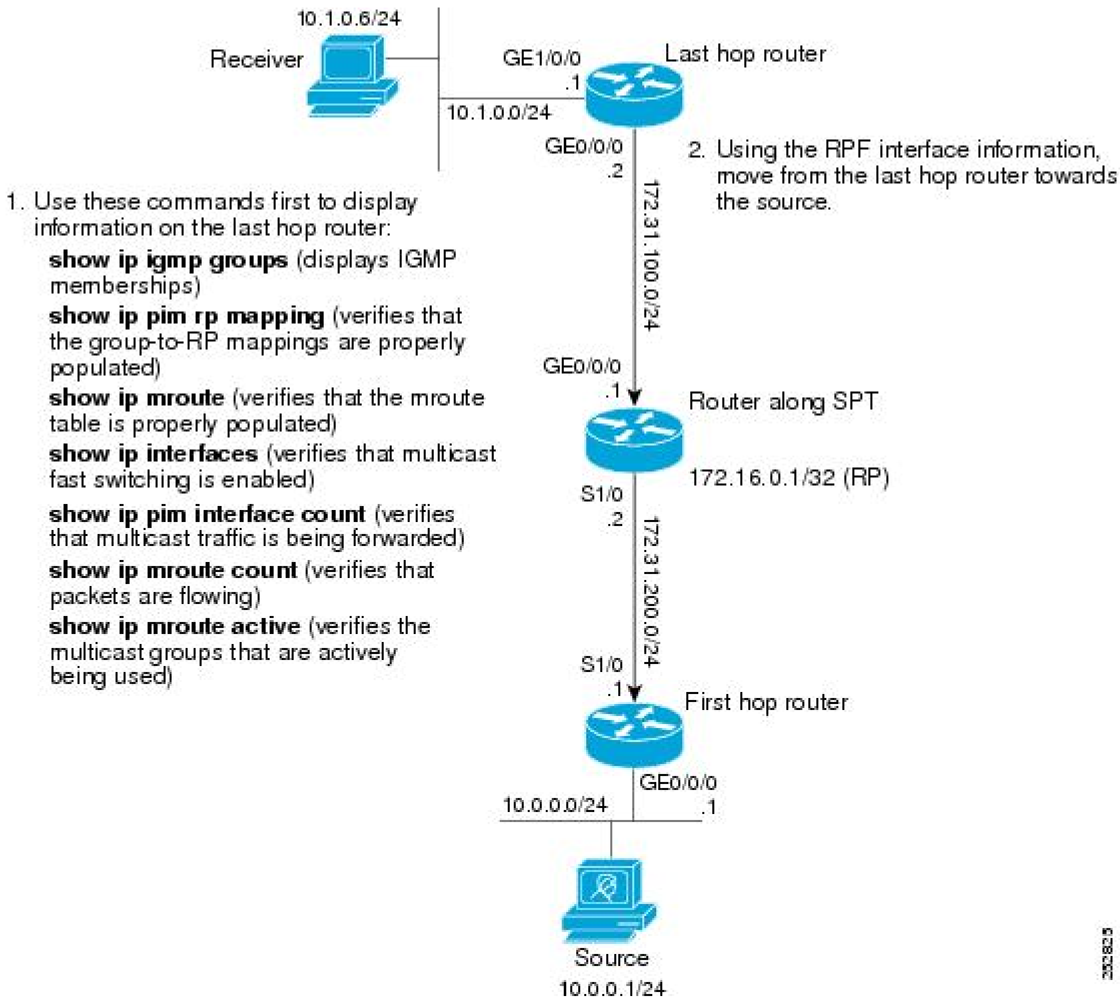
```
Router# show ip mroute active
Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)
```

Configuration Examples for Verifying IP Multicast Operation

Verifying IP Multicast Operation in a PIM-SM or PIM-SSM Network Example

The following example shows how to verify IP multicast operation after PIM-SM has been deployed in a network. The example is based on the PIM-SM topology illustrated in the figure.

From the last hop router to the first hop router shown in the figure, this example shows how to verify IP multicast operation for this particular PIM-SM network topology.



Verifying IP Multicast on the Last Hop Router Example

The following is sample output from the **show ip igmp groups** command. The sample output displays the IGMP memberships on the last hop router shown in the figure. This command is used in this example to confirm that the IGMP cache is being properly populated for the groups that receivers on the LAN have joined.

```
Router# show ip igmp groups
```

```
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
239.1.2.3        GigabitEthernet1/0/0     00:05:14  00:02:14  10.1.0.6
224.0.1.39       GigabitEthernet0/0/0     00:09:11  00:02:08  172.31.100.1
```

The following is sample output from the **show ip pim rp mapping**command. In the sample output, notice the RP address displayed for the RP field. Use the RP address and group information to verify that the group-to-RP mappings have been properly populated on the last hop router shown in the figure.

Note In the output, the "(?)" indicates that the router is unable to resolve an IP address to a hostname.

```
Router# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 172.16.0.1 (?), v2v1
    Info source: 172.16.0.1 (?), elected via Auto-RP
         Uptime: 00:09:11, expires: 00:02:47
```

The following is sample output from the **show ip mroute** command. This command is used to verify that the mroute table is being properly populated on the last hop router shown in the figure. In the sample output, notice the T flag for the (10.0.0.1, 239.1.2.3) mroute. The T flag indicates that the SPT-bit has been set, which means a multicast packet was received on the SPT tree for this particular mroute. In addition, the RPF nbr field should point toward the RPF neighbor with the highest IP address determined by unicast routing toward the multicast source.

```
Router# show ip mroute
(*, 239.1.2.3), 00:05:14/00:03:04, RP 172.16.0.1, flags: SJC
  Incoming interface: GigabitEthernet0/0/0, RPF nbr 172.31.100.1
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:05:10/00:03:04

(10.0.0.1, 239.1.2.3), 00:02:49/00:03:29, flags: T
  Incoming interface: GigabitEthernet0/0/0, RPF nbr 172.31.100.1
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:02:49/00:03:04

(*, 224.0.1.39), 00:10:05/stopped, RP 0.0.0.0, flags: DC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:05:15/00:00:00
    Ethernet0/0, Forward/Sparse-Dense, 00:10:05/00:00:00
```

The following is sample output from the **show ip interface** command for the incoming interface. This command is used in this example to confirm that IP multicast fast switching is enabled on the last hop router shown in the figure. When IP multicast fast switching is enabled, the line "IP multicast fast switching is enabled" displays in the output.

```
Router# show ip interface GigabitEthernet 0/0/0
GigabitEthernet0/0/0 is up, line protocol is up
  Internet address is 172.31.100.2/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.1 224.0.0.22 224.0.0.13
```

```
      224.0.0.5 224.0.0.6
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is disabled
  IP Fast switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect inbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
```

The following is sample output from the **show ip pim interface count** command. This command is used in this example to confirm that multicast traffic is being forwarded to the last hop router shown in the figure. In the sample output, notice the Mpackets In/Out field. This field displays the number of multicast packets received by and sent on each interface listed in the output.

```
Router# show ip pim interface count

State: * - Fast Switched, D - Distributed Fast Switched
       H - Hardware Switching Enabled
Address          Interface                FS Mpackets In/Out
172.31.100.2     GigabitEthernet0/0/0     *   4122/0
10.1.0.1         GigabitEthernet1/0/0     *   0/3193
```

The following is sample output from the **show ip mroute**command with the **count** keyword. This command is used on the last hop router shown in the figure to verify the packets being sent to groups from active sources. In the sample output, notice the packet count displayed for the Forwarding field. This field displays the packet forwarding count for sources sending to groups.

```
Router# show ip mroute count
IP Multicast Statistics
6 routes using 4008 bytes of memory
3 groups, 1.00 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 239.1.2.3, Source count: 1, Packets forwarded: 3165, Packets received: 3165
  RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0
  Source: 10.0.0.1/32, Forwarding: 3165/20/28/4, Other: 0/0/0

Group: 224.0.1.39, Source count: 1, Packets forwarded: 21, Packets received: 120
  Source: 172.16.0.1/32, Forwarding: 21/1/48/0, Other: 120/0/99
```

```
Group: 224.0.1.40, Source count: 1, Packets forwarded: 10, Packets received: 10
  Source: 172.16.0.1/32, Forwarding: 10/1/48/0, Other: 10/0/0
```

The following is sample output from the **show ip mroute**command with the **active** keyword. This command is used on the last hop router shown in the figure to confirm the multicast groups with active sources on the last hop router.

Note In the output, the "(?)" indicates that the router is unable to resolve an IP address to a hostname.

```
Router# show ip mroute active
Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 50 secs), 4 kbps(life avg)
```

Verifying IP Multicast on Routers Along the SPT Example

The following is sample output from the **show ip mroute**for a particular group. This command is used in this example to verify that the RPF neighbor toward the source is the expected RPF neighbor for the router along the SPT shown in the figure.

```
Router# show ip mroute 239.1.2.3
(*, 239.1.2.3), 00:17:56/00:03:02, RP 172.16.0.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse-Dense, 00:17:56/00:03:02

(10.0.0.1, 239.1.2.3), 00:15:34/00:03:28, flags: T
  Incoming interface: Serial1/0, RPF nbr 172.31.200.1
  Outgoing interface list:
```

Ethernet0/0, Forward/Sparse-Dense, 00:15:34/00:03:02

The following is sample output from the **show ip mroute** command with the **active** keyword from the router along the SPT shown in the figure. This command is used to confirm the multicast groups with active sources on this router.

Note In the output, the "(?)" indicates that the router is unable to resolve an IP address to a hostname.

```
Router# show ip mroute active
Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)
```

Verifying IP Multicast on the First Hop Router Example

The following is sample output from the **show ip mroute**for a particular group. This command is used in this example to verify the packets being sent to groups from active sources on the first hop router shown in the figure. In the sample output, notice the packet count displayed for the Forwarding field. This field displays the packet forwarding count for sources sending to groups on the first hop router.

Note The RPF nbr 0.0.0.0 field indicates that the source of an mroute has been reached.

```
Router# show ip mroute 239.1.2.3
(*, 239.1.2.3), 00:18:10/stopped, RP 172.16.0.1, flags: SPF
  Incoming interface: Serial1/0, RPF nbr 172.31.200.2
  Outgoing interface list: Null

(10.0.0.1, 239.1.2.3), 00:18:10/00:03:22, flags: FT
  Incoming interface: GigabitEthernet0/0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/0/0, Forward/Sparse-Dense, 00:18:10/00:03:19
```

The following is sample output from the **show ip mroute** command with the **active** keyword from the first hop router shown in the figure:

Note In the output, the “(?)” indicates that the router is unable to resolve an IP address to a host name.

```
Router# show ip mroute active
Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)
```

Additional References

Related Documents

Related Topic	Document Title
Overview of the IP multicast technology area	“ IP Multicast Technology Overview ” module
PIM-SM and SSM concepts and configuration examples	“ Configuring Basic IP Multicast ” module
IP multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Verifying IP Multicast Operation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 13

Monitoring and Maintaining IP Multicast

This module describes many ways to monitor and maintain an IP multicast network, such as

- displaying which neighboring multicast routers are peering with the local router
- displaying multicast packet rates and loss information
- tracing the path from a source to a destination branch for a multicast distribution tree
- displaying the contents of the IP multicast routing table, information about interfaces configured for PIM, the PIM neighbors discovered by the router, and contents of the IP fast-switching cache
- clearing caches, tables, and databases
- monitoring the delivery of IP multicast packets and being alerted if the delivery fails to meet certain parameters (IP multicast heartbeat)
- using session description and announcement protocols and applications to assist the advertisement of multicast multimedia conferences and other multicast sessions and communicating the relevant session setup information to prospective participants (SAP listener support)
- storing IP multicast packet headers in a cache and displaying them to find out information such as who is sending IP multicast packets to what groups and any multicast forwarding loops in your network
- using managed objects to remotely monitor and configure PIM using Simple Network Management Protocol (SNMP)
- disabling fast switching of IP multicast in order to log debug messages

Prerequisites for Monitoring and Maintaining IP Multicast

- Before performing the tasks in this module, you should be familiar with the concepts described in the "IP Multicast Technology Overview" module.

- You must also have enabled IP multicast and have Protocol Independent Multicast (PIM) configured and running on your network. Refer to the “Configuring Basic IP Multicast” module.

Information About Monitoring and Maintaining IP Multicast

IP Multicast Heartbeat

The IP Multicast Heartbeat feature enables you to monitor the delivery of IP multicast packets and to be alerted if the delivery fails to meet certain parameters.

Although you could alternatively use MRM to monitor IP multicast, you can perform the following tasks with IP multicast heartbeat that you cannot perform with MRM:

- Generate an SNMP trap
- Monitor a production multicast stream

When IP multicast heartbeat is enabled, the router monitors IP multicast packets destined for a particular multicast group at a particular interval. If the number of packets observed is less than a configured minimum amount, the router sends an SNMP trap to a specified network management station to indicate a loss of heartbeat exception.

The **ip multicast heartbeat** command does not create a heartbeat if there is no existing multicast forwarding state for *group* in the router. This command will not create a multicast forwarding state in the router. Use the **ip igmp static-group**commandon the router or on a downstream router to force forwarding of IP multicast traffic. Use the **snmp-server host ipmulticast** command to enable the sending of IP multicast traps to specific receiver hosts. Use the **debug ip mhbeat**command to debug the Multicast Heartbeat feature.

Session Announcement Protocol (SAP)

Session Announcement Protocol (SAP) listener support is needed to use session description and announcement protocols and applications to assist the advertisement of multicast multimedia conferences and other multicast sessions and to communicate the relevant session setup information to prospective participants.

Sessions are described by the Session Description Protocol (SDP), which is defined in RFC 2327. SDP provides a formatted, textual description of session properties (for example, contact information, session lifetime, and the media) being used in the session (for example, audio, video, and whiteboard) with their specific attributes such as time-to-live (TTL) scope, group address, and User Datagram Protocol (UDP) port number.

Many multimedia applications rely on SDP for session descriptions. However, they may use different methods to disseminate these session descriptions. For example, IP/TV relies on the web to disseminate session descriptions to participants. In this example, participants must know of a web server that provides the session information.

MBONE applications (for example, vic, vat, and wb) and other applications rely on multicast session information sent throughout the network. In these cases, SAP is used to transport the SDP session announcements. SAP Version 2 uses the well-known session directory multicast group 224.2.127.254 to disseminate SDP session descriptions for global scope sessions and group 239.255.255.255 for administrative scope sessions.

Note The Session Directory (SDR) application is commonly used to send and receive SDP/SAP session announcements.

PIM MIB Extensions for SNMP Traps for IP Multicast

Protocol Independent Multicast (PIM) is an IP multicast routing protocol used for routing multicast data packets to multicast groups. RFC 2934 defines the PIM MIB for IPv4, which describes managed objects that enable users to remotely monitor and configure PIM using Simple Network Management Protocol (SNMP).

PIM MIB extensions introduce the following new classes of PIM notifications:

- neighbor-change--This notification results from the following conditions:
  - A dDevice’s PIM interface is disabled or enabled (using the **ip pim** command in interface configuration mode)
  - A dDevice's PIM neighbor adjacency expires (defined in RFC 2934)
- rp-mapping-change--This notification results from a change in the rendezvous point (RP) mapping information due to either Auto-RP messages or bootstrap router (BSR) messages.
- invalid-pim-message--This notification results from the following conditions:
  - An invalid (*, G) Join or Prune message is received by the device (for example, when a dDevice receives a Join or Prune message for which the RP specified in the packet is not the RP for the multicast group)
  - An invalid PIM register message is received by the device (for example, when a dDevice receives a register message from a multicast group for which it is not the RP)

Benefits of PIM MIB Extensions

PIM MIB extensions:

- Allow users to identify changes in the multicast topology of their network by detecting changes in the RP mapping.
- Provide traps to monitor the PIM protocol on PIM-enabled interfaces.
- Help users identify routing issues when multicast neighbor adjacencies expire on a multicast interface.
- Enable users to monitor RP configuration errors (for example, errors due to flapping in dynamic RP allocation protocols like Auto-RP).

How to Monitor and Maintain IP Multicast

Displaying Multicast Peers Packet Rates and Loss Information and Tracing a Path

Monitor IP multicast routing when you want to know which neighboring multicast routers are peering with the local router, what the multicast packet rates and loss information are, or when you want to trace the path from a source to a destination branch for a multicast distribution tree.

SUMMARY STEPS

1. **enable**
2. **mrinfo** [*host-name* | *host-address*] [*source-address* | *interface*]
3. **mstat** {*source-name* | *source-address*} [*destination-name* | *destination-address*] [*group-name* | *group-address*]
4. **mtrace** {*source-name* | *source-address*} [*destination-name* | *destination-address*] [*group-name* | *group-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**mrinfo** [*host-name* | *host-address*] [*source-address* | *interface*] **Example:** `Router# mrinfo`	(Optional) Queries which neighboring multicast routers are "peering" with the local router.
Step 3	**mstat** {*source-name* | *source-address*} [*destination-name* | *destination-address*] [*group-name* | *group-address*] **Example:** `Router# mstat allsource`	(Optional) Displays IP multicast packet rate and loss information.
Step 4	**mtrace** {*source-name* | *source-address*} [*destination-name* | *destination-address*] [*group-name* | *group-address*] **Example:** `Router# mtrace allsource`	(Optional) Traces the path from a source to a destination branch for a multicast distribution tree.

Displaying IP Multicast System and Network Statistics

Display IP multicast system statistics to show the contents of the IP multicast routing table, information about interfaces configured for PIM, the PIM neighbors discovered by the router, contents of the IP fast-switching cache, and the contents of the circular cache header buffer.

SUMMARY STEPS

1. **enable**
2. **ping** [*group-name* | *group-address*]
3. **show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*type number*] [**summary**] [**count**] [**active** *kbps*]
4. **show ip pim interface** [*type number*] [**df** | **count**] [*rp-address*] [**detail**]
5. **show ip pim neighbor** [*type number*]
6. **show ip pim rp** [**mapping** | **metric**] [*rp-address*]
7. **show ip rpf** {*source-address* | *source-name*} [**metric**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** [*group-name* | *group-address*] **Example:** `Router# ping cbone-audio`	(Optional) Sends an ICMP echo request message to a multicast group address or group name.
Step 3	**show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*type number*] [**summary**] [**count**] [**active** *kbps*] **Example:** `Router# show ip mroute cbone-audio`	(Optional) Displays the contents of the IP multicast routing table.
Step 4	**show ip pim interface** [*type number*] [**df** | **count**] [*rp-address*] [**detail**] **Example:** `Router# show ip pim interface gigabitethernet1/0/0 detail`	(Optional) Displays information about interfaces configured for PIM.
Step 5	**show ip pim neighbor** [*type number*] **Example:** `Router# show ip pim neighbor`	(Optional) Lists the PIM neighbors discovered by the router.

	Command or Action	Purpose
Step 6	**show ip pim rp** [**mapping** | **metric**] [*rp-address*] **Example:** `Router# show ip pim rp metric`	(Optional) Displays the RP routers associated with a sparse mode multicast group.
Step 7	**show ip rpf** {*source-address* | *source-name*} [**metric**] **Example:** `Router# show ip rpf 172.16.10.13`	(Optional) Displays how the router is doing RPF (that is, from the unicast routing table, DVMRP routing table, or static mroutes). Also displays the unicast routing metric.

Clearing IP Multicast Routing Table or Caches

Clear IP multicast caches and tables to delete entries from the IP multicast routing table, the Auto-RP cache, the IGMP cache, and the caches of Catalyst switches. When these entries are cleared, the information is refreshed by being relearned, thus eliminating any incorrect entries.

SUMMARY STEPS

1. **enable**
2. **clear ip mroute** {***** | *group-name* [*source-name* | *source-address*] | *group-address* [*source-name* | *source-address*]}
3. **clear ip pim auto-rp** *rp-address*
4. **clear ip igmp group** [*group-name* | *group-address* | *interface-type interface-number*]
5. **clear ip cgmp** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip mroute** {***** | *group-name* [*source-name* | *source-address*] | *group-address* [*source-name* | *source-address*]} **Example:** `Router# clear ip mroute 224.2.205.42 228.3.0.0`	(Optional) Deletes entries from the IP multicast routing table.
Step 3	**clear ip pim auto-rp** *rp-address* **Example:** `Router# clear ip pim auto-rp 224.5.6.7`	(Optional) Clears the Auto-RP cache.

	Command or Action	Purpose
Step 4	**clear ip igmp group** [*group-name* | *group-address* | *interface-type interface-number*] **Example:** `Router# clear ip igmp group 224.0.255.1`	(Optional) Deletes entries from the IGMP cache.
Step 5	**clear ip cgmp** [*interface-type interface-number*] **Example:** `Router# clear ip cgmp`	(Optional) Clears all group entries from the caches of Catalyst switches.

Monitoring IP Multicast Delivery Using IP Multicast Heartbeat

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing distributed**
4. **snmp-server host** {*hostname* | *ip-address*} [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string*[**udp-port** *port*] [*notification-type*]
5. **snmp-server enable traps ipmulticast**
6. **ip multicast heartbeat** *group-address minimum-number window-size interval*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing distributed** **Example:** `Router(config)# ip multicast-routing distributed`	Enables IP multicast routing.
Step 4	**snmp-server host** {*hostname* | *ip-address*} [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string*[**udp-port** *port*] [*notification-type*] **Example:**	Specifies the recipient of an SNMP notification operation.

	Command or Action	Purpose
	`Router(config)# snmp-server host 224.1.0.1 traps public`	
Step 5	**snmp-server enable traps ipmulticast** **Example:** `Router(config)# snmp-server enable traps ipmulticast`	Enables the router to send IP multicast traps.
Step 6	**ip multicast heartbeat** *group-address minimum-number window-size interval* **Example:** `Router(config)# ip multicast heartbeat 224.1.1.1 1 1 10`	Enables the monitoring of the IP multicast packet delivery. • The *interval* should be set to a multiple of 10 seconds on platforms that use Multicast Distributed Fast Switching (MDFS) because on those platforms, the packet counters are only updated once every 10 seconds. Other platforms may have other increments.

Advertising Multicast Multimedia Sessions Using SAP Listener

Enable SAP listener support when you want to use session description and announcement protocols and applications to assist the advertisement of multicast multimedia conferences and other multicast sessions and to communicate the relevant session setup information to prospective participants.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sap cache-timeout** *minutes*
4. **interface** *type number*
5. **ip sap listen**
6. **end**
7. **clear ip sap** [*group-address* | **"** *session-name* **"**]
8. **show ip sap** [*group-address* | **"** *session-name* **"**| **detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip sap cache-timeout** *minutes* **Example:** `Router(config)# ip sap cache-timeout 600`	(Optional) Limits how long a SAP cache entry stays active in the cache. • By default, SAP cache entries are deleted 24 hours after they are received from the network.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.
Step 5	**ip sap listen** **Example:** `Router(config-if)# ip sap listen`	Enables the Cisco IOS XE software to listen to session directory announcements.
Step 6	**end** **Example:** `Router(config-if)# end`	Ends the session and returns to EXEC mode.
Step 7	**clear ip sap** [*group-address* | **"** *session-name* **"**] **Example:** `Router# clear ip sap "Sample Session"`	Deletes a SAP cache entry or the entire SAP cache.
Step 8	**show ip sap** [*group-address* | **"** *session-name* **"**| **detail**] **Example:** `Router# show ip sap 224.2.197.250 detail`	(Optional) Displays the SAP cache.

Disabling Fast Switching of IP Multicast

Disable fast switching if you want to log debug messages, because when fast switching is enabled, debug messages are not logged.

You might also want to disable fast switching, which places the router in process switching, if packets are not reaching their destinations. If fast switching is disabled and packets are reaching their destinations, then switching may be the cause.

Fast switching of IP multicast packets is enabled by default on all interfaces (including generic routing encapsulation [GRE] and DVMRP tunnels), with one exception: It is disabled and not supported over X.25 encapsulated interfaces. The following are properties of fast switching:

- If fast switching is disabled on an *incoming* interface for a multicast routing table entry, the packet is sent at process level for all interfaces in the outgoing interface list.

- If fast switching is disabled on an *outgoing* interface for a multicast routing table entry, the packet is process-level switched for that interface, but may be fast switched for other interfaces in the outgoing interface list.
- When fast switching is enabled, debug messages are not logged.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip mroute-cache**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Specifies an interface.
Step 4	**no ip mroute-cache** **Example:** `Router(config-if)# no ip mroute-cache`	Disables fast switching of IP multicast.

Enabling PIM MIB Extensions for IP Multicast

Perform this task to enable PIM MIB extensions for IP multicast.

Note

- The pimInterfaceVersion object was removed from RFC 2934 and, therefore, is no longer supported in software.
- The following MIB tables are not supported in Cisco software:
  - pimIpMRouteTable
  - pimIpMRouteNextHopTable

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps pim** [**neighbor-change** | **rp-mapping-change** | **invalid-pim-message**]
4. **snmp-server host** *host-address* [**traps** | **informs**] *community-string* **pim**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps pim** [**neighbor-change** | **rp-mapping-change** | **invalid-pim-message**] **Example:** `Device(config)# snmp-server enable traps pim neighbor-change`	Enables a device to send PIM notifications. • **neighbor-change** --This keyword enables notifications indicating when a device's PIM interface is disabled or enabled, or when a device's PIM neighbor adjacency expires. • **rp-mapping-change** --This keyword enables notifications indicating a change in RP mapping information due to either Auto-RP messages or BSR messages. • **invalid-pim-message** --This keyword enables notifications for monitoring invalid PIM protocol operations (for example, when a device receives a join or prune message for which the RP specified in the packet is not the RP for the multicast group or when a device receives a register message from a multicast group for which it is not the RP).
Step 4	**snmp-server host** *host-address* [**traps** | **informs**] *community-string* **pim** **Example:** `Device(config)# snmp-server host 10.10.10.10 traps public pim`	Specifies the recipient of a PIM SNMP notification operation.

Configuration Examples for Monitoring and Maintaining IP Multicast

Displaying IP Multicast System and Network Statistics Example

The following is sample output from the **mrinfo** command:

```
Router# mrinfo

192.31.7.37 (labs-allcompany) [version cisco 12.3] [flags: PMSA]:
192.31.7.37 -> 192.31.7.34 (lab-southwest) [1/0/pim]
192.31.7.37 -> 192.31.7.47 (lab-northwest) [1/0/pim]
192.31.7.37 -> 192.31.7.44 (lab-southeast) [1/0/pim]
131.119.26.10 -> 131.119.26.9 (lab-northeast) [1/32/pim]
```

The following is sample output from the **mstat** command in user EXEC mode:

```
Router> mstat labs-in-china 172.16.0.1 224.0.255.255

Type escape sequence to abort.
Mtrace from 172.16.0.0 to 172.16.0.10 via group 224.0.255.255
>From source (labs-in-china) to destination (labs-in-africa)
Waiting to accumulate statistics......
Results after 10 seconds:
Source Response Dest Packet Statistics For Only For Traffic
172.16.0.0      172.16.0.10 All Multicast Traffic From 172.16.0.0
| __/ rtt 48 ms Lost/Sent = Pct Rate To 224.0.255.255
v / hop 48 ms --------------------- --------------------
172.16.0.1      labs-in-england
| ^ ttl 1
v | hop 31 ms 0/12 = 0% 1 pps 0/1 = --% 0 pps
172.16.0.2
172.16.0.3   infolabs.com
| ^ ttl 2
v | hop -17 ms -735/12 = --% 1 pps 0/1 = --% 0 pps
172.16.0.4
172.16.0.5      infolabs2.com
| ^ ttl 3
v | hop -21 ms -678/23 = --% 2 pps 0/1 = --% 0 pps
172.16.0.6
172.16.0.7      infolabs3.com
| ^ ttl 4
v | hop 5 ms 605/639 = 95% 63 pps 1/1 = --% 0 pps
172.16.0.8
172.16.0.9      infolabs.cisco.com
| \__ ttl 5
v \ hop 0 ms 4 0 pps 0 0 pps
172.16.0.0      172.16.0.10
Receiver Query Source
```

The following is sample output from the **mtrace** command in user EXEC mode:

```
Router> mtrace 172.16.0.0 172.16.0.10 239.254.254.254
Type escape sequence to abort.
Mtrace from 172.16.0.0 to 172.16.0.10 via group 239.254.254.254
From source (?) to destination (?)
```

```
Querying full reverse path...
0 172.16.0.10
-1 172.16.0.8 PIM thresh^ 0 0 ms
-2 172.16.0.6 PIM thresh^ 0 2 ms
-3 172.16.0.5 PIM thresh^ 0 894 ms
-4 172.16.0.3 PIM thresh^ 0 893 ms
-5 172.16.0.2 PIM thresh^ 0 894 ms
-6 172.16.0.1 PIM thresh^ 0 893 ms
```

Monitoring IP Multicast Delivery Using IP Multicast Heartbeat Example

The following example shows how to monitor IP multicast packets forwarded through this router to group address 244.1.1.1. If no packet for this group is received in a 10-second interval, an SNMP trap will be sent to the SNMP management station with the IP address of 224.1.0.1.

```
!
ip multicast-routing
!
snmp-server host 224.1.0.1 traps public
snmp-server enable traps ipmulticast
ip multicast heartbeat 224.1.1.1 1 1 10
```

Advertising Multicast Multimedia Sessions Using SAP Listener Example

The following example enables a router to listen to session directory announcements and changes the SAP cache timeout to 30 minutes.

```
ip multicast routing
ip sap cache-timeout 30
interface loopback 0
 ip address 10.0.0.51 255.255.255.0
 ip pim sparse-dense mode
 ip sap listen
```

The following is sample output from the **show ip sap** command for a session using multicast group 224.2.197.250:

```
Router# show ip sap 224.2.197.250
SAP Cache - 198 entries
Session Name: Session1
   Description: This broadcast is brought to you courtesy of Name1.
   Group: 0.0.0.0, ttl: 0, Contiguous allocation: 1
   Lifetime: from 10:00:00 PDT Jul 4 1999 until 10:00:00 PDT Aug 1 1999
   Uptime: 4d05h, Last Heard: 00:01:40
   Announcement source: 128.102.84.134
   Created by: sample 3136541828 3139561476 IN IP4 128.102.84.134
   Phone number: Sample Digital Video Lab (555) 555-5555
   Email: email1 <name@email.com>
   URL: http://url.com/
   Media: audio 20890 RTP/AVP 0
     Media group: 224.2.197.250, ttl: 127
     Attribute: ptime:40
   Media: video 62806 RTP/AVP 31
     Media group: 224.2.190.243, ttl: 127
```

Displaying IP Multicast System and Network Statistics Example

show ip mroute

The following is sample output from the **show ip mroute** command for a router operating in sparse mode:

```
Router# show ip mroute

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel,
       Y - Joined MDT-data group, y - Sending to MDT-data group
Timers: Uptime/Expires
Interface state: Interface, Next-Hop, State/Mode
(*, 224.0.255.3), uptime 5:29:15, RP is 192.168.37.2, flags: SC
  Incoming interface: Tunnel0, RPF neighbor 10.3.35.1, Dvmrp
  Outgoing interface list:
    GigabitEthernet0, Forward/Sparse, 5:29:15/0:02:57
(192.168.46.0/24, 224.0.255.3), uptime 5:29:15, expires 0:02:59, flags: C
  Incoming interface: Tunnel0, RPF neighbor 10.3.35.1
  Outgoing interface list:
 GigabitEthernet0, Forward/Sparse, 5:29:15/0:02:57
```

show ip pim interface

The following is sample output from the **show ip pim interface** command when an interface is specified:

```
Router# show ip pim interface GigabitEthernet1/0/0

Address          Interface                Ver/   Nbr    Query  DR     DR
                                          Mode   Count  Intvl  Prior
172.16.1.4       GigabitEthernet1/0/0     v2/S   1      100 ms 1      172.16.1.4
```

The following is sample output from the **show ip pim rp** command:

```
Router# show ip pim rp

Group:227.7.7.7, RP:10.10.0.2, v2, v1, next RP-reachable in 00:00:48
```

show ip pim rp

The following is sample output from the **show ip pim rp** command when the **mapping** keyword is specified:

```
Router# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent
Group(s) 227.0.0.0/8
  RP 10.10.0.2 (?), v2v1, bidir
    Info source:10.10.0.2 (?), via Auto-RP
         Uptime:00:01:42, expires:00:00:32
Group(s) 228.0.0.0/8
  RP 10.10.0.3 (?), v2v1, bidir
    Info source:10.10.0.3 (?), via Auto-RP
         Uptime:00:01:26, expires:00:00:34
Group(s) 229.0.0.0/8
```

```
  RP 10.10.0.5 (mcast1.cisco.com), v2v1, bidir
    Info source:10.10.0.5 (mcast1.cisco.com), via Auto-RP
         Uptime:00:00:52, expires:00:00:37
Group(s) (-)230.0.0.0/8
  RP 10.10.0.5 (mcast1.cisco.com), v2v1, bidir
    Info source:10.10.0.5 (mcast1.cisco.com), via Auto-RP
         Uptime:00:00:52, expires:00:00:37
```

The following is sample output from the **show ip pim rp** command when the **metric** keyword is specified:

```
Router# show ip pim rp metric

RP Address       Metric Pref    Metric        Flags   RPF Type   Interface
10.10.0.2        0              0               L     unicast    Loopback0
10.10.0.3        90             409600          L     unicast    GigabitEthernet3/3/0
10.10.0.5        90             435200          L     unicast    GigabitEthernet3/3/0
```

show ip rpf

The following is sample output from the **show ip rpf** command:

```
Router# show ip rpf 172.16.10.13

RPF information for host1 (172.16.10.13)
  RPF interface: BRI0
  RPF neighbor: sj1.cisco.com (172.16.121.10)
  RPF route/mask: 172.16.0.0/255.255.0.0
  RPF type: unicast
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
```

The following is sample output from the **show ip rpf**command when the **metric**keyword is specified:

```
Router# show ip rpf 172.16.10.13 metric
RPF information for host1.cisco.com (172.16.10.13)
  RPF interface: BRI0
  RPF neighbor: neighbor.cisco.com (172.16.121.10)
  RPF route/mask: 172.16.0.0/255.255.0.0
  RPF type: unicast
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
  Metric preference: 110
```

Enabling PIM MIB Extensions for IP Multicast Example

The following example shows how to configure a router to generate notifications indicating that a PIM interface of the router has been enabled. The first line configures PIM traps to be sent as SNMP v2c traps to the host with IP address 10.0.0.1. The second line configures the router to send the neighbor-change class of trap notification to the host.

```
snmp-server host 10.0.0.1 traps version 2c public pim
snmp-server enable traps pim neighbor-change
interface ethernet0/0
 ip pim sparse-dense-mode
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP Multicast Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2934	*Protocol Independent Multicast for IPv4 MIB*

MIBs

MIB	MIBs Link
• CISCO-IPMROUTE-MIB • MSDP-MIB • IGMP-STD-MIB	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Monitoring and Maintaining IP Multicast

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 14

Multicast User Authentication and Profile Support

Restrictions for Multicast User Authentication and Profile Support

The port, interface, VC, or VLAN ID is the user or subscriber identity. User identity by hostname, user ID, or password is not supported.

Information About Multicast User Authentication and Profile Support

IPv6 Multicast User Authentication and Profile Support

IPv6 multicast by design allows any host in the network to become a receiver or a source for a multicast group. Therefore, multicast access control is needed to control multicast traffic in the network. Access control functionality consists mainly of source access control and accounting, receiver access control and accounting, and provisioning of this access control mechanism.

Multicast access control provides an interface between multicast and authentication, authorization, and accounting (AAA) for provisioning, authorizing, and accounting at the last-hop device, receiver access control functions in multicast, and group or channel disabling capability in multicast.

When you deploy a new multicast service environment, it is necessary to add user authentication and provide a user profile download on a per-interface basis. The use of AAA and IPv6 multicast supports user authentication and downloading of the user profile in a multicast environment.

The event that triggers the download of a multicast access-control profile from the RADIUS server to the access device is arrival of an MLD join on the access device. When this event occurs, a user can cause the authorization cache to time out and request download periodically or use an appropriate multicast clear command to trigger a new download in case of profile changes.

Accounting occurs via RADIUS accounting. Start and stop accounting records are sent to the RADIUS server from the access device. In order for you to track resource consumption on a per-stream basis, these accounting records provide information about the multicast source and group. The start record is sent when the last-hop device receives a new MLD report, and the stop record is sent upon MLD leave or if the group or channel is deleted for any reason.

How to Configure Multicast User Authentication and Profile Support

Enabling AAA Access Control for IPv6 Multicast

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables the AAA access control system.

Specifying Method Lists and Enabling Multicast Accounting

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **aaa authorization multicast default** [*method3* | *method4*
4. **aaa accounting multicast default** [**start-stop** | **stop-only**] [**broadcast**] [*method1*] [*method2*] [*method3*] [*method4*
5. **interface** *type number*
6. **ipv6 multicast aaa account receive** *access-list-name* [**throttle** *throttle-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa authorization multicast default** [*method3* | *method4* **Example:** `Device(config)# aaa authorization multicast default`	Enables AAA authorization and sets parameters that restrict user access to an IPv6 multicast network.
Step 4	**aaa accounting multicast default** [**start-stop** | **stop-only**] [**broadcast**] [*method1*] [*method2*] [*method3*] [*method4* **Example:** `Device(config)# aaa accounting multicast default`	Enables AAA accounting of IPv6 multicast services for billing or security purposes when you use RADIUS.
Step 5	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 6	**ipv6 multicast aaa account receive** *access-list-name* [**throttle** *throttle-number* **Example:** `Device(config-if)# ipv6 multicast aaa account receive list1`	Enables AAA accounting on specified groups or channels.

Disabling the Device from Receiving Unauthenticated Multicast Traffic

In some situations, access control may be needed to prevent multicast traffic from being received unless the subscriber is authenticated and the channels are authorized as per access control profiles. That is, there should be no traffic at all unless specified otherwise by access control profiles.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast group-range** [*access-list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 multicast group-range** [*access-list-name*] **Example:** `Device(config)# ipv6 multicast group-range`	Disables multicast protocol actions and traffic forwarding for unauthorized groups or channels on all the interfaces in a device.

Configuration Examples for Multicast User Authentication and Profile Support

Example: Enabling AAA Access Control, Specifying Method Lists, and Enabling Multicast Accounting for IPv6

```
Device(config)# aaa new-model
Device(config)# aaa authorization multicast default
Device(config)# aaa accounting multicast default
Device(config)# interface FastEthernet 1/0
Device(config-if)# ipv6 multicast aaa account receive list1
```

Additional References for IPv6 Services: AAAA DNS Lookups

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*

Related Topic	Document Title
IPv4 services configuration	*IP Application Services Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
None.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multicast User Authentication and Profile Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 15

IPv6 Multicast: Bootstrap Router

Information About IPv6 Multicast: Bootstrap Router

IPv6 BSR

PIM devices in a domain must be able to map each multicast group to the correct RP address. The BSR protocol for PIM-SM provides a dynamic, adaptive mechanism to distribute group-to-RP mapping information rapidly throughout a domain. With the IPv6 BSR feature, if an RP becomes unreachable, it will be detected and the mapping tables will be modified so that the unreachable RP is no longer used, and the new tables will be rapidly distributed throughout the domain.

Every PIM-SM multicast group needs to be associated with the IP or IPv6 address of an RP. When a new multicast sender starts sending, its local DR will encapsulate these data packets in a PIM register message and send them to the RP for that multicast group. When a new multicast receiver joins, its local DR will send a PIM join message to the RP for that multicast group. When any PIM device sends a (*, G) join message, the PIM device needs to know which is the next device toward the RP so that G (Group) can send a message to that device. Also, when a PIM device is forwarding data packets using (*, G) state, the PIM device needs to know which is the correct incoming interface for packets destined for G, because it needs to reject any packets that arrive on other interfaces.

A small set of devices from a domain are configured as candidate bootstrap routers (C-BSRs) and a single BSR is selected for that domain. A set of devices within a domain are also configured as candidate RPs (C-RPs); typically, these devices are the same devices that are configured as C-BSRs. Candidate RPs periodically unicast candidate-RP-advertisement (C-RP-Adv) messages to the BSR of that domain, advertising their willingness to be an RP. A C-RP-Adv message includes the address of the advertising C-RP, and an optional list of group addresses and mask length fields, indicating the group prefixes for which the candidacy is advertised. The BSR then includes a set of these C-RPs, along with their corresponding group prefixes, in bootstrap messages (BSMs) it periodically originates. BSMs are distributed hop-by-hop throughout the domain.

Bidirectional BSR support allows bidirectional RPs to be advertised in C-RP messages and bidirectional ranges in the BSM. All devices in a system must be able to use the bidirectional range in the BSM; otherwise, the bidirectional RP feature will not function.

IPv6 BSR: Configure RP Mapping

The IPv6 BSR ability to configure RP mapping allows IPv6 multicast devices to be statically configured to announce scope-to-RP mappings directly from the BSR instead of learning them from candidate-RP messages. Announcing RP mappings from the BSR is useful in several situations:

- When an RP address never changes because there is only a single RP or the group range uses an anycast RP, it may be less complex to configure the RP address announcement statically on the candidate BSRs.
- When an RP address is a virtual RP address (such as when using bidirectional PIM), it cannot be learned by the BSR from a candidate-RP. Instead, the virtual RP address must be configured as an announced RP on the candidate BSRs.

IPv6 BSR: Scoped Zone Support

BSR provides scoped zone support by distributing group-to-RP mappings in networks using administratively scoped multicast. The user can configure candidate BSRs and a set of candidate RPs for each administratively scoped region in the user's domain.

For BSR to function correctly with administrative scoping, a BSR and at least one C-RP must be within every administratively scoped region. Administratively scoped zone boundaries must be configured at the zone border devices, because they need to filter PIM join messages that might inadvertently cross the border due to error conditions. In addition, at least one C-BSR within the administratively scoped zone must be configured to be a C-BSR for the administratively scoped zone's address range.

A separate BSR election will then take place (using BSMs) for every administratively scoped range, plus one for the global range. Administratively scoped ranges are identified in the BSM because the group range is marked to indicate that this is an administrative scope range, not just a range that a particular set of RPs is configured to handle.

Unless the C-RP is configured with a scope, it discovers the existence of the administratively scoped zone and its group range through reception of a BSM from the scope zone's elected BSR containing the scope zone's group range. A C-RP stores each elected BSR's address and the administratively scoped range contained in its BSM. It separately unicasts C-RP-Adv messages to the appropriate BSR for every administratively scoped range within which it is willing to serve as an RP.

All PIM devices within a PIM bootstrap domain where administratively scoped ranges are in use must be able to receive BSMs and store the winning BSR and RP set for all administratively scoped zones that apply.

IPv6 Multicast: RPF Flooding of BSR Packets

Cisco IPv6 devices provide support for the RPF flooding of BSR packets so that the device will not disrupt the flow of BSMs. The device will recognize and parse enough of the BSM to identify the BSR address. The device performs an RPF check for this BSR address and forwards the packet only if it is received on the RPF interface. The device also creates a BSR entry containing RPF information to use for future BSMs from the same BSR. When BSMs from a given BSR are no longer received, the BSR entry is timed out.

How to Configure IPv6 Multicast: Bootstrap Router

Configuring a BSR and Verifying BSR Information

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim** [**vrf** *vrf-name*] **bsr candidate bsr** *ipv6-address*[*hash-mask-length*] [**priority** *priority-value*]
4. **interface** *type number*
5. **ipv6 pim bsr border**
6. **end**
7. **show ipv6 pim** [**vrf** *vrf-name*] **bsr** {**election** | **rp-cache** | **candidate-rp**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim** [**vrf** *vrf-name*] **bsr candidate bsr** *ipv6-address*[*hash-mask-length*] [**priority** *priority-value*] **Example:** `Device(config)# ipv6 pim bsr candidate bsr 2001:DB8:3000:3000::42 124 priority 10`	Configures a device to be a candidate BSR.
Step 4	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 5	**ipv6 pim bsr border** **Example:** `Device(config-if)# ipv6 pim bsr border`	Configures a border for all BSMs of any scope on a specified interface.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.
Step 7	**show ipv6 pim** [**vrf** *vrf-name*] **bsr** {**election** | **rp-cache** | **candidate-rp**} **Example:** `Device# show ipv6 pim bsr election`	Displays information related to PIM BSR protocol processing.

Sending PIM RP Advertisements to the BSR

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim** [**vrf** *vrf-name*] **bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** *seconds*] [**scope** *scope-value*] [**bidir**]
4. **interface** *type number*
5. **ipv6 pim bsr border**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim** [**vrf** *vrf-name*] **bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** *seconds*] [**scope** *scope-value*] [**bidir**] **Example:** `Device(config)# ipv6 pim bsr candidate rp 2001:DB8:3000:3000::42 priority 0`	Sends PIM RP advertisements to the BSR.
Step 4	**interface** *type number* **Example:**	Specifies an interface type and number, and places the device in interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface FastEthernet 1/0`	
Step 5	**ipv6 pim bsr border** **Example:** `Device(config-if)# ipv6 pim bsr border`	Configures a border for all BSMs of any scope on a specified interface.

Configuring BSR for Use Within Scoped Zones

A user can configure candidate BSRs and a set of candidate RPs for each administratively scoped region in the domain.

If scope is specified on the candidate RP, then this device will advertise itself as C-RP only to the BSR for the specified scope. If the group list is specified along with the scope, then only prefixes in the access list with the same scope as that configured will be advertised.

If a scope is specified on the bootstrap device, the BSR will originate BSMs including the group range associated with the scope and accept C-RP announcements for groups that belong to the given scope.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim** [**vrf** *vrf-name*] **bsr candidate bsr** *ipv6-address* [*hash-mask-length*] [**priority** *priority-value*]
4. **ipv6 pim** [**vrf** *vrf-name*] **bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** *seconds*] [**scope** *scope-value*] [**bidir**]
5. **interface** *type number*
6. **ipv6 multicast boundary scope** *scope-value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim** [**vrf** *vrf-name*] **bsr candidate bsr** *ipv6-address* [*hash-mask-length*] [**priority** *priority-value*] **Example:**	Configures a device to be a candidate BSR.

	Command or Action	Purpose
	`Device(config)# ipv6 pim bsr candidate bsr 2001:DB8:1:1:4`	
Step 4	**ipv6 pim** [**vrf** *vrf-name*] **bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** *seconds*] [**scope** *scope-value*] [**bidir**] **Example:** `Device(config)# ipv6 pim bsr candidate rp 2001:DB8:1:1:1 group-list list scope 6`	Configures the candidate RP to send PIM RP advertisements to the BSR.
Step 5	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 6	**ipv6 multicast boundary scope** *scope-value* **Example:** `Device(config-if)# ipv6 multicast boundary scope 6`	Configures a multicast boundary on the interface for a specified scope.

Configuring BSR Devices to Announce Scope-to-RP Mappings

IPv6 BSR devices can be statically configured to announce scope-to-RP mappings directly instead of learning them from candidate-RP messages. A user might want to configure a BSR device to announce scope-to-RP mappings so that an RP that does not support BSR is imported into the BSR. Enabling this feature also allows an RP positioned outside the enterprise's BSR domain to be learned by the known remote RP on the local candidate BSR devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim** [**vrf** *vrf-name*] **bsr announced rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**bidir**] [**scope** *scope-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim** [**vrf** *vrf-name*] **bsr announced rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**bidir**] [**scope** *scope-value*] **Example:** `Device(config)# ipv6 pim bsr announced rp 2001:DB8:3000:3000::42 priority 0`	Announces scope-to-RP mappings directly from the BSR for the specified candidate RP.

Configuration Examples for IPv6 Multicast: Bootstrap Router

Example: Configuring a BSR

```
Device# show ipv6 pim bsr election

PIMv2 BSR information
BSR Election Information
Scope Range List: ff00::/8
This system is the Bootstrap Router (BSR)
BSR Address: 60::1:1:4
Uptime: 00:11:55, BSR Priority: 0, Hash mask length: 126
RPF: FE80::A8BB:CCFF:FE03:C400,Ethernet0/0
BS Timer: 00:00:07
This system is candidate BSR
Candidate BSR address: 60::1:1:4, priority: 0, hash mask length: 126
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Multicast: Bootstrap Router

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 16

IPv6 Multicast: PIM Sparse Mode

IPv6 multicast provides support for intradomain multicast routing using PIM sparse mode (PIM-SM). PIM-SM uses unicast routing to provide reverse-path information for multicast tree building, but it is not dependent on any particular unicast routing protocol.

Information About IPv6 Multicast PIM Sparse Mode

Protocol Independent Multicast

Protocol Independent Multicast (PIM) is used between devices so that they can track which multicast packets to forward to each other and to their directly connected LANs. PIM works independently of the unicast routing protocol to perform send or receive multicast route updates like other protocols. Regardless of which unicast routing protocols are being used in the LAN to populate the unicast routing table, Cisco IOS PIM uses the existing unicast table content to perform the Reverse Path Forwarding (RPF) check instead of building and maintaining its own separate routing table.

You can configure IPv6 multicast to use either a PIM- Sparse Mode (SM) or PIM-Source Specific Multicast (SSM) operation, or you can use both PIM-SM and PIM-SSM together in your network.

PIM-Sparse Mode

IPv6 multicast provides support for intradomain multicast routing using PIM-SM. PIM-SM uses unicast routing to provide reverse-path information for multicast tree building, but it is not dependent on any particular unicast routing protocol.

PIM-SM is used in a multicast network when relatively few devices are involved in each multicast and these devices do not forward multicast packets for a group, unless there is an explicit request for the traffic. PIM-SM distributes information about active sources by forwarding data packets on the shared tree. PIM-SM initially uses shared trees, which requires the use of an RP.

Requests are accomplished via PIM joins, which are sent hop by hop toward the root node of the tree. The root node of a tree in PIM-SM is the RP in the case of a shared tree or the first-hop device that is directly connected to the multicast source in the case of a shortest path tree (SPT). The RP keeps track of multicast groups and the hosts that send multicast packets are registered with the RP by that host's first-hop device.

As a PIM join travels up the tree, devices along the path set up multicast forwarding state so that the requested multicast traffic will be forwarded back down the tree. When multicast traffic is no longer needed, a device sends a PIM prune up the tree toward the root node to prune (or remove) the unnecessary traffic. As this PIM prune travels hop by hop up the tree, each device updates its forwarding state appropriately. Ultimately, the forwarding state associated with a multicast group or source is removed.

A multicast data sender sends data destined for a multicast group. The designated router (DR) of the sender takes those data packets, unicast-encapsulates them, and sends them directly to the RP. The RP receives these encapsulated data packets, de-encapsulates them, and forwards them onto the shared tree. The packets then follow the (*, G) multicast tree state in the devices on the RP tree, being replicated wherever the RP tree branches, and eventually reaching all the receivers for that multicast group. The process of encapsulating data packets to the RP is called registering, and the encapsulation packets are called PIM register packets.

Designated Router

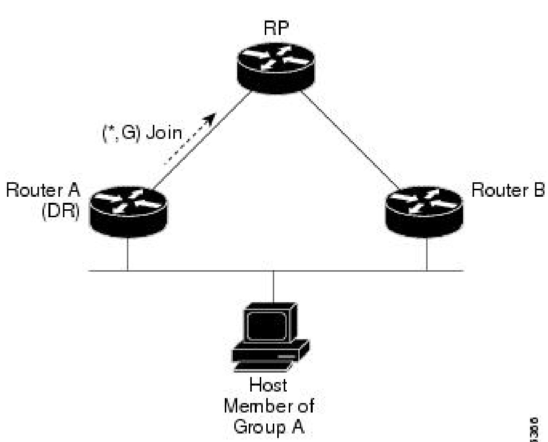
Cisco devices use PIM-SM to forward multicast traffic and follow an election process to select a designated device when there is more than one device on a LAN segment.

The designated router (DR) is responsible for sending PIM register and PIM join and prune messages toward the RP to inform it about active sources and host group membership.

If there are multiple PIM-SM devices on a LAN, a DR must be elected to avoid duplicating multicast traffic for connected hosts. The PIM device with the highest IPv6 address becomes the DR for the LAN unless you choose to force the DR election by use of the **ipv6 pim dr-priority** command. This command allows you to specify the DR priority of each device on the LAN segment (default priority = 1) so that the device with the highest priority will be elected as the DR. If all devices on the LAN segment have the same priority, then the highest IPv6 address is again used as the tiebreaker.

The figure below illustrates what happens on a multiaccess segment. Device A and Device B are connected to a common multiaccess Ethernet segment with Host A as an active receiver for Group A. Only Device A, operating as the DR, sends joins to the RP to construct the shared tree for Group A. If Device B was also permitted to send (*, G) joins to the RP, parallel paths would be created and Host A would receive duplicate multicast traffic. Once Host A begins to source multicast traffic to the group, the DR's responsibility is to send register messages to the RP. If both devices were assigned the responsibility, the RP would receive duplicate multicast packets and result in wastage of bandwidth.

Figure 20: Designated Router Election on a Multiaccess Segment



If the DR should fail, the PIM-SM provides a way to detect the failure of Device A and elect a failover DR. If the DR (Device A) became inoperable, Device B would detect this situation when its neighbor adjacency with Device A timed out. Because Device B has been hearing MLD membership reports from Host A, it already has MLD state for Group A on this interface and would immediately send a join to the RP when it became the new DR. This step reestablishes traffic flow down a new branch of the shared tree via Device B. Additionally, if Host A were sourcing traffic, Device B would initiate a new register process immediately after receiving the next multicast packet from Host A. This action would trigger the RP to join the SPT to Host A via a new branch through Device B.

Tip Two PIM devices are neighbors if there is a direct connection between them. To display your PIM neighbors, use the **show ipv6 pim neighbor** command in privileged EXEC mode.

Note The DR election process is required only on multiaccess LANs.

Rendezvous Point

IPv6 PIM provides embedded RP support. Embedded RP support allows the device to learn RP information using the multicast group destination address instead of the statically configured RP. For devices that are the RP, the device must be statically configured as the RP.

The device searches for embedded RP group addresses in MLD reports or PIM messages and data packets. On finding such an address, the device learns the RP for the group from the address itself. It then uses this learned RP for all protocol activity for the group. For devices that are the RP, the device is advertised as an embedded RP must be configured as the RP.

To select a static RP over an embedded RP, the specific embedded RP group range or mask must be configured in the access list of the static RP. When PIM is configured in sparse mode, you must also choose one or more devices to operate as an RP. An RP is a single common root placed at a chosen point of a shared distribution tree and is configured statically in each box.

PIM DRs forward data from directly connected multicast sources to the RP for distribution down the shared tree. Data is forwarded to the RP in one of two ways:

- Data is encapsulated in register packets and unicast directly to the RP by the first-hop device operating as the DR.
- If the RP has itself joined the source tree, it is multicast-forwarded per the RPF forwarding algorithm described in the PIM-Sparse Mode section.

The RP address is used by first-hop devices to send PIM register messages on behalf of a host sending a packet to the group. The RP address is also used by last-hop devices to send PIM join and prune messages to the RP to inform it about group membership. You must configure the RP address on all devices (including the RP device).

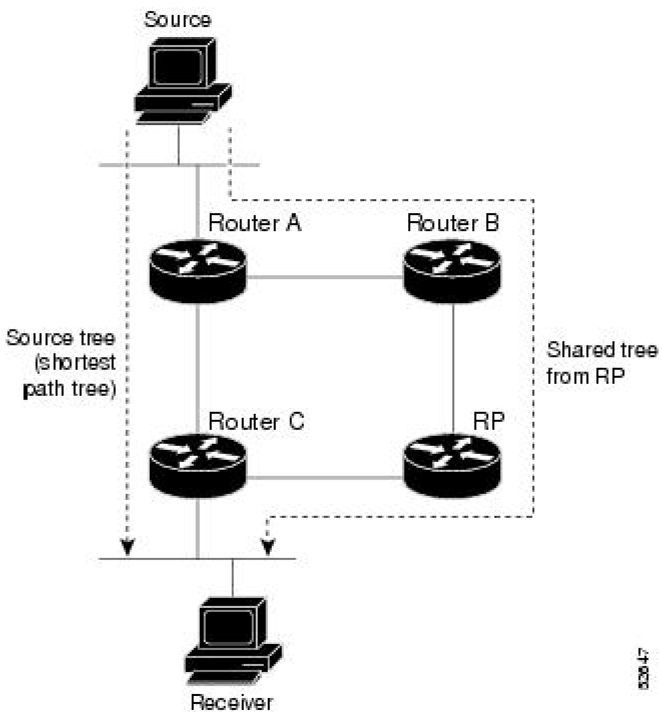
A PIM device can be an RP for more than one group. Only one RP address can be used at a time within a PIM domain for a certain group. The conditions specified by the access list determine for which groups the device is an RP.

IPv6 multicast supports the PIM accept register feature, which is the ability to perform PIM-SM register message filtering at the RP. The user can match an access list or compare the AS path for the registered source with the AS path specified in a route map.

PIM Shared Tree and Source Tree (Shortest-Path Tree)

By default, members of a group receive data from senders to the group across a single data distribution tree rooted at the RP. This type of distribution tree is called shared tree or rendezvous point tree (RPT), as illustrated in the figure below. Data from senders is delivered to the RP for distribution to group members joined to the shared tree.

Figure 21: Shared Tree and Source Tree (Shortest Path Tree)



If the data threshold warrants, leaf devices on the shared tree may initiate a switch to the data distribution tree rooted at the source. This type of distribution tree is called a shortest path tree or source tree. By default, the software switches to a source tree upon receiving the first data packet from a source.

The following process details the move from shared tree to source tree:

1. Receiver joins a group; leaf Device C sends a join message toward the RP.
2. RP puts the link to Device C in its outgoing interface list.
3. Source sends the data; Device A encapsulates the data in the register and sends it to the RP.
4. RP forwards the data down the shared tree to Device C and sends a join message toward the source. At this point, data may arrive twice at Device C, once encapsulated and once natively.
5. When data arrives natively (unencapsulated) at the RP, the RP sends a register-stop message to Device A.
6. By default, receipt of the first data packet prompts Device C to send a join message toward the source.
7. When Device C receives data on (S, G), it sends a prune message for the source up the shared tree.
8. RP deletes the link to Device C from the outgoing interface of (S, G).
9. RP triggers a prune message toward the source.

Join and prune messages are sent for sources and RPs. They are sent hop-by-hop and are processed by each PIM device along the path to the source or RP. Register and register-stop messages are not sent hop-by-hop. They are sent by the designated router (DR) that is directly connected to a source and are received by the RP for the group.

Reverse Path Forwarding

Reverse-path forwarding is used for forwarding multicast datagrams. It functions as follows:

- If a device receives a datagram on an interface it uses to send unicast packets to the source, the packet has arrived on the RPF interface.
- If the packet arrives on the RPF interface, a device forwards the packet out the interfaces present in the outgoing interface list of a multicast routing table entry.
- If the packet does not arrive on the RPF interface, the packet is silently discarded to prevent loops.

PIM uses both source trees and RP-rooted shared trees to forward datagrams; the RPF check is performed differently for each, as follows:

- If a PIM device has source-tree state (that is, an (S, G) entry is present in the multicast routing table), the device performs the RPF check against the IPv6 address of the source of the multicast packet.
- If a PIM device has shared-tree state (and no explicit source-tree state), it performs the RPF check on the RP's address (which is known when members join the group).

Sparse-mode PIM uses the RPF lookup function to determine where it needs to send joins and prunes. (S, G) joins (which are source-tree states) are sent toward the source. (*, G) joins (which are shared-tree states) are sent toward the RP.

Note To do a RPF check, use the **show ipv6 rpf hostname** or **show ipv6 rpf vrf vrf_name hostname** command.

How to Configure IPv6 Multicast PIM Sparse Mode

Enabling IPv6 Multicast Routing

IPv6 multicast uses MLD version 2. This version of MLD is fully backward-compatible with MLD version 1 (described in *RFC 2710*). Hosts that support only MLD version 1 will interoperate with a device running MLD version 2. Mixed LANs with both MLD version 1 and MLD version 2 hosts are likewise supported.

Before you begin

You must first enable IPv6 unicast routing on all interfaces of the device on which you want to enable IPv6 multicast routing .

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ipv6 multicast-routing** [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 multicast-routing** [**vrf** *vrf-name*] **Example:** `Device(config)# ipv6 multicast-routing`	Enables multicast routing on all IPv6-enabled interfaces and enables multicast forwarding for PIM and MLD on all enabled interfaces of the device. • IPv6 multicast routing is disabled by default when IPv6 unicast routing is enabled. IPv6 multicast-routing needs to be enabled for IPv6 multicast routing to function.

Configuring PIM-SM and Displaying PIM-SM Information for a Group Range

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim** [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-access-list*] [**bidir**]
4. **end**
5. **show ipv6 pim** [**vrf** *vrf-name*] **interface** [**state-on**] [**state-off**] [*type number*]
6. **show ipv6 pim** [**vrf** *vrf-name*] **group-map** [*group-name* | *group-address*] | [*group-range* | *group-mask*] [**info-source** {**bsr** | **default** | **embedded-rp** | **static**}]
7. **show ipv6 pim** [**vrf** *vrf-name*] **neighbor** [**detail**] [*interface-type interface-number* | **count**]
8. **show ipv6 pim** [**vrf** *vrf-name*] **range-list**[**config**] [*rp-address* | *rp-name*]
9. **show ipv6 pim** [**vrf** *vrf-name*] **tunnel** [*interface-type interface-number*]
10. **debug ipv6 pim** [*group-name* | *group-address* | **interface** *interface-type* | **bsr** | **group** | **mvpn** | **neighbor**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim** [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-access-list*] [**bidir**] **Example:** `Device(config)# ipv6 pim rp-address 2001:DB8::01:800:200E:8C6C acc-grp-1`	Configures the address of a PIM RP for a particular group range.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 5	**show ipv6 pim** [**vrf** *vrf-name*] **interface** [**state-on**] [**state-off**] [*type number*] **Example:** `Device# show ipv6 pim interface`	Displays information about interfaces configured for PIM.
Step 6	**show ipv6 pim** [**vrf** *vrf-name*] **group-map** [*group-name* | *group-address*] | [*group-range* | *group-mask*] [**info-source** {**bsr** | **default** | **embedded-rp** | **static**}] **Example:** `Device# show ipv6 pim group-map`	Displays an IPv6 multicast group mapping table.
Step 7	**show ipv6 pim** [**vrf** *vrf-name*] **neighbor** [**detail**] [*interface-type interface-number* | **count**] **Example:** `Device# show ipv6 pim neighbor`	Displays the PIM neighbors discovered by the Cisco IOS software.
Step 8	**show ipv6 pim** [**vrf** *vrf-name*] **range-list**[**config**] [*rp-address* | *rp-name*] **Example:** `Device# show ipv6 pim range-list`	Displays information about IPv6 multicast range lists.
Step 9	**show ipv6 pim** [**vrf** *vrf-name*] **tunnel** [*interface-type interface-number*] **Example:**	Displays information about the PIM register encapsulation and de-encapsulation tunnels on an interface.

	Command or Action	Purpose
	`Device# show ipv6 pim tunnel`	
Step 10	**debug ipv6 pim** [*group-name* | *group-address* | **interface** *interface-type* | **bsr** | **group** | **mvpn** | **neighbor**] **Example:** `Device# debug ipv6 pim`	Enables debugging on PIM protocol activity.

Configuring PIM Options

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim** [**vrf** *vrf-name*] **spt-threshold infinity** [**group-list** *access-list-name*]
4. **ipv6 pim** [**vrf** *vrf-name*] **accept-register** {**list** *access-list* | **route-map** *map-name*}
5. **interface** *type number*
6. **ipv6 pim dr-priority** *value*
7. **ipv6 pim hello-interval** *seconds*
8. **ipv6 pim join-prune-interval** *seconds*
9. **exit**
10. **show ipv6 pim** [**vrf** *vrf-name*] **join-prune statistic** [*interface-type*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim** [**vrf** *vrf-name*] **spt-threshold infinity** [**group-list** *access-list-name*] **Example:** `Device(config)# ipv6 pim spt-threshold infinity group-list acc-grp-1`	Configures when a PIM leaf device joins the SPT for the specified groups.
Step 4	**ipv6 pim** [**vrf** *vrf-name*] **accept-register** {**list** *access-list* | **route-map** *map-name*}	Accepts or rejects registers at the RP.

	Command or Action	Purpose
	Example: `Device(config)# ipv6 pim accept-register route-map reg-filter`	
Step 5	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 6	**ipv6 pim dr-priority** *value* **Example:** `Device(config-if)# ipv6 pim dr-priority 3`	Configures the DR priority on a PIM device.
Step 7	**ipv6 pim hello-interval** *seconds* **Example:** `Device(config-if)# ipv6 pim hello-interval 45`	Configures the frequency of PIM hello messages on an interface.
Step 8	**ipv6 pim join-prune-interval** *seconds* **Example:** `Device(config-if)# ipv6 pim join-prune-interval 75`	Configures periodic join and prune announcement intervals for a specified interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 10	**show ipv6 pim** [**vrf** *vrf-name*] **join-prune statistic** [*interface-type*] **Example:** `Device# show ipv6 pim join-prune statistic`	Displays the average join-prune aggregation for the most recently aggregated packets for each interface.

Resetting the PIM Traffic Counters

If PIM malfunctions, or in order to verify that the expected number of PIM packets are received and sent, clear PIM traffic counters. Once the traffic counters are cleared, you can verify that PIM is functioning correctly and that PIM packets are being received and sent correctly.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 pim** [**vrf** *vrf-name*] **traffic**
3. **show ipv6 pim** [**vrf** *vrf-name*] **traffic**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 pim** [**vrf** *vrf-name*] **traffic** **Example:** `Device# clear ipv6 pim traffic`	Resets the PIM traffic counters.
Step 3	**show ipv6 pim** [**vrf** *vrf-name*] **traffic** **Example:** `Device# show ipv6 pim traffic`	Displays the PIM traffic counters.

Turning Off IPv6 PIM on a Specified Interface

A user might want only specified interfaces to perform IPv6 multicast and will therefore want to turn off PIM on a specified interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ipv6 pim**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number, and places the device in interface configuration mode.

	Command or Action	Purpose
Step 4	**no ipv6 pim** **Example:** `Device(config-if)# no ipv6 pim`	Turns off IPv6 PIM on a specified interface.

Configuration Examples for IPv6 Multicast PIM Sparse Mode

Example: Enabling IPv6 Multicast Routing

The following example enables multicast routing on all interfaces and also enables multicast forwarding for PIM and MLD on all enabled interfaces of the device.

```
Device> enable
Device# configure terminal
Device(config)# ipv6 multicast-routing
```

Example: Configuring PIM

The following example shows how to configure a device to use PIM-SM using 2001:DB8::1 as the RP. It sets the SPT threshold to infinity to prevent switchover to the source tree when a source starts sending traffic and sets a filter on all sources that do not have a local multicast BGP prefix.

```
Device(config)# ipv6 multicast-routing
Device(config)# ipv6 pim rp-address 2001:DB8::1
Device(config)# ipv6 pim spt-threshold infinity
Device(config)# ipv6 pim accept-register route-map reg-filter
```

Example: Displaying IPv6 PIM Topology Information

```
Device# show ipv6 pim topology

IP PIM Multicast Topology Table
Entry state:(*/S,G)[RPT/SPT] Protocol Uptime Info
Entry flags:KAT - Keep Alive Timer, AA - Assume Alive, PA - Probe Alive,
    RA - Really Alive, LH - Last Hop, DSS - Don't Signal Sources,
    RR - Register Received, SR - Sending Registers, E - MSDP External,
    DCC - Don't Check Connected
Interface state:Name, Uptime, Fwd, Info
Interface flags:LI - Local Interest, LD - Local Dissinterest,
II - Internal Interest, ID - Internal Dissinterest,
LH - Last Hop, AS - Assert, AB - Admin Boundary

(*,FF05::1)
SM UP:02:26:56 JP:Join(now) Flags:LH
RP:2001:DB8:1:1:2
RPF:Ethernet1/1,FE81::1
  Ethernet0/1          02:26:56  fwd LI LH

(2001:DB8:1:1:200,FF05::1)
SM UP:00:00:07 JP:Null(never) Flags:
```

```
RPF:Ethernet1/1,FE80::30:1:4
  Ethernet1/1          00:00:07  off LI
```

Example: Displaying PIM-SM Information for a Group Range

This example displays information about interfaces configured for PIM:

```
Device# show ipv6 pim interface state-on

Interface          PIM  Nbr   Hello  DR
                        Count Intvl  Prior

Ethernet0          on   0     30     1
    Address:FE80::208:20FF:FE08:D7FF
    DR     :this system
POS1/0             on   0     30     1
    Address:FE80::208:20FF:FE08:D554
    DR     :this system
POS4/0             on   1     30   1
    Address:FE80::208:20FF:FE08:D554
    DR     :FE80::250:E2FF:FE8B:4C80
POS4/1             on   0     30   1
    Address:FE80::208:20FF:FE08:D554
    DR     :this system
Loopback0          on   0     30     1
    Address:FE80::208:20FF:FE08:D554
    DR     :this system
```

This example displays an IPv6 multicast group mapping table:

```
Device# show ipv6 pim group-map

FF33::/32*
      SSM
      Info source:Static
      Uptime:00:08:32, Groups:0
  FF34::/32*
      SSM
      Info source:Static
      Uptime:00:09:42, Groups:0
```

This example displays information about IPv6 multicast range lists:

```
Device# show ipv6 pim range-list

config SSM Exp:never Learnt from :::
 FF33::/32 Up:00:26:33
 FF34::/32 Up:00:26:33
 FF35::/32 Up:00:26:33
 FF36::/32 Up:00:26:33
 FF37::/32 Up:00:26:33
 FF38::/32 Up:00:26:33
 FF39::/32 Up:00:26:33
 FF3A::/32 Up:00:26:33
 FF3B::/32 Up:00:26:33
 FF3C::/32 Up:00:26:33
 FF3D::/32 Up:00:26:33
 FF3E::/32 Up:00:26:33
 FF3F::/32 Up:00:26:33
config SM RP:40::1:1:1 Exp:never Learnt from :::
 FF13::/64 Up:00:03:50
```

```
config SM RP:40::1:1:3 Exp:never Learnt from :::
 FF09::/64 Up:00:03:50
```

Example: Configuring PIM Options

The following example sets the DR priority, the PIM hello interval, and the periodic join and prune announcement interval on Ethernet interface 0/0.

```
Device(config)# interface Ethernet0/0
Device(config)# ipv6 pim hello-interval 60
Device(config)# ipv6 pim dr-priority 3
```

Example: Displaying Information About PIM Traffic

```
Device# show ipv6 pim traffic

PIM Traffic Counters
Elapsed time since counters cleared:00:05:29

                              Received     Sent
Valid PIM Packets               22          22
Hello                           22          22
Join-Prune                      0           0
Register                        0           0
Register Stop                   0           0
Assert                          0           0
Bidir DF Election               0           0

Errors:
Malformed Packets                           0
Bad Checksums                               0
Send Errors                                 0
Packet Sent on Loopback Errors              0
Packets Received on PIM-disabled Interface  0
Packets Received with Unknown PIM Version   0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Multicast PIM Sparse Mode

CHAPTER 17

IPv6 Multicast: Static Multicast Routing for IPv6

IPv6 static multicast routes, or mroutes, share the same database as IPv6 static routes and are implemented by extending static route support for reverse path forwarding (RPF) checks.

Information About IPv6 Static Mroutes

IPv6 static mroutes behave much in the same way as IPv4 static mroutes used to influence the RPF check. IPv6 static mroutes share the same database as IPv6 static routes and are implemented by extending static route support for RPF checks. Static mroutes support equal-cost multipath mroutes, and they also support unicast-only static routes.

How to Configure IPv6 Static Multicast Routes

Configuring Static Mroutes

Static multicast routes (mroutes) in IPv6 can be implemented as an extension of IPv6 static routes. You can configure your device to use a static route for unicast routing only, to use a static multicast route for multicast RPF selection only, or to use a static route for both unicast routing and multicast RPF selection.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** *ipv6-prefix* **/** *prefix-length* *ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | **unicast**| **multicast**] [**tag** *tag*
4. **end**
5. **show ipv6 mroute** [**vrf** *vrf-name*] [**link-local** | [*group-name* | *group-address* [*source-address* | *source-name*]] [**summary**] [**count**]

6. **show ipv6 mroute** [**vrf** *vrf-name*] [**link-local** | *group-name* | *group-address*] **active**[*kbps*]
7. **show ipv6 rpf** [**vrf** *vrf-name*] *ipv6-prefix*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route** *ipv6-prefix* **/** *prefix-length* *ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | **unicast**| **multicast**] [**tag** *tag* **Example:** `Device(config)# ipv6 route 2001:DB8::/64 6::6 100`	Establishes static IPv6 routes. The example shows a static route used for both unicast routing and multicast RPF selection.
Step 4	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.
Step 5	**show ipv6 mroute** [**vrf** *vrf-name*] [**link-local** | [*group-name* | *group-address* [*source-address* | *source-name*]] [**summary**] [**count**] **Example:** `Device# show ipv6 mroute ff07::1`	Displays the contents of the IPv6 multicast routing table.
Step 6	**show ipv6 mroute** [**vrf** *vrf-name*] [**link-local** | *group-name* | *group-address*] **active**[*kbps*] **Example:** `Device# show ipv6 mroute active`	Displays the active multicast streams on the device.
Step 7	**show ipv6 rpf** [**vrf** *vrf-name*] *ipv6-prefix* **Example:** `Device# show ipv6 rpf 2001:DB8::1:1:2`	Checks RPF information for a given unicast host address and prefix.

Configuration Examples for IPv6 Static Multicast Routes

Example: Configuring Static Mroutes

Using the **show ipv6 mroute** command allows you to verify that multicast IPv6 data is flowing:

```
Device# show ipv6 mroute ff07::1

Multicast Routing Table
Flags:D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT
Timers:Uptime/Expires
Interface state:Interface, State

(*, FF07::1), 00:04:45/00:02:47, RP 2001:DB8:6::6, flags:S
  Incoming interface:Tunnel5
  RPF nbr:6:6:6::6
  Outgoing interface list:
    POS4/0, Forward, 00:04:45/00:02:47

(2001:DB8:999::99, FF07::1), 00:02:06/00:01:23, flags:SFT
  Incoming interface:POS1/0
  RPF nbr:2001:DB8:999::99
  Outgoing interface list:
    POS4/0, Forward, 00:02:06/00:03:27
```

The following sample output displays information from the **show ipv6 mroute active** command:

```
Device# show ipv6 mroute active

Active IPv6 Multicast Sources - sending >= 4 kbps
Group:FF05::1
 Source:2001:DB8:1:1:1
   Rate:11 pps/8 kbps(1sec), 8 kbps(last 8 sec)
```

The following example displays RPF information for the unicast host with the IPv6 address of 2001:DB8:1:1:2:

```
Device# show ipv6 rpf 2001:DB8:1:1:2

RPF information for 2001:DB8:1:1:2
  RPF interface:Ethernet3/2
  RPF neighbor:FE80::40:1:3
  RPF route/mask:20::/64
  RPF type:Unicast
  RPF recursion count:0
  Metric preference:110
  Metric:30
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Multicast: Static Multicast Routing for IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 18

IPv6 Multicast: PIM Source-Specific Multicast

The PIM source-specific multicast (SSM) routing protocol supports SSM implementation and is derived from PIM-SM. However, unlike PIM-SM data from all multicast sources are sent when there is a PIM join, the SSM feature forwards datagram traffic to receivers from only those multicast sources that the receivers have explicitly joined, thus optimizing bandwidth utilization and denying unwanted Internet broadcast traffic.

Prerequisites for IPv6 Multicast: PIM Source-Specific Multicast

- Multicast Listener Discovery (MLD) version 2 is required for source-specific multicast (SSM) to operate.
- Before SSM will run with MLD, SSM must be supported by the Cisco IPv6 device, the host where the application is running, and the application itself.

Information About IPv6 Multicast: PIM Source-Specific Multicast

IPv6 Multicast Routing Implementation

Cisco software supports the following protocols to implement IPv6 multicast routing:

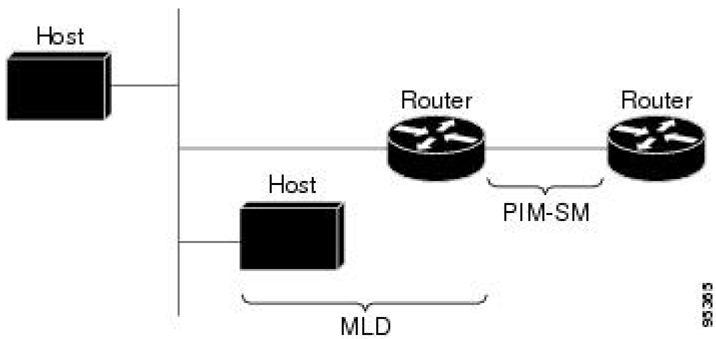
- MLD is used by IPv6 devices to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. There are two versions of MLD:
  - MLD version 1 is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4.
  - MLD version 2 is based on version 3 of the IGMP for IPv4.
- IPv6 multicast for Cisco software uses both MLD version 2 and MLD version 1. MLD version 2 is fully backward-compatible with MLD version 1 (described in RFC 2710). Hosts that support only MLD

version 1 will interoperate with a device running MLD version 2. Mixed LANs with both MLD version 1 and MLD version 2 hosts are likewise supported.

- PIM-SM is used between devices so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- PIM in Source Specific Multicast (PIM-SSM) is similar to PIM-SM with the additional ability to report interest in receiving packets from specific source addresses (or from all but the specific source addresses) to an IP multicast address.

The figure below shows where MLD and PIM-SM operate within the IPv6 multicast environment.

Figure 22: IPv6 Multicast Routing Protocols Supported for IPv6



Protocol Independent Multicast

Protocol Independent Multicast (PIM) is used between devices so that they can track which multicast packets to forward to each other and to their directly connected LANs. PIM works independently of the unicast routing protocol to perform send or receive multicast route updates like other protocols. Regardless of which unicast routing protocols are being used in the LAN to populate the unicast routing table, Cisco IOS PIM uses the existing unicast table content to perform the Reverse Path Forwarding (RPF) check instead of building and maintaining its own separate routing table.

You can configure IPv6 multicast to use either a PIM- Sparse Mode (SM) or PIM-Source Specific Multicast (SSM) operation, or you can use both PIM-SM and PIM-SSM together in your network.

PIM-Source Specific Multicast

PIM-SSM is the routing protocol that supports the implementation of SSM and is derived from PIM-SM. However, unlike PIM-SM where data from all multicast sources are sent when there is a PIM join, the SSM feature forwards datagram traffic to receivers from only those multicast sources that the receivers have explicitly joined, thus optimizing bandwidth utilization and denying unwanted Internet broadcast traffic. Further, instead of the use of RP and shared trees, SSM uses information found on source addresses for a multicast group. This information is provided by receivers through the source addresses relayed to the last-hop devices by MLD membership reports, resulting in shortest-path trees directly to the sources.

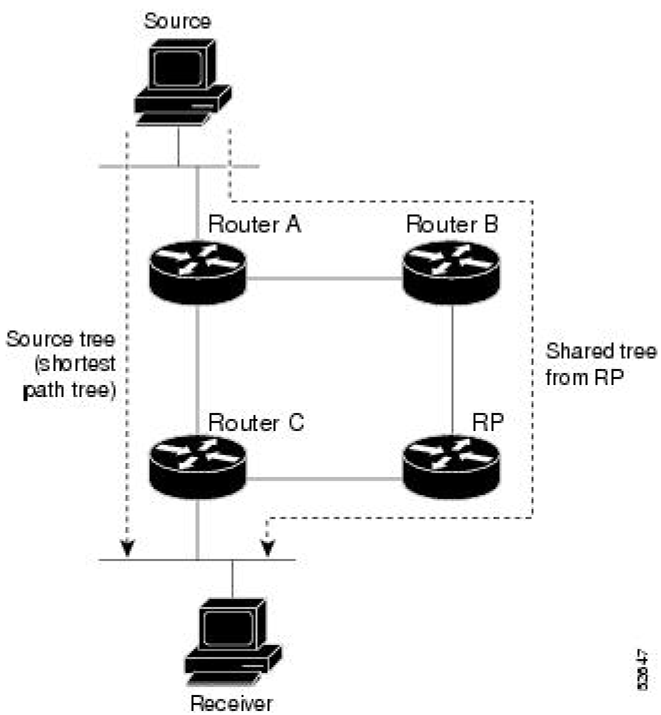
In SSM, delivery of datagrams is based on (S, G) channels. Traffic for one (S, G) channel consists of datagrams with an IPv6 unicast source address S and the multicast group address G as the IPv6 destination address. Systems will receive this traffic by becoming members of the (S, G) channel. Signaling is not required, but receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources.

MLD version 2 is required for SSM to operate. MLD allows the host to provide source information. Before SSM will run with MLD, SSM must be supported in the Cisco IPv6 device, the host where the application is running, and the application itself.

PIM Shared Tree and Source Tree (Shortest-Path Tree)

By default, members of a group receive data from senders to the group across a single data distribution tree rooted at the RP. This type of distribution tree is called shared tree or rendezvous point tree (RPT), as illustrated in the figure below. Data from senders is delivered to the RP for distribution to group members joined to the shared tree.

Figure 23: Shared Tree and Source Tree (Shortest Path Tree)



If the data threshold warrants, leaf devices on the shared tree may initiate a switch to the data distribution tree rooted at the source. This type of distribution tree is called a shortest path tree or source tree. By default, the software switches to a source tree upon receiving the first data packet from a source.

The following process details the move from shared tree to source tree:

1. Receiver joins a group; leaf Device C sends a join message toward the RP.
2. RP puts the link to Device C in its outgoing interface list.
3. Source sends the data; Device A encapsulates the data in the register and sends it to the RP.
4. RP forwards the data down the shared tree to Device C and sends a join message toward the source. At this point, data may arrive twice at Device C, once encapsulated and once natively.
5. When data arrives natively (unencapsulated) at the RP, the RP sends a register-stop message to Device A.
6. By default, receipt of the first data packet prompts Device C to send a join message toward the source.

7. When Device C receives data on (S, G), it sends a prune message for the source up the shared tree.
8. RP deletes the link to Device C from the outgoing interface of (S, G).
9. RP triggers a prune message toward the source.

Join and prune messages are sent for sources and RPs. They are sent hop-by-hop and are processed by each PIM device along the path to the source or RP. Register and register-stop messages are not sent hop-by-hop. They are sent by the designated router (DR) that is directly connected to a source and are received by the RP for the group.

Reverse Path Forwarding

Reverse-path forwarding is used for forwarding multicast datagrams. It functions as follows:

- If a device receives a datagram on an interface it uses to send unicast packets to the source, the packet has arrived on the RPF interface.
- If the packet arrives on the RPF interface, a device forwards the packet out the interfaces present in the outgoing interface list of a multicast routing table entry.
- If the packet does not arrive on the RPF interface, the packet is silently discarded to prevent loops.

PIM uses both source trees and RP-rooted shared trees to forward datagrams; the RPF check is performed differently for each, as follows:

- If a PIM device has source-tree state (that is, an (S, G) entry is present in the multicast routing table), the device performs the RPF check against the IPv6 address of the source of the multicast packet.
- If a PIM device has shared-tree state (and no explicit source-tree state), it performs the RPF check on the RP's address (which is known when members join the group).

Sparse-mode PIM uses the RPF lookup function to determine where it needs to send joins and prunes. (S, G) joins (which are source-tree states) are sent toward the source. (*, G) joins (which are shared-tree states) are sent toward the RP.

Note To do a RPF check, use the **show ipv6 rpf hostname** or **show ipv6 rpf vrf vrf_name hostname** command.

How to Configure IPv6 Multicast: PIM Source-Specific Multicast

Configuring PIM Options

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim** [**vrf** *vrf-name*] **spt-threshold infinity** [**group-list** *access-list-name*]
4. **ipv6 pim** [**vrf** *vrf-name*] **accept-register** {**list** *access-list* | **route-map** *map-name*}
5. **interface** *type number*

6. **ipv6 pim dr-priority** *value*
7. **ipv6 pim hello-interval** *seconds*
8. **ipv6 pim join-prune-interval** *seconds*
9. **exit**
10. **show ipv6 pim** [**vrf** *vrf-name*] **join-prune statistic** [*interface-type*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim** [**vrf** *vrf-name*] **spt-threshold infinity** [**group-list** *access-list-name*] **Example:** `Device(config)# ipv6 pim spt-threshold infinity group-list acc-grp-1`	Configures when a PIM leaf device joins the SPT for the specified groups.
Step 4	**ipv6 pim** [**vrf** *vrf-name*] **accept-register** {**list** *access-list* | **route-map** *map-name*} **Example:** `Device(config)# ipv6 pim accept-register route-map reg-filter`	Accepts or rejects registers at the RP.
Step 5	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 6	**ipv6 pim dr-priority** *value* **Example:** `Device(config-if)# ipv6 pim dr-priority 3`	Configures the DR priority on a PIM device.
Step 7	**ipv6 pim hello-interval** *seconds* **Example:** `Device(config-if)# ipv6 pim hello-interval 45`	Configures the frequency of PIM hello messages on an interface.

	Command or Action	Purpose
Step 8	**ipv6 pim join-prune-interval** *seconds* **Example:** `Device(config-if)# ipv6 pim join-prune-interval 75`	Configures periodic join and prune announcement intervals for a specified interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 10	**show ipv6 pim** [**vrf** *vrf-name*] **join-prune statistic** [*interface-type*] **Example:** `Device# show ipv6 pim join-prune statistic`	Displays the average join-prune aggregation for the most recently aggregated packets for each interface.

Resetting the PIM Traffic Counters

If PIM malfunctions, or in order to verify that the expected number of PIM packets are received and sent, clear PIM traffic counters. Once the traffic counters are cleared, you can verify that PIM is functioning correctly and that PIM packets are being received and sent correctly.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 pim** [**vrf** *vrf-name*] **traffic**
3. **show ipv6 pim** [**vrf** *vrf-name*] **traffic**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 pim** [**vrf** *vrf-name*] **traffic** **Example:** `Device# clear ipv6 pim traffic`	Resets the PIM traffic counters.
Step 3	**show ipv6 pim** [**vrf** *vrf-name*] **traffic** **Example:** `Device# show ipv6 pim traffic`	Displays the PIM traffic counters.

Clearing the PIM Topology Table to Reset the MRIB Connection

No configuration is necessary to use the MRIB. However, users may in certain situations want to clear the PIM topology table in order to reset the MRIB connection and verify MRIB information.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 pim** [**vrf** *vrf-name*] **topology** [*group-name* | *group-address*]
3. **show ipv6 mrib** [**vrf** *vrf-name*] **client** [**filter**] [**name** {*client-name* | *client-name* **:** *client-id*}]
4. **show ipv6 mrib** [**vrf** *vrf-name*] **route** [**link-local**| **summary** | [*sourceaddress-or-name* | *****] [*groupname-or-address* [*prefix-length*]]]
5. **show ipv6 pim** [**vrf** *vrf-name*] **topology** [*groupname-or-address* [*sourcename-or-address*] | **link-local** | **route-count** [**detail**]]
6. **debug ipv6 mrib** [**vrf** *vrf-name*] **client**
7. **debug ipv6 mrib** [**vrf** *vrf-name*] **io**
8. **debug ipv6 mrib proxy**
9. **debug ipv6 mrib** [**vrf** *vrf-name*] **route** [*group-name* | *group-address*]
10. **debug ipv6 mrib** [**vrf** *vrf-name*] **table**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 pim** [**vrf** *vrf-name*] **topology** [*group-name* | *group-address*] **Example:** `Device# clear ipv6 pim topology FF04::10`	Clears the PIM topology table.
Step 3	**show ipv6 mrib** [**vrf** *vrf-name*] **client** [**filter**] [**name** {*client-name* | *client-name* **:** *client-id*}] **Example:** `Device# show ipv6 mrib client`	Displays multicast-related information about an interface.
Step 4	**show ipv6 mrib** [**vrf** *vrf-name*] **route** [**link-local**| **summary** | [*sourceaddress-or-name* | *****] [*groupname-or-address* [*prefix-length*]]] **Example:** `Device# show ipv6 mrib route`	Displays the MRIB route information.

	Command or Action	Purpose
Step 5	**show ipv6 pim** [**vrf** *vrf-name*] **topology** [*groupname-or-address* [*sourcename-or-address*] | **link-local** | **route-count** [**detail**]] **Example:** `Device# show ipv6 pim topology`	Displays PIM topology table information for a specific group or all groups.
Step 6	**debug ipv6 mrib** [**vrf** *vrf-name*] **client** **Example:** `Device# debug ipv6 mrib client`	Enables debugging on MRIB client management activity.
Step 7	**debug ipv6 mrib** [**vrf** *vrf-name*] **io** **Example:** `Device# debug ipv6 mrib io`	Enables debugging on MRIB I/O events.
Step 8	**debug ipv6 mrib proxy** **Example:** `Device# debug ipv6 mrib proxy`	Enables debugging on MRIB proxy activity between the route processor and line cards on distributed router platforms.
Step 9	**debug ipv6 mrib** [**vrf** *vrf-name*] **route** [*group-name* | *group-address*] **Example:** `Device# debug ipv6 mrib route`	Displays information about MRIB routing entry-related activity.
Step 10	**debug ipv6 mrib** [**vrf** *vrf-name*] **table** **Example:** `Device# debug ipv6 mrib table`	Enables debugging on MRIB table management activity.

Configuration Examples for IPv6 Multicast: PIM Source-Specific Multicast

Example: Displaying IPv6 PIM Topology Information

```
Device# show ipv6 pim topology

IP PIM Multicast Topology Table
Entry state:(*/S,G)[RPT/SPT] Protocol Uptime Info
Entry flags:KAT - Keep Alive Timer, AA - Assume Alive, PA - Probe Alive,
    RA - Really Alive, LH - Last Hop, DSS - Don't Signal Sources,
```

```
    RR - Register Received, SR - Sending Registers, E - MSDP External,
    DCC - Don't Check Connected
Interface state:Name, Uptime, Fwd, Info
Interface flags:LI - Local Interest, LD - Local Dissinterest,
II - Internal Interest, ID - Internal Dissinterest,
LH - Last Hop, AS - Assert, AB - Admin Boundary

(*,FF05::1)
SM UP:02:26:56 JP:Join(now) Flags:LH
RP:2001:DB8:1:1:2
RPF:Ethernet1/1,FE81::1
  Ethernet0/1          02:26:56  fwd LI LH

(2001:DB8:1:1:200,FF05::1)
SM UP:00:00:07 JP:Null(never) Flags:
RPF:Ethernet1/1,FE80::30:1:4
  Ethernet1/1          00:00:07  off LI
```

Example: Configuring Join/Prune Aggregation

The following example shows how to provide the join/prune aggregation on Ethernet interface 0/0:

```
Device# show ipv6 pim join-prune statistic Ethernet0/0

PIM Average Join/Prune Aggregation for last (1K/10K/50K) packets
Interface              Transmitted             Received
Ethernet0/0      0    / 0              1    / 0
```

Example: Displaying Information About PIM Traffic

```
Device# show ipv6 pim traffic

PIM Traffic Counters
Elapsed time since counters cleared:00:05:29

                              Received     Sent
Valid PIM Packets               22          22
Hello                           22          22
Join-Prune                      0           0
Register                        0           0
Register Stop                   0           0
Assert                          0           0
Bidir DF Election               0           0

Errors:
Malformed Packets                           0
Bad Checksums                               0
Send Errors                                 0
Packet Sent on Loopback Errors              0
Packets Received on PIM-disabled Interface  0
Packets Received with Unknown PIM Version   0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Multicast: PIM Source-Specific Multicast

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 19

IPv6 Source Specific Multicast Mapping

Source-specific multicast (SSM) SSM mapping for IPv6 supports both static and dynamic Domain Name System (DNS) mapping for MLD version 1 receivers. This feature allows deployment of IPv6 SSM with hosts that are incapable of providing MLD version 2 support in their TCP/IP host stack and their IP multicast receiving application.

Information About IPv6 Source Specific Multicast Mapping

SSM mapping for IPv6 supports both static and dynamic Domain Name System (DNS) mapping for MLD version 1 receivers. This feature allows deployment of IPv6 SSM with hosts that are incapable of providing MLD version 2 support in their TCP/IP host stack and their IP multicast receiving application.

SSM mapping allows the device to look up the source of a multicast MLD version 1 report either in the running configuration of the device or from a DNS server. The device can then initiate an (S, G) join toward the source.

How to Configure IPv6 Source Specific Multicast Mapping

Configuring IPv6 SSM

When the SSM mapping feature is enabled, DNS-based SSM mapping is automatically enabled, which means that the device will look up the source of a multicast MLD version 1 report from a DNS server.

You can configure either DNS-based or static SSM mapping, depending on your device configuration. If you choose to use static SSM mapping, you can configure multiple static SSM mappings. If multiple static SSM mappings are configured, the source addresses of all matching access lists will be used.

Before you begin

> **Note** To use DNS-based SSM mapping, the device needs to find at least one correctly configured DNS server to which the device can be directly attached.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 mld** [**vrf** *vrf-name*] **ssm-map enable**
4. **no ipv6 mld** [**vrf** *vrf-name*] **ssm-map query dns**
5. **ipv6 mld** [**vrf** *vrf-name*] **ssm-map static** *access-list source-address*
6. **end**
7. **show ipv6 mld** [**vrf** *vrf-name*] **ssm-map** [*source-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 mld** [**vrf** *vrf-name*] **ssm-map enable** **Example:** `Device(config)# ipv6 mld ssm-map enable`	Enables the SSM mapping feature for groups in the configured SSM range.
Step 4	**no ipv6 mld** [**vrf** *vrf-name*] **ssm-map query dns** **Example:** `Device(config)# no ipv6 mld ssm-map query dns`	Disables DNS-based SSM mapping.
Step 5	**ipv6 mld** [**vrf** *vrf-name*] **ssm-map static** *access-list source-address* **Example:** `Device(config)# ipv6 mld ssm-map static SSM_MAP_ACL_2 2001:DB8:1::1`	Configures static SSM mappings.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show ipv6 mld** [**vrf** *vrf-name*] **ssm-map** [*source-address*] **Example:** `Device# show ipv6 mld ssm-map`	Displays SSM mapping information.

Configuration Examples for IPv6 Source Specific Multicast Mapping

Example: IPv6 SSM Mapping

```
Device# show ipv6 mld ssm-map 2001:DB8::1

 Group address   : 2001:DB8::1
 Group mode ssm : TRUE
 Database        : STATIC
 Source list     : 2001:DB8::2
                   2001:DB8::3

Device# show ipv6 mld ssm-map 2001:DB8::2

 Group address   : 2001:DB8::2
 Group mode ssm : TRUE
 Database        : DNS
 Source list     : 2001:DB8::3
                   2001:DB8::1
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference

Related Topic	Document Title
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Source Specific Multicast Mapping

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 20

IPv6 Multicast: Explicit Tracking of Receivers

Information About IPv6 Multicast Explicit Tracking of Receivers

Explicit Tracking of Receivers

The explicit tracking feature allows a device to track the behavior of the hosts within its IPv6 network. This feature also enables the fast leave mechanism to be used with MLD version 2 host reports.

How to Configure IPv6 Multicast Explicit Tracking of Receivers

Configuring Explicit Tracking of Receivers to Track Host Behavior

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 mld explicit-tracking** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 mld explicit-tracking** *access-list-name* **Example:** `Device(config-if)# ipv6 mld explicit-tracking list1`	Enables explicit tracking of hosts.

Configuration Examples for IPv6 Multicast Explicit Tracking of Receivers

Example: Configuring Explicit Tracking of Receivers

```
Device> enable
Device# configure terminal
Device(config)# interface FastEthernet 1/0
Device(config-if)# ipv6 mld explicit-tracking list1
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Multicast: Explicit Tracking of Receivers

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 21

IPv6 Bidirectional PIM

Restrictions for IPv6 Bidirectional PIM

When the bidirectional (bidir) range is used in a network, all devices in that network must be able to understand the bidirectional range in the bootstrap message (BSM).

Information About IPv6 Bidirectional PIM

Bidirectional PIM

Bidirectional PIM allows multicast devices to keep reduced state information, as compared with unidirectional shared trees in PIM-SM. Bidirectional shared trees convey data from sources to the RPA and distribute them from the RPA to the receivers. Unlike PIM-SM, bidirectional PIM does not switch over to the source tree, and there is no register encapsulation of data from the source to the RP.

A single designated forwarder (DF) exists for each RPA on every link within a bidirectional PIM domain (including multiaccess and point-to-point links). The only exception is the RPL on which no DF exists. The DF is the device on the link with the best route to the RPA, which is determined by comparing MRIB-provided metrics. A DF for a given RPA forwards downstream traffic onto its link and forwards upstream traffic from its link toward the rendezvous point link (RPL). The DF performs this function for all bidirectional groups that map to the RPA. The DF on a link is also responsible for processing Join messages from downstream devices on the link as well as ensuring that packets are forwarded to local receivers discovered through a local membership mechanism such as MLD.

Bidirectional PIM offers advantages when there are many moderate or low-rate sources. However, the bidirectional shared trees may have worse delay characteristics than do the source trees built in PIM-SM (depending on the topology).

Only static configuration of bidirectional RPs is supported in IPv6.

How to Configure IPv6 Bidirectional PIM

Configuring Bidirectional PIM and Displaying Bidirectional PIM Information

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim** [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-access-list*] [**bidir**]
4. **exit**
5. **show ipv6 pim** [**vrf** *vrf-name*] **df** [*interface-type interface-number*] [*rp-address*]
6. **show ipv6 pim** [**vrf** *vrf-name*] **df winner**[*interface-type interface-number*] [*rp-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim** [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-access-list*] [**bidir**] **Example:** `Device(config)# ipv6 pim rp-address 2001:DB8::01:800:200E:8C6C bidir`	Configures the address of a PIM RP for a particular group range. Use of the **bidir** keyword means that the group range will be used for bidirectional shared-tree forwarding.
Step 4	**exit** **Example:** `Device(config-if)# exit`	Exits global configuration mode, and returns the device to privileged EXEC mode.
Step 5	**show ipv6 pim** [**vrf** *vrf-name*] **df** [*interface-type interface-number*] [*rp-address*] **Example:** `Device# show ipv6 pim df`	Displays the designated forwarder (DF)-election state of each interface for RP.

	Command or Action	Purpose
Step 6	**show ipv6 pim** [**vrf** *vrf-name*] **df winner**[*interface-type interface-number*] [*rp-address*] **Example:** `Device# show ipv6 pim df winner ethernet 1/0 200::1`	Displays the DF-election winner on each interface for each RP.

Configuration Examples for IPv6 Bidirectional PIM

Example: Configuring Bidirectional PIM and Displaying Bidirectional PIM Information

The following example displays the DF-election states:

```
Device# show ipv6 pim df

Interface          DF State     Timer         Metrics
Ethernet0/0        Winner       4s 8ms        [120/2]
    RP :200::1
Ethernet1/0        Lose         0s 0ms        [inf/inf]
    RP :200::1
```

The following example displays information on the RP:

```
Device# show ipv6 pim df

Interface          DF State     Timer         Metrics
Ethernet0/0        None:RP LAN 0s 0ms        [inf/inf]
    RP :200::1
Ethernet1/0        Winner       7s 600ms      [0/0]
    RP :200::1
Ethernet2/0        Winner       9s 8ms        [0/0]
    RP :200::1
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference

Related Topic	Document Title
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Bidirectional PIM

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 22

IPv6 PIM Passive Mode

This feature allows PIM passive mode to be enabled on an interface so that a PIM passive interface cannot send and receive PIM control messages, but it can act as a reverse path forwarding (RPF) interface for multicast route entries, and it can accept and forward multicast data packets.

Information About IPv6 PIM Passive Mode

A device configured with PIM will always send out PIM hello messages to all interfaces enabled for IPv6 multicast routing, even if the device is configured not to accept PIM messages from any neighbor on the LAN. The IPv6 PIM passive mode feature allows PIM passive mode to be enabled on an interface so that a PIM passive interface cannot send and receive PIM control messages, but it can act as RPF interface for multicast route entries, and it can accept and forward multicast data packets.

How to Configure IPv6 PIM Passive Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast pim-passive-enable**
4. **interface** *type number*
5. **ipv6 pim passive**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ipv6 multicast pim-passive-enable**

Example:

```
Device(config)# ipv6 multicast pim-passive-enable
```

Enables the PIM passive feature on an IPv6 device.

Step 4 **interface** *type number*

Example:

```
Device(config)# interface GigabitEthernet 1/0/0
```

Specifies an interface type and number, and places the device in interface configuration mode.

Step 5 **ipv6 pim passive**

Example:

```
Device(config-if)# ipv6 pim passive
```

Enables the PIM passive feature on a specific interface.

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 PIM Passive

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 23

IPv6 Multicast: Routable Address Hello Option

The routable address hello option adds a PIM hello message option that includes all the addresses on the interface on which the PIM hello message is advertised.

Information About the Routable Address Hello Option

When an IPv6 interior gateway protocol is used to build the unicast routing table, the procedure to detect the upstream device address assumes the address of a PIM neighbor is always same as the address of the next-hop device, as long as they refer to the same device. However, it may not be the case when a device has multiple addresses on a link.

Two typical situations can lead to this situation for IPv6. The first situation can occur when the unicast routing table is not built by an IPv6 interior gateway protocol such as multicast BGP. The second situation occurs when the address of an RP shares a subnet prefix with downstream devices (note that the RP address has to be domain-wide and therefore cannot be a link-local address).

The routable address hello option allows the PIM protocol to avoid such situations by adding a PIM hello message option that includes all the addresses on the interface on which the PIM hello message is advertised. When a PIM device finds an upstream device for some address, the result of RPF calculation is compared with the addresses in this option, in addition to the PIM neighbor's address itself. Because this option includes all the possible addresses of a PIM device on that link, it always includes the RPF calculation result if it refers to the PIM device supporting this option.

Because of size restrictions on PIM messages and the requirement that a routable address hello option fits within a single PIM hello message, a limit of 16 addresses can be configured on the interface.

How to Configure IPv6 Multicast: Routable Address Hello Option

Configuring the Routable Address Hello Option

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 pim hello-interval** *seconds*

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Device(config)# interface FastEthernet 1/0
```

Specifies an interface type and number, and places the device in interface configuration mode.

Step 4 **ipv6 pim hello-interval** *seconds*

Example:

```
Device(config-if)# ipv6 pim hello-interval 45
```

Configures the frequency of PIM hello messages on an interface.

Configuration Example for the Routable Address Hello Option

The following example shows output from the **show ipv6 pim neighbor** command using the **detail** keyword to identify the additional addresses of the neighbors learned through the routable address hello option:

```
Device# show ipv6 pim neighbor detail

Neighbor Address(es)        Interface     Uptime    Expires DR pri Bidir

FE80::A8BB:CCFF:FE00:401   Ethernet0/0   01:34:16  00:01:16 1      B
60::1:1:3

FE80::A8BB:CCFF:FE00:501   Ethernet0/0   01:34:15  00:01:18 1      B
60::1:1:4
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Multicast: Routable Address Hello Option

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 24

PIMv6 Anycast RP Solution

Information About the PIMv6 Anycast RP Solution

PIMv6 Anycast RP Solution Overview

The anycast RP solution in IPv6 PIM allows an IPv6 network to support anycast services for the PIM-SM RP. It allows anycast RP to be used inside a domain that runs PIM only. Anycast RP can be used in IPv4 as well as IPv6, but it does not depend on the Multicast Source Discovery Protocol (MSDP), which runs only on IPv4. This feature is useful when interdomain connection is not required.

Anycast RP is a mechanism that ISP-based backbones use to get fast convergence when a PIM RP device fails. To allow receivers and sources to rendezvous to the closest RP, the packets from a source need to get to all RPs to find joined receivers.

A unicast IP address is chosen as the RP address. This address is either statically configured or distributed using a dynamic protocol to all PIM devices throughout the domain. A set of devices in the domain is chosen to act as RPs for this RP address; these devices are called the anycast RP set. Each device in the anycast RP set is configured with a loopback interface using the RP address. Each device in the anycast RP set also needs a separate physical IP address to be used for communication between the RPs. Each device in the Anycast set must contain the list of all the devices in the Anycast set.

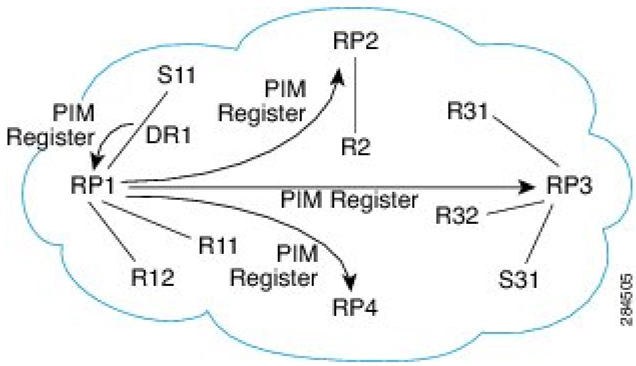
The RP address, or a prefix that covers the RP address, is injected into the unicast routing system inside of the domain. Each device in the anycast RP set is configured with the addresses of all other devices in the anycast RP set, and this configuration must be consistent in all RPs in the set. The IP address of the local device must be included in the set so that all devices in anycast set have the same IP addresses.

PIMv6 Anycast RP Normal Operation

The following illustration shows PIMv6 anycast RP normal operation and assumes the following:

- RP1, RP2, RP3, and RP4 are members in the same anycast RP group.

- S11 and S31 are sources that use RP1 and RP3, respectively, based on their unicast routing metric.
- R11, R12, R2, R31, and R32 are receivers. Based on their unicast routing metrics, R11 and R12 join to RP1, R2 joins to RP2 and R31, and R32 joins to RP3, respectively.

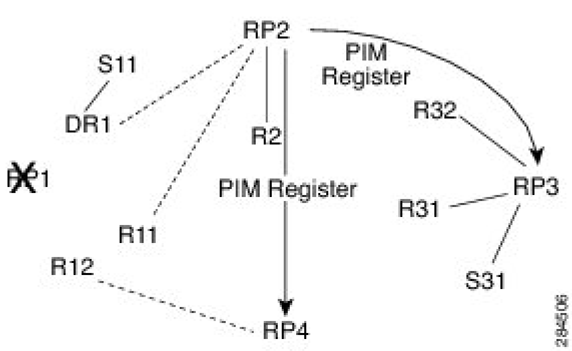


The following sequence of events occurs when S11 starts sending packets:

1. DR1 creates (S,G) states and sends a register to RP1. DR1 may also encapsulate the data packet in the register.
2. Upon receiving the register, RP1 performs normal PIM-SM RP functionality, and forwards the packets to R11 and R12.
3. RP1 also sends the register (which may encapsulate the data packets) to RP2, RP3, and RP4.
4. RP2, RP3, and RP4 do not further forward the register to each other.
5. RP2, RP3, and RP4 perform normal PIM-SM RP functionality, and if there is a data packet encapsulated, RP2 forwards the data packet to R2 and RP3 forwards the data packet to R31 and R32, respectively.
6. The previous five steps repeat for null registers sent by DR1.

PIMv6 Anycast RP Failover

The following illustration shows PIM anycast RP failover.



In failover, when RP1 is not reachable, the following occurs:

- Registers from DR1 will be routed transparently to RP2.
- R11 uses RP2 as the RP, and R12 uses RP4 as the RP.

- Registers from DR1 will be routed from RP2 to RP3 and RP4.

In this way, the loss of the RP (RP1 in this case) is transparent to DR1, R11, and R12, and the network can converge as soon as the IGP is converged.

How to Configure the PIMv6 Anycast RP Solution

Configuring PIMv6 Anycast RP

This task describes how to configure two PIMv6 anycast RP peers. Steps 3 through 11 show the configuration for RP1, and Steps 12 through 19 show the configuration for RP2.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits /prefix-length*}
4. **interface** *type number*
5. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits /prefix-length*}
6. **no shut**
7. **interface** *type number*
8. **ipv6 pim** [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-address-list*] [**bidir**]
9. **no shut**
10. **exit**
11. **ipv6 pim anycast-RP** *rp-address peer-address*
12. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits /prefix-length*}
13. **interface** *type number*
14. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits /prefix-length*}
15. **no shut**
16. **interface** *type number*
17. **ipv6 pim** [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-address-list*] [**bidir**]
18. **no shut**
19. **ipv6 pim anycast-RP** *rp-address peer-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits /prefix-length*} **Example:** `Device(config-if)# ipv6 add 2001:DB8::1:1/128`	Configures an IPv6 address based on an IPv6 general prefix and enable IPv6 processing on an interface.
Step 4	**interface** *type number* **Example:** `Device(config)# interface Loopback4`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 5	**ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits /prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:DB8::4:4/64`	Configures an IPv6 address based on an IPv6 general prefix and enable IPv6 processing on an interface.
Step 6	**no shut** **Example:** `Device(config-if)# no shut`	Enables an interface.
Step 7	**interface** *type number* **Example:** `Device(config-if)# interface Loopback5`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 8	**ipv6 pim** [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-address-list*] [**bidir**] **Example:** `Device(config)# ipv6 pim rp-address 2001:DB8::1:1 acl_sparse1`	Configures the address of a PIM RP for a particular group range.
Step 9	**no shut** **Example:** `Device(config-if)# no shut`	Enables an interface.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Enter this command to exit interface configuration mode and enter global configuration mode.

Step 11	**ipv6 pim anycast-RP** *rp-address peer-address*
	Example:
	The following example shows configuring PIM RP for an anycast group range for a remote and local router:
	`Device(config)# ipv6 pim anycast-rp 2001:DB8::1:1 2001:DB8::3:3`
	`# ipv6 pim anycast-rp 2001:DB8::1:1 2001:DB8::4:4`
		Use this command to configure the address of the PIM RP for an anycast group range.
		• The IP address of the local device must be included in the set so that all devices in anycast set have the same IP addresses.

	Command or Action	Purpose
Step 12	**ipv6 address** {*ipv6-address/prefix-length* | *prefix-name* *sub-bits* /*prefix-length*} **Example:** `Device(config-if)# ipv6 add 2001:DB8::1:1/128`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 13	**interface** *type number* **Example:** `Device(config)# interface Loopback4`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 14	**ipv6 address** {*ipv6-address/prefix-length* | *prefix-name* *sub-bits* /*prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:DB8::3:3/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 15	**no shut** **Example:** `Device(config-if)# no shut`	Enables an interface.
Step 16	**interface** *type number* **Example:** `Device(config-if)# interface Loopback5`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 17	**ipv6 pim** [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-address-list*] [**bidir**] **Example:** `Device(config)# ipv6 pim rp-address 2001:DB8::1:1 acl_sparse1`	Configures the address of a PIM RP for a particular group range.
Step 18	**no shut** **Example:** `Device(config-if)# no shut`	Enables an interface

Step 19	**ipv6 pim anycast-RP** *rp-address* *peer-address*
	Example:
	The following example shows configuring PIM RP for an anycast group range for a remote and local router:
	`Device(config-if)# ipv6 pim anycast-rp 2001:DB8::1:1 2001:DB8::3:3`
	`# ipv6 pim anycast-rp 2001:DB8::1:1 2001:DB8::4:4`
		Use this command to configure the address of the PIM RP for an anycast group range for a remote or local router.

Configuration Examples for the PIMv6 Anycast RP Solution

Example: Configuring PIMv6 Anycast RP

```
RP1
Device1(config)# ipv6 pim rp-address 2001:DB8::1:1 acl_sparse1
Device1(config)# interface Loopback4
Device1(config-if)# ipv6 address 2001:DB8::4:4/64
Device1(config-if)# no shut

Device1(config)# interface Loopback5
Device1(config-if)# ipv6 address 2001:DB8:0:ABCD::1/64
Device1(config-if)# no shut
Device1(config-if)# exit
Device1(config)# ipv6 pim anycast-rp 2001:DB8:0:ABCD::1 2001:DB8::3:3

RP2 (Anycast RP Peer)
Device2(config)# ipv6 pim rp-address 2001:DB8::1:1 acl_sparse1
Device2(config)# interface Loopback4
Device2(config-if)# ipv6 address 2001:DB8::3:3/64
Device2(config-if)# no shut

Device2(config)# interface Loopback5
Device2(config-if)# ipv6 address 2001:DB8:0:ABCD::1/64
Device2(config-if)# no shut
Device2(config)# ipv6 pim anycast-rp 2001:DB8::1:1 2001:DB8::4:4


Device2 show ipv6 pim anycast-rp 2001:DB8::1:1

Anycast RP Peers For 2001:DB8::1:1   Last Register/Register-Stop received
  2001:DB8::3:3 00:00:00/00:00:00
  2001:DB8::4:4 00:00:00/00:00:00
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFC 4610	*Anycast-RP Using Protocol Independent Multicast (PIM)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for PIMv6 Anycast RP Solution

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 25

MTR in VRF

The MTR in VRF feature extends to IPv4 VRF contexts the Cisco IOS software's capability that allows users to configure one or more non-congruent multicast topologies in global IPv4 routing context. These contexts can be used to forward unicast and multicast traffic over different links in the network, or in the case of non-base topologies to provide a Live-Live multicast service using multiple non-congruent multicast topologies mapped to different (S,G) groups.

Information About MTR in VRF

MTR in VRF Overview

The MTR in VRF feature extends to IPv4 VRF contexts, Cisco IOS software's capability that allows users to configure one or more non-congruent multicast topologies in global IPv4 routing context. These contexts can be used to forward unicast and multicast traffic over different links in the network, or in the case of non-base topologies to provide a Live-Live multicast service using multiple non-congruent multicast topologies mapped to different (S,G) groups.

The Cisco IOS Software allows a set of attributes, primarily used by BGP/MPLS L3VPNs, to be configured on a per-address family basis within a VRF. The MTR in VRF feature allows these attributes to be independently configured for the multicast sub-address families within a VRF address family.

How to Configure VRF in MTR

Configuring MTR in VRF

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **ipv4 multicast multitoplogy**
6. **address-family ipv4**
7. **exit-address-family**
8. **address-family ipv4 multicast**
9. **topology** *topology-instance-name*
10. **all-interfaces**
11. **exit**
12. **exit-address-family**
13. **exit**
14. **interface** *type number*
15. **interface** *type number*
16. **vrf forwarding** *vrf-name*
17. **ip address** *ip-address mask*
18. **ip pim sparse-dense-modeip**
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vd1`	Configures a VRF routing table and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 10:1`	Creates routing and forwarding tables for a VRF.
Step 5	**ipv4 multicast multitoplogy** **Example:** `Device(config-vrf)# ipv4 multicast multitoplogy`	Enables IPv4 multicast support for multi-topology routing (MTR) in a VRF instance.
Step 6	**address-family ipv4** **Example:** `Device(config-vrf)# address-family ipv4`	Specifies the IPv4 address family type and enters address family configuration mode.

	Command or Action	Purpose
Step 7	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits address family configuration mode and removes the IPv4 address family.
Step 8	**address-family ipv4 multicast** **Example:** `Device(config-vrf)# address-family ipv4 multicast`	Specifies the IPv4 address family multicast type and enters VRF address family configuration mode.
Step 9	**topology** *topology-instance-name* **Example:** `Device(config-vrf-af)# topology red`	Specifies a topology instance and a name to it and enters VRF address family topology configuration mode.
Step 10	**all-interfaces** **Example:** `Device(config-vrf-af-topology)# all-interfaces`	Configure the topology instance to use all interfaces on the device.
Step 11	**exit** **Example:** `Device(config-vrf-af-topology)# exit`	Exits VRF address-family topology configuration mode and enters VRF address-family configuration mode.
Step 12	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits address family configuration mode and removes the IPv4 address family.
Step 13	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 14	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/1`	Selects the Ethernet interface and enters the interface configuration mode.
Step 15	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/1`	Selects the Ethernet interface and enters the interface configuration mode.
Step 16	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwwarding vrf1`	Associates a VRF instance with the interface.
Step 17	**ip address** *ip-address mask* **Example:**	Sets a primary or secondary IP address for an interface.

	Command or Action	Purpose
	`Device(config-if)# ip address 10.1.10.1 255.255.255.0`	
Step 18	**ip pim sparse-dense-modeip** **Example:** `Device(config-if)# ip pim sparse-dense-mode`	Enables Protocol Independent Multicast (PIM) on an interface.
Step 19	**end** **Example:** `Device(config-if)# end`	Exits the interface configuration mode and enters privileged EXEC mode.

Configuring Examples for MTR in VRF

Example for MTR in VRF

```
Device> enable
Device# configuration terminal
Device(config)# vrf definition vd1
Device(config-vrf)# rd 10:1
Device(config-vrf)# ipv4 multicast multitoplogy
Device(config-vrf)# address-family ipv4
Device(config-vrf)# exit-address-family
Device(config-vrf)# address-family ipv4 multicast
Device(config-vrf-af)# topology red
Device(config-vrf-af-topology)# all-interfaces
Device(config-vrf-af-topology)# exit
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
Device(config)# vrf forwarding vrf1
Device(config)# ip address 10.1.10.1 255.255.255.0
Device(config)# ip pim sparse-dense-mode
Device(config)# end
```

Additional References for MTR in VRF

Related Documents

Related Topic	Document Title
Multitopology Routing (MTR) commands	Cisco IOS Multitopology Routing Command Reference
IP multicast commands	Cisco IOS Multicast Command Reference

Related Topic	Document Title
IP multicast concepts and tasks	*IP Multicast Configuration Guide Library*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for MTR in VRF

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 26

Configuring IP Multicast Over Unidirectional Links

Internet routing protocol assumes that links in a network are bidirectional, and neighborships are formed on bidirectional links that are directly connected. However, in certain scenarios such as satellite links, the links in the network are unidirectional. For routing protocols to work over unidirectional links, you must perform certain configurations. See the following feature document that explains how to configure IP Multicast over unidirectional link (UDL) on the IOS XE platform.

Information About IP Multicast over UDL

Unicast and multicast routing protocols assume that links in a network are bidirectional, and forward data on interfaces from which they have received the routing control information. However, some network links are unidirectional, and usually, in a unidirectional network, the physical send-only interface is on the upstream router (for example, a satellite). The physical receive-only interface is on the downstream router (for example, a ship). In this case, a method of communication that allows the routing protocol to operate in a unidirectional environment is necessary.

Configuring Multicast over UDL helps you achieve control information in a unidirectional environment. You must configure a Unidirectional Link Routing (UDLR) tunnel as a unidirectional generic routing encapsulation (GRE) tunnel and map this tunnel to a one-way satellite link. By doing so, the UDLR tunnel mechanism enables the associated unicast and multicast routing protocols to treat the UDL as a bidirectional link. Before getting into the configuration, you must understand what is UDLR.

What is UDLR?

In unicast routing, when a router receives an update message on an interface for a prefix, it forwards the data to the destinations that match that prefix. Similarly, in multicast routing, when a router receives a join message for a multicast group on an interface, it forwards copies of the data that is destined for that group out from that same interface. Based on these principles, unicast and multicast routing protocols are not supported over UDLs without the use of UDLR. UDLR enables the operation of routing protocols over UDLs without changing the routing protocols themselves. UDLR also enables a router to emulate the behaviour of a bidirectional link for IP operations over the UDLs through a tunnel.

UDLR Tunnel

The UDLR tunnel is the back channel of a unidirectional high-capacity link which transparently emulates a single, bidirectional link for unicast and multicast traffic. A UDLR tunnel enables IP and its associated unicast and multicast routing protocols to treat the UDL as being logically bidirectional. A packet that is destined on a receive-only interface is picked up by the UDLR tunnel mechanism and sent to an upstream router using a generic routing encapsulation (GRE) tunnel. The control traffic thus flows in the opposite direction of the user data flow. When the upstream router receives this packet, the UDLR tunnel makes it appear that the packet was received on a send-only interface on the UDL.

The purpose of the GRE tunnel is to move control packets from a downstream node to an upstream node. This one-way tunnel is mapped to a one-way interface that goes in the opposite direction. The mapping is performed at the link layer so that the one-way interface appears bidirectional. When the upstream node receives packets over the tunnel, it causes the upper-layer protocols to act as if the packets were received on the send-capable UDL.

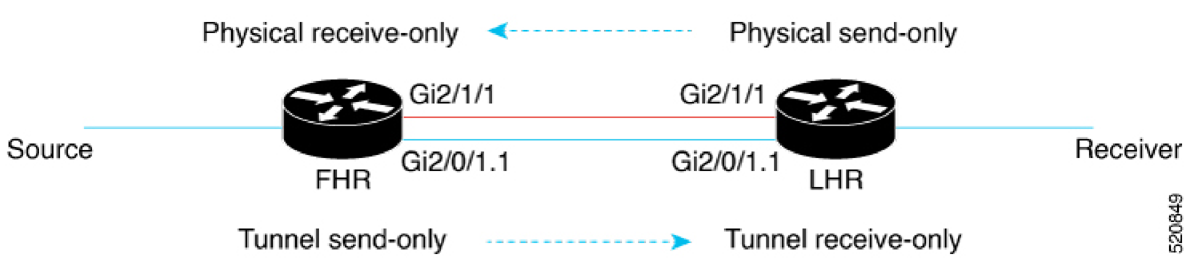
A UDLR tunnel supports the following functionalities:

- Address Resolution Protocol (ARP) and Next Hop Resolution Protocol (NHRP) over a UDL
- Emulation of bidirectional links for all the IP traffic
- Support for IP GRE multipoint at a receive-only tunnel

Note A UDL router can have many routing peers (for example, routers interconnected via a broadcast satellite link). As with bidirectional links, ensure that the number of peer routers is relatively small, to limit the volume of routing updates that are processed.

Multicast Over UDL For Cisco IOS XE

The Multicast over UDL functionality is supported for PIM starting from the IOS XE Amsterdam 17.3.1 release. See the following sample topology that specifies the events that happen sequentially:



- The PIM router receives a PIM join from the downstream PIM neighbor.
- The PIM router connects to the FHR router through the UDLR tunnel.
- The PIM router forwards the PIM join towards the FHR router through the physical interface.
- After the PIM join is received, the FHR router forwards the corresponding multicast traffic to the PIM router through the UDLR tunnel interface. The traffic is encapsulated so that the traffic is not blocked.

In an encap-helper (encapsulated) tunnel on a physical interface, all the (*/S, G) MROUTE entries having the physical interface as the outgoing interface (OIF), use the tunnel interface as the OIF. All the multicast traffic destined to the physical interface flows through the configured tunnel with the encapsulation.

A new flag is propagated and is first introduced to MROUTE. This flag is set against an entry, provided the entry has at least one physical interface as the OIF on which the encap-helper is configured. Consequently, a new flag is introduced to MRIB, which is set against the entry that corresponds to the MROUTE entry. Similarly, a new flag is introduced to MFIB that corresponds to the MRIB entry. The flag is then visible to the platform which punts the multicast traffic only for those (*/S, G) entries based on the status of the flag. The IOS process then switches the packets and sends out these GRE-encapsulated packets.

Note When you remove or disable the encap-helper tunnel configuration on a physical interface, all the (*/S, G) MROUTE entries having the tunnel interface as the OIF use the physical interface as the OIF. The encap-helper feature is removed on the interface and the default behavior is enabled.

Prerquisites for Multicast Over UDL

- Configure IP multicast in your network. For more information, see the Configuring Basic IP Multicast section.
- Ensure that the physical interface and the tunnel interface are on the same mvrf.

Restrictions for Multicast Over UDL

- This functionality is supported only for IPv4 addresses and not IPv6 addresses.
- Bidirectional PIM is not supported.
- The **encap-helper** command is visible only on the physical interfaces, in the supported platforms.
- If the tunnel interface is not up, or if the interface is on a different mvrf compared to the physical interface, the encap-helper configuration does not work.
- This functionality is supported only in the PIM sparse mode.

How to Configure Multicast Over UDL

To configure the IP multicast over UDL functionality, execute the `Router(config-if)# [no] ip pim sparse-mode encap-helper tunnel <tunnel-number>` command.

Note You can configure the UDLR tunnel only in the pim sparse-mode.

```
Router (config)# interface Tunnel1001
vrf forwarding VRF1
no ip address
```

```
ip pim sparse-mode
tunnel source GigabitEthernet2/0/1.1
tunnel destination 3.3.5.3
tunnel key 7354
tunnel udlr send-only GigabitEthernet2/1/1
tunnel udlr address-resolution
tunnel vrf VRF1
end
LMA1#sh runn int  GigabitEthernet2/1/1
Building configuration...

Current configuration : 177 bytes
!
interface GigabitEthernet2/1/1
vrf forwarding VRF1
ip address 12.12.2.1 255.255.255.0
ip pim sparse-mode encap-helper Tunnel1001
ip ospf 101 area 0
negotiation auto
end
```

When you execute the **udlr send-only** command, it associates the tunnel send-only interface with the receive-only port.

When you execute the **encap-helper** command, the encap-helper tunnel is configured on a physical interface. All the (*/S, G) MROUTE entries with the physical interface as the OIF use the tunnel interface as the OIF. All the multicast traffic destined to the physical interface flows through the configured tunnel interface with the encapsulation.

Note When the tunnel interface goes down, the tunnel interfaces in the OIF list are removed.

Verifying Multicast Over UDL Configuration

Verify whether the configuration is successful by executing the following show commands:

```
Router# show ip mroute vrf VRF1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group,
G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
* - determined by Assert, # - iif-starg configured on rpf intf,
e - encap-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 239.1.1.1), 00:28:39/stopped, RP 4.4.3.4, flags: SPF
  Incoming interface: GigabitEthernet2/1/1, RPF nbr 12.12.2.2
  Outgoing interface list: Null
(100.100.3.2, 239.1.1.1), 00:28:39/00:02:58, flags: FTe, eh_tun_count :1
```

```
Incoming interface: GigabitEthernet2/0/4.2, RPF nbr 0.0.0.0
Outgoing interface list:
Tunnel1001, Forward/Sparse, 00:24:59/00:03:21
(*, 224.0.1.40), 02:10:54/00:02:08, RP 4.4.3.4, flags: SJCL
Incoming interface: GigabitEthernet2/1/1, RPF nbr 12.12.2.2
 Outgoing interface list:
 Loopback1, Forward/Sparse, 02:10:52/00:02:08
```

```
Router(config-if)# show ip mrib vrf VRF1 route
IP Multicast Routing Information Base
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
C - Directly-Connected Check, S - Signal, IA - Inherit Accept, D - Drop
ET - Data Rate Exceeds Threshold,K - Keepalive,DDE - Data Driven Event
ME - MoFRR ECMP Flow based, MNE - MoFRR Non-ECMP Flow based,
MP - Primary MoFRR Non-ECMP Flow based entry,
e  - Encap helper tunnel flag
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
LD - Local Disinterest, MD - mCAC Denied, MI - mLDP Interest
 A2 - MoFRR ECMP Backup Accept
(*,224.0.0.0/4) Flags: C
(*,224.0.1.40) RPF nbr: 12.12.2.2 Flags: C
 GigabitEthernet2/1/1 Flags: A NS
 Loopback1 Flags: F IC NS
(*,239.1.1.1) RPF nbr: 12.12.2.2 Flags: C
 GigabitEthernet2/1/1 Flags: A
(100.100.3.2,239.1.1.1) RPF nbr: 0.0.0.0 Flags: e
 GigabitEthernet2/0/4.2 Flags: A
 Tunnel1001 Flags: F NS
```

```
Router# show ip mfib vrf VRF1
Entry Flags:  C - Directly Connected, S - Signal, IA - Inherit A flag,
             ET - Data Rate Exceeds Threshold, K - Keepalive
             DDE - Data Driven Event, HW - Hardware Installed
             ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
             MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
             MS  - MoFRR  Entry in Sync, MC - MoFRR entry in MoFRR Client,
             e   - Encap helper tunnel flag.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
             NS - Negate Signalling, SP - Signal Present,
             A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
             MA - MFIB Accept, A2 - Accept backup,
             RA2 - MRIB Accept backup, MA2 - MFIB Accept backup

Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   HW Pkt Count/FS Pkt Count/PS Pkt Count   Egress Rate in pps
VRF VRF1
(*,224.0.0.0/4) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   0/0/0/0, Other: 19/19/0
(*,224.0.1.40) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   0/0/0/0, Other: 0/0/0
   GigabitEthernet2/1/1 Flags: A NS
   Loopback1 Flags: F IC NS
     Pkts: 0/0/0    Rate: 0 pps
(*,239.1.1.1) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   0/0/0/0, Other: 0/0/0
   GigabitEthernet2/1/1 Flags: A
(100.100.3.2,239.1.1.1) Flags: HW e
   SW Forwarding: 1623356/997/46/358, Other: 1/0/1
   HW Forwarding:   101680/0/63/0, Other: 33853/1/33852
```

```
   GigabitEthernet2/0/4.2 Flags: A
   Tunnel1001 Flags: F NS
     Pkts: 0/0/1539215    Rate: 0 pps
```

The encap tunnel prefixed with ‘e’ indicates the entry of the new flag for MROUTE, MFIB. The highlighted portions also indicate the successful configuration of the tunnel interface and the SW Forwarding counters.

Verifying Multicast Over UDL Configuration

Verify whether the configuration is successful by executing the following show commands:

```
Router# show ip mroute vrf VRF1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group,
G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
* - determined by Assert, # - iif-starg configured on rpf intf,
e - encap-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 239.1.1.1), 00:28:39/stopped, RP 4.4.3.4, flags: SPF
  Incoming interface: GigabitEthernet2/1/1, RPF nbr 12.12.2.2
  Outgoing interface list: Null
(100.100.3.2, 239.1.1.1), 00:28:39/00:02:58, flags: FTe, eh_tun_count :1
Incoming interface: GigabitEthernet2/0/4.2, RPF nbr 0.0.0.0
Outgoing interface list:
Tunnel1001, Forward/Sparse, 00:24:59/00:03:21
(*, 224.0.1.40), 02:10:54/00:02:08, RP 4.4.3.4, flags: SJCL
Incoming interface: GigabitEthernet2/1/1, RPF nbr 12.12.2.2
 Outgoing interface list:
 Loopback1, Forward/Sparse, 02:10:52/00:02:08
```

```
Router(config-if)# show ip mrib vrf VRF1 route
IP Multicast Routing Information Base
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
C - Directly-Connected Check, S - Signal, IA - Inherit Accept, D - Drop
ET - Data Rate Exceeds Threshold,K - Keepalive,DDE - Data Driven Event
ME - MoFRR ECMP Flow based, MNE - MoFRR Non-ECMP Flow based,
MP - Primary MoFRR Non-ECMP Flow based entry,
e  - Encap helper tunnel flag
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
LD - Local Disinterest, MD - mCAC Denied, MI - mLDP Interest
 A2 - MoFRR ECMP Backup Accept
(*,224.0.0.0/4) Flags: C
(*,224.0.1.40) RPF nbr: 12.12.2.2 Flags: C
 GigabitEthernet2/1/1 Flags: A NS
 Loopback1 Flags: F IC NS
(*,239.1.1.1) RPF nbr: 12.12.2.2 Flags: C
 GigabitEthernet2/1/1 Flags: A
(100.100.3.2,239.1.1.1) RPF nbr: 0.0.0.0 Flags: e
```

```
 GigabitEthernet2/0/4.2 Flags: A
 Tunnel1001 Flags: F NS

Router# show ip mfib vrf VRF1
Entry Flags:  C - Directly Connected, S - Signal, IA - Inherit A flag,
             ET - Data Rate Exceeds Threshold, K - Keepalive
             DDE - Data Driven Event, HW - Hardware Installed
             ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
             MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
             MS  - MoFRR  Entry in Sync, MC - MoFRR entry in MoFRR Client,
             e   - Encap helper tunnel flag.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
             NS - Negate Signalling, SP - Signal Present,
             A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
             MA - MFIB Accept, A2 - Accept backup,
             RA2 - MRIB Accept backup, MA2 - MFIB Accept backup

Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   HW Pkt Count/FS Pkt Count/PS Pkt Count   Egress Rate in pps
VRF VRF1
(*,224.0.0.0/4) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   0/0/0/0, Other: 19/19/0
(*,224.0.1.40) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   0/0/0/0, Other: 0/0/0
   GigabitEthernet2/1/1 Flags: A NS
   Loopback1 Flags: F IC NS
     Pkts: 0/0/0    Rate: 0 pps
(*,239.1.1.1) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   0/0/0/0, Other: 0/0/0
   GigabitEthernet2/1/1 Flags: A
(100.100.3.2,239.1.1.1) Flags: HW e
   SW Forwarding: 1623356/997/46/358, Other: 1/0/1
   HW Forwarding:   101680/0/63/0, Other: 33853/1/33852
   GigabitEthernet2/0/4.2 Flags: A
   Tunnel1001 Flags: F NS
     Pkts: 0/0/1539215    Rate: 0 pps
```

The encap tunnel prefixed with ‘e’ indicates the entry of the new flag for MROUTE, MFIB. The highlighted portions also indicate the successful configuration of the tunnel interface and the SW Forwarding counters.

PART II

Multicast Services

CHAPTER 27

Implementing Multicast Service Reflection

The Cisco Multicast Service Reflection feature provides the capability for users to translate externally received multicast or unicast destination addresses to multicast or unicast addresses that conform to their organization's internal addressing policy. Using this feature, users do not need to redistribute unicast routes from external sources at the translation boundary into their network infrastructure for Reverse Path Forwarding (RPF) to work properly. In addition, users can receive identical feeds from two ingress points in the network and route them independently.

Prerequisites for Implementing Multicast Service Reflection

- Configure your multicast-enabled network with the necessary infrastructure to run either Protocol Independent Multicast Sparse Mode (PIM-SM), Bidirectional PIM (bidir-PIM), or PIM Source Specific Multicast (PIM-SSM). The configuration process may include configuring RPs, interface boundaries, or SSM ranges.

For configuration information, see *Configuring Basic IP Multicast*.

- Confirm that the virtual interface for multicast service reflection (Vif1 interface) is installed in your border router and the Multicast Service Reflection application is installed and operational.
- Each active receiver must initiate an Internet Group Management Protocol (IGMP) join to the multicast group that is defined on the router in the PIM domain.

Restrictions for Implementing Multicast Service Reflection

- When translating groups of multicast packets that are destined for the same multicast group but are originating from different sources, as in the case when using PIM-SSM, all multicast packets destined for a particular SSM group will get mapped to a single (S, G) after translation has occurred. For example, if (10.1.1.1, 232.1.1.1) and (10.1.1.2, 232.1.1.1) need to be translated, they will appear as a single entry, for example, (192.168.1.2, 232.239.1.1), where 192.168.1.2 is an IP address that resides in the Vif IP subnet.
- PIM and IGMP control packets are not translated.
- Only one Vif is allowed in each VRF.
- Only Vif1 is allowed in global configuration. Vif2 and Vif333 are allowed in VRFs.
- For unicast-to-multicast destination translation and splitting configuration:
    - Rules with input interface configured have higher preference over those without input interface.
    - Extra static routes with the same mask length must be configured for each multicast service reflection rule. The mask length of the static route must be the same as the mask length of the rule.
    - The maximum split number for unicast-to-multicast destination splitting is limited to 2.

Information About Implementing Multicast Service Reflection

The following topics provide detailed information about implementing multicast service reflection, including its benefits.

Benefits of Implementing Multicast Service Reflection

- Allows users to translate externally received multicast or unicast destination addresses to multicast or unicast addresses that conform to their company's internal addressing policy. This allows the seperation of the private addressing scheme used by the content provider from the public addressing used by the service provider. The following types of translations are supported:
    - Multicast-to-Multicast Destination Translation
    - Multicast-to-Unicast Destination Translation
    - Unicast-to-Multicast Destination Translation
    - Multicast-to-Multicast Destination Splitting
    - Multicast-to-Unicast Destination Splitting
    - Unicast-to-Multicast Destination Splitting
- Provides logical separation between private and public multicast networks.
- Provides the flexibility to forward multicast packets—translated or untranslated—out the same outgoing interface.

- Provides redundancy by allowing users to get identical feeds from two ingress points in the network and route them independently.
- Allows users to use the subnet of their choice to be the source network and scope it appropriately.

Rendezvous Points

A rendezvous point (RP) is a role that a router performs when operating in PIM-SM or bidirectional PIM. An RP is required only in networks running PIM-SM or bidirectional PIM. In PIM-SM, only network segments with active receivers that have explicitly requested multicast data will be forwarded the traffic.

An RP acts as the meeting place for sources and receivers of multicast data. In a PIM-SM network, first hop designated routers with directly connected sources initially send their traffic to the RP. This traffic is then forwarded to receivers down a shared distribution tree. By default, when the last hop router with a directly connected receiver receives traffic from the shared tree, it immediately performs a shortest path tree switchover and sends a Join message towards the source, creating a source-based distribution tree between the source and the receiver.

PIM Sparse Mode

PIM sparse mode (PIM-SM) uses a pull model to deliver multicast traffic. Only network segments with active receivers that have explicitly requested the data will receive the traffic.

Unlike dense mode interfaces, sparse mode interfaces are added to the multicast routing table only when periodic Join messages are received from downstream routers, or when a directly connected member is on the interface. When forwarding from a LAN, sparse mode operation occurs if an RP is known for the group. If so, the packets are encapsulated and sent toward the RP. When no RP is known, the packet is flooded in a dense mode fashion. If the multicast traffic from a specific source is sufficient, the first hop router of the receiver may send Join messages toward the source to build a source-based distribution tree.

PIM-SM distributes information about active sources by forwarding data packets on the shared tree. Because PIM-SM uses shared trees (at least, initially), it requires the use of a rendezvous point (RP). The RP must be administratively configured in the network.

In sparse mode, a router assumes that other routers do not want to forward multicast packets for a group, unless there is an explicit request for the traffic. When hosts join a multicast group, the directly connected routers send PIM Join messages toward the RP. The RP tracks multicast groups. Hosts that send multicast packets are registered with the RP by the first hop router of that host. The RP then sends Join messages toward the source. At this point, packets are forwarded on a shared distribution tree. If the multicast traffic from a specific source is sufficient, the first hop router of the host may send Join messages toward the source to build a source-based distribution tree.

First-hop designated routers with directly connected sources register with the RP and then data is forwarded down the shared tree to the receivers. The edge routers learn about a particular source when they receive data packets on the shared tree from that source through the RP. The edge router then sends PIM (S, G) Join messages toward that source. Each router along the reverse path compares the unicast routing metric of the RP address to the metric of the source address. If the metric for the source address is better, it will forward a PIM (S, G) Join message toward the source. If the metric for the RP is the same or better, then the PIM (S, G) Join message will be sent in the same direction as the RP. In this case, the shared tree and the source tree would be considered congruent.

If the shared tree is not an optimal path between the source and the receiver, the routers dynamically create a source tree and stop traffic from flowing down the shared tree. This behavior is the default behavior in Cisco

IOS software. Network administrators can force traffic to stay on the shared tree by using the Cisco IOS **ip pim spt-threshold infinity** command.

PIM-SM scales well to a network of any size, including those with WAN links. The explicit join mechanism prevents unwanted traffic from flooding the WAN links.

Vif Interface

The Vif interface is similar to a loopback interface--it is a logical IP interface that is always up when the router is active.

The Vif interface needs to reside on its own unique subnet, and that subnet should be advertised in the Interior Gateway Protocol (IGP) updates (RIP, EIGRP, OSPF, ISIS).

The Vif interface maintains information about the input interface, private-to-public mgroup mappings, mask length, which defines your pool range, and the source of the translated packet.

Note

- Multicast-to-multicast and multicast-to-unicast scenarios can be configured only under the Vif1 interface. Unicast-to-multicast scenarios can be configured under all the Vif interfaces, and are not restricted to the Vif1 interface.
- Vif1 is attached to the default VRF; Vif*n* can be used for user-created VRF.

Multicast Service Reflection Application

Cisco multicast service reflection is an application running in Cisco IOS XE software interrupt level switching that processes packets forwarded by Cisco IOS XE software to the Vif interface. Unlike IP multicast Network Address Translation (NAT), which only translates the source IP address, the IP reflect service translates both source and destination addresses. Multicast service reflection is especially useful when users that have not yet moved to the new multicast group still need to receive the untranslated stream.

Multicast service reflection is implemented using an interface CLI statement. Each configured multicast service reflection CLI statement establishes a packet match and rewrite operation acting on packets sent by Cisco IOS XE unicast or multicast packet routing onto the Vif interface. The matched and rewritten packet is sent back into Cisco IOS XE unicast or multicast packet routing, where it is handled like any other packet arriving from an interface.

The Vif interface is a receiver for the original stream and makes it appear that the new stream is coming from a source directly connected to the Vif subnet. The Vif interface is a Designated Router (DR) for active sources and registers with the appropriate RP.

More than one multicast service reflection operation can be configured to match the same packets, which allows you to replicate the same received traffic to multiple destination addresses. The maximum split number for unicast-to-multicast destination splitting is limited to 2.

How to Implement Multicast Service Reflection

Configuring Multicast Service Reflection

Perform this task to configure multicast service reflection.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **interface** *type number*
5. **ip pim sparse-mode**
6. no shutdown
7. exit
8. Repeat Steps 4 through 7 for each PIM interface.
9. **interface Vif**
10. **ip address** *ip-address mask* [**secondary**]
11. **ip pim sparse-mode**
12. **ip service reflect** *input-interface* **destination** *destination-address* **to** *new-destination-address* **mask-len** *number* **source** *ip-address* OR **ip service reflect** *input-interface* **destination** *destination-address* **to** *new-destination-address* **mask-len** *number* **source** *old-range* **to** *new-range* **src-mask-len** *mask*
13. **ip igmp static-group** {***** | *group-address* [**source** {*source-address* | **ssm-map**}]}
14. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing. • Use the **distributed** keyword to enable the Multicast Distributed Switching feature.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Enters interface configuration mode for the specified interface type and number.
Step 5	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables PIM sparse mode on the interface.
Step 6	no shutdown **Example:** `Router(config-if)# no shutdown`	Enables an interface.
Step 7	exit **Example:** `Router(config-if)# exit`	Exits interface configuration mode, and returns to global configuration mode.
Step 8	Repeat Steps 4 through 7 for each PIM interface.	--
Step 9	**interface Vif** **Example:** `Router(config)# interface Vif1`	Enters interface configuration mode for the Vif interface.
Step 10	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 11	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables PIM sparse mode on an interface.
Step 12	**ip service reflect** *input-interface* **destination** *destination-address* **to** *new-destination-address* **mask-len** *number* **source** *ip-address* OR **ip service reflect** *input-interface* **destination** *destination-address* **to** *new-destination-address* **mask-len** *number* **source** *old-range* **to** *new-range* **src-mask-len** *mask* **Example:** `Router(config-if)# ip service reflect ethernet0 destination 66.0.0.7 to 239.3.3.0 mask-len 32 source 10.1.1.2`	Matches and rewrites multicast packets routed to the Vif interface. • The matched and rewritten packets are sent back into Cisco multicast packet routing (or unicast routing if the destination is unicast), where they are handled like any other packets arriving from an interface. The **ip service reflect destination** *A.B.C.D.* **to** *E.F.G.H.* **mask-len source** command can be used to configure

	Command or Action	Purpose
	`Router(config-if)# ip service reflect GigabitEthernet5 destination 66.0.0.7 to 239.3.3.0 mask-len 32 source 10.1.1.0 to 22.1.1.0 src-mask-len 24`	multicast-to-multicast, multicast-to-unicast, and unicast-to-multicast scenarios. • If A.B.C.D is a unicast address and E.F.G.H is a multicast address, a unicast-to-multicast scenario is configured. • If A.B.C.D is a multicast address and E.F.G.H is a multicast address, a multicast-to-multicast scenario is configured. • If A.B.C.D is a multicast address and E.F.G.H is a unicast address, a multicast-to-unicast scenario is configured.
Step 13	**ip igmp static-group** {***** | *group-address* [**source** {*source-address* | **ssm-map**}]} **Example:** `Router(config-if)# ip igmp static-group 224.1.1.1`	Configures the router to be a statically connected member of the specified group on the interface, and forwards traffic destined for the multicast group onto the interface. • This step is only applicable for multicast-to-multicast and multicast-to-unicast scenarios; not applicable for unicast-to-multicast scenarios.
Step 14	end **Example:** `Router(config-if)# end`	Exits interface configuration mode, and returns to privileged EXEC mode.

Configuration Examples for Multicast Service Reflection

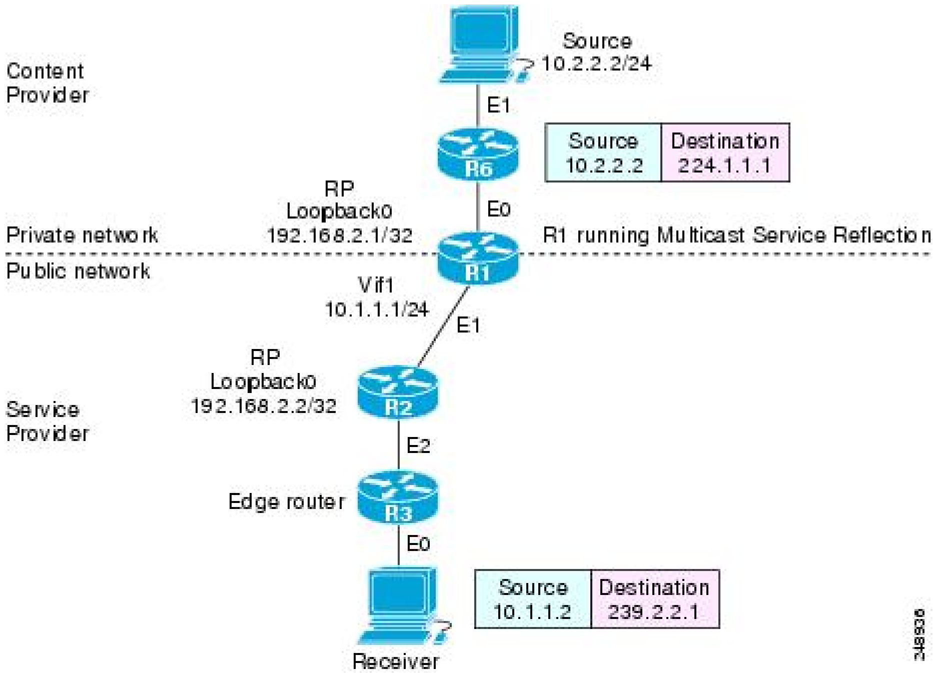
The following examples show the configurations for multicast service reflection.

Example: Multicast-to-Multicast Destination Translation

The following example shows how to implement multicast service reflection (multicast-to-multicast destination translation) in a service provider network. Multicast-to-Multicast Destination Translation allows service providers to translate externally received content provider multicast destination addresses to multicast destination addresses that conform to the service provider’s internal addressing policy.

This example uses the topology illustrated in the following figure.

Figure 24: Multicast Service Reflection (Multicast-to-Multicast Destination Translation) in a Service Provider Network: Example Topology



In this example topology, a content provider is sending financial market information to a service provider, which in turn is sending that information to active receivers (brokerage houses). The service provider may be receiving market data from multiple content providers.

Router R1 is an edge router in the service provider's PIM domain.

Router R1 has a loopback interface and is acting as the RP for the 224.1.1.0/24 address range.

Router R1 has a Vif1 interface and is running the multicast service reflection application.

Router R2 has a loopback interface and is acting as the RP for the 239.2.2.0/24 address range.

Enter these commands on the router running the multicast service reflection application (R1):

```
configure terminal
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
! Configure the loopback interface for the Service Provider RP
!
interface loopback 0
ip address 192.168.2.1 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.2.1 mcast-content-provider-groups override
ip pim rp-address 192.168.2.2 mcast-service-provider-groups override
ip access-list standard mcast-content-provider-groups
permit 224.1.1.0 0.0.0.255
ip access-list standard mcast-service-provider-groups
permit 239.2.2.0 0.0.0.255
!
! Configure the Vif1 virtual interface for multicast service reflection
!
```

```
interface Vif1
ip address 10.1.1.1 255.255.255.0
ip pim sparse-mode
ip service reflect Ethernet 0 destination 224.1.1.0 to 239.2.2.0 mask-len 24 source 10.1.1.2
ip igmp static-group 224.1.1.0
ip igmp static-group 224.1.1.1
ip igmp static-group 224.1.1.2
ip igmp static-group 224.1.1.3
.
.
.
ip igmp static-group 224.1.1.255
```

Enter these commands on the router that is the RP in the service provider network (R2):

```
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
interface loopback 0
ip address 192.168.2.2 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.2.2 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups
permit 239.2.2.0 0.0.0.255
!
```

Enter these commands on all the other routers in the service provider network (R3):

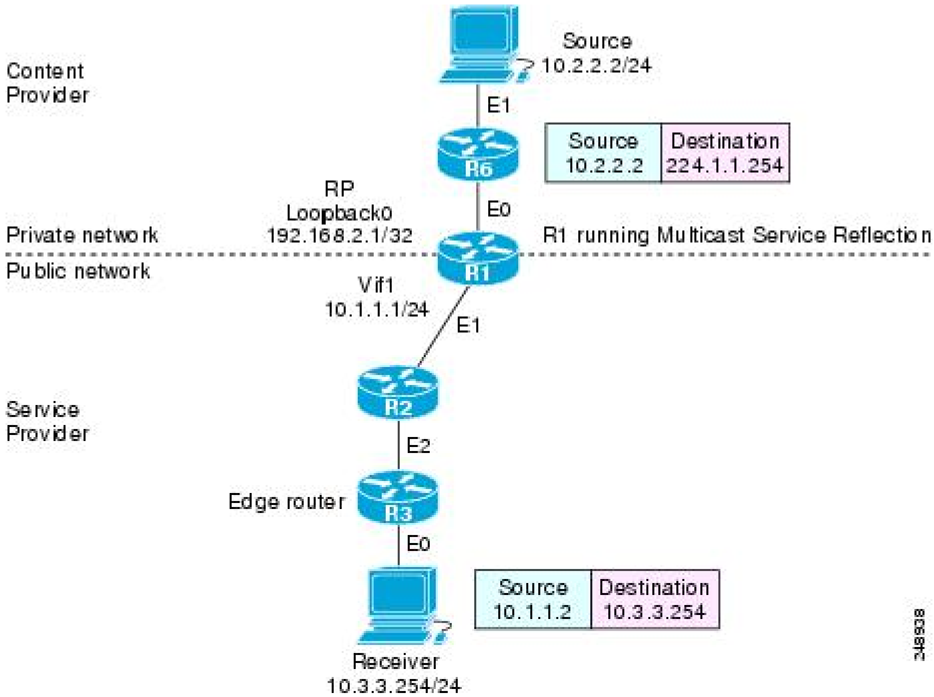
```
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
ip pim rp-address 192.168.2.2 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups
permit 239.2.2.0 0.0.0.255
end
```

Example: Multicast-to-Unicast Destination Translation

The following example shows how to implement multicast service reflection (multicast-to-unicast destination translation) in a service provider network. Multicast-to-Unicast Destination Translation allows service providers to translate externally received content provider multicast destination addresses to unicast destination addresses that conform to the service provider's internal addressing policy.

This example uses the topology illustrated in the following figure.

Figure 25: Multicast Service Reflection (Multicast-to-Unicast Destination Translation) in a Service Provider Network: Example Topology



In this example topology, a content provider is sending financial market information to a service provider, which in turn is sending that information to active receivers (brokerage houses). The service provider may be receiving market data from multiple content providers.

Router R1 is an edge router in the service provider's PIM domain.

Router R1 has a loopback interface and is acting as the RP for the 224.1.1.0/24 address range.

Router R1 has a Vif1 interface and is running the multicast service reflection application.

Routers R2 and R3 are non PIM enabled routers running unicast routing only in the service provider network.

Enter these commands on the router running the multicast service reflection application (R1):

```
configure terminal
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
! Configure the loopback interface for the Service Provider RP
!
interface loopback 0
ip address 192.168.2.1 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.2.1 mcast-content-provider-groups override
ip access-list standard mcast-content-provider-groups
permit 224.1.1.10 0.0.0.255
!
! Configure the Vif1 virtual interface for multicast service reflection
!
interface Vif1
ip address 10.1.1.1 255.255.255.0
```

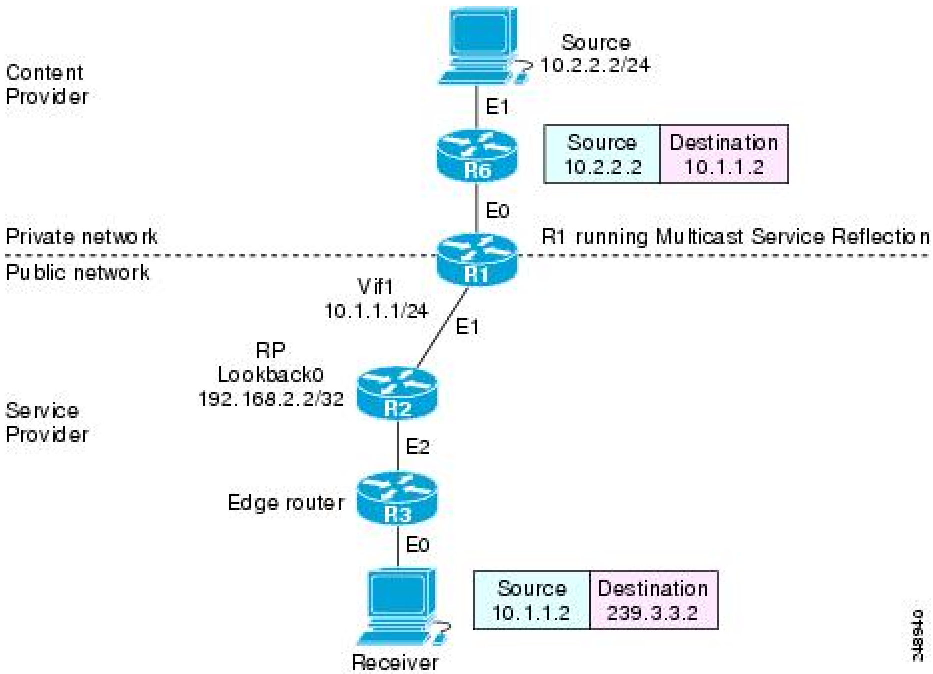
```
ip pim sparse-mode
ip service reflect Ethernet 0 destination 224.1.1.0 to 10.3.3.0 mask-len 24 source 10.1.1.2
end
```

Example: Unicast-to-Multicast Destination Translation

The following example shows how to implement multicast service reflection (unicast-to-multicast destination translation) in a service provider network. Unicast-to-Multicast Destination Translation allows service providers to translate externally received content provider unicast destination addresses to multicast destination addresses that conform to the service provider's internal addressing policy.

This example uses the topology illustrated in the following figure.

Figure 26: Multicast Service Reflection (Unicast-to-Multicast Destination Translation) in a Service Provider Network: Example Topology



In this example topology, a content provider is sending financial market information to a service provider, which in turn is sending that information to active receivers (brokerage houses). The service provider may be receiving market data from multiple content providers.

Router R1 is an edge router in the service provider's PIM domain.

Router R1 has a Vif1 interface and is running the multicast service reflection application.

Router R2 has a loopback interface and is acting as the RP for the 239.3.3.0/24 address range.

Router R3 is another edge router in the service provider's PIM domain.

Enter these commands on the router running the multicast service reflection application (R1):

```
configure terminal
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
```

```
ip pim rp-address 192.168.2.2 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups
permit 239.3.3.0 0.0.0.255
!
! Configure the Vif1 virtual interface for multicast service reflection
!
interface Vif1
ip address 10.1.1.1 255.255.255.0
ip pim sparse-mode
ip service reflect Ethernet 0 destination 10.1.1.2 to 239.3.3.2 mask-len 32 source 10.1.1.2
ip route 10.1.1.2 255.255.255.255 vif1
```

Enter these commands on the router that is the RP in the service provider network (R2):

```
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
interface loopback 0
ip address 192.168.2.2 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.2.2 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups
permit 239.3.3.0 0.0.0.255
```

Enter these commands on all the other routers in the service provider network (R3):

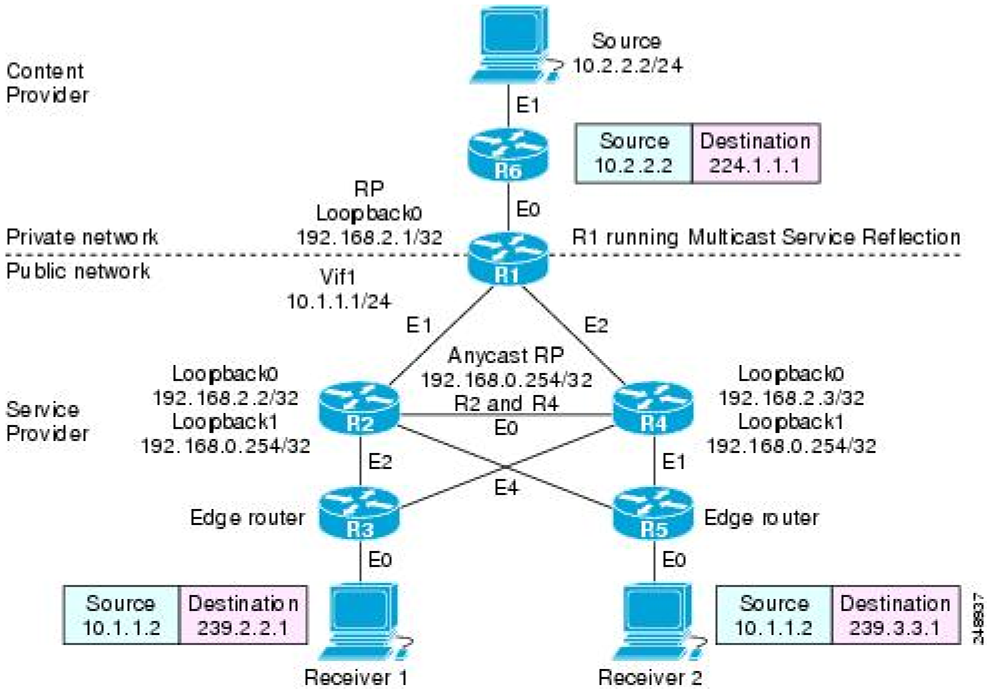
```
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
ip pim rp-address 192.168.2.2 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups
permit 239.3.3.0 0.0.0.255
end
```

Example: Multicast-to-Multicast Destination Splitting

The following example shows how to implement multicast service reflection (multicast-to-multicast destination splitting, where the multicast single stream is converted into two unique multicast streams) in a service provider network. Multicast-to-Multicast Destination Splitting allows service providers to translate externally received content provider multicast destination addresses to multiple multicast destination addresses that conform to the service provider's internal addressing policy.

This example uses the topology illustrated in the following figure.

Figure 27: Multicast Service Reflection (Multicast-to-Multicast Destination Splitting) in a Service Provider Network: Example Topology



In this example topology, a content provider is sending financial market information to a service provider, which in turn is sending that information to active receivers (brokerage houses). The service provider may be receiving market data from multiple content providers.

Router R1 is an edge router in the service provider's PIM domain.

Router R1 has a loopback configured and is acting as an RP for the 224.1.1.0/24 address range.

Router R1 has a Vif1 interface and is running the multicast service reflection application.

Routers R2 and R4 have multiple loopback interfaces and are acting as anycast RPs for the 239.2.2.0 and 239.3.3.0 address ranges.

Router R3 and R5 are edge routers in the service provider's PIM domain.

Enter these commands on the router running the multicast service reflection application (R1):

```
configure terminal
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
! Configure the loopback interface for the Service Provider RP
!
interface loopback 0
ip address 192.168.2.1 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.2.1 mcast-content-provider-groups override
ip pim rp-address 192.168.0.254 mcast-service-provider-groups override
ip access-list standard mcast-content-provider-groups
permit 224.1.1.0 0.0.0.255
ip access-list standard mcast-service-provider-groups
permit 239.2.2.0 0.0.0.255
```

```
ip access-list standard mcast-service-provider-groups
permit 239.3.3.0 0.0.0.255
!
! Configure the Vif1 virtual interface for multicast service reflection
!
interface Vif1
ip address 10.1.1.1 255.255.255.0
ip pim sparse-mode
ip service reflect Ethernet 0 destination 224.1.1.0 to 239.2.2.0 mask-len 24 source 10.1.1.2
ip service reflect Ethernet 0 destination 224.1.1.0 to 239.3.3.0 mask-len 24 source 10.1.1.2
 ip igmp static-group 224.1.1.0
ip igmp static-group 224.1.1.1 ip igmp static-group 224.1.1.2 ip igmp static-group 224.1.1.3
.
.
.
ip igmp static-group 224.1.1.254
```

Enter these commands on the R2 router that is an anycast RP in the service provider network:

```
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
interface loopback 0
ip address 192.168.2.2 255.255.255.255
ip pim sparse-mode
!
interface loopback 1
description --- Anycast RP ---
ip address 192.168.0.254 255.255.255.255
ip pim sparse-mode
!
ip msdp peer 192.168.2.3 connect-source Loopback0 ip msdp originator-id Loopback0
!
ip pim rp-address 192.168.0.254 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups
permit 239.2.2.0 0.0.0.255
ip access-list standard mcast-service-provider-groups
permit 239.3.3.0 0.0.0.255
```

Enter these commands on the R4 router that is an anycast RP in the service provider network:

```
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
interface loopback 0
ip address 192.168.2.3 255.255.255.255
ip pim sparse-mode interface loopback 1
ip address 192.168.0.254 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.0.254 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups
permit 239.2.2.0 0.0.0.255
ip access-list standard mcast-service-provider-groups
permit 239.3.3.0 0.0.0.255
!
ip msdp peer 192.168.2.2 connect-source Loopback0 ip msdp originator-id Loopback0
```

Enter these commands on the R3 and R5 routers in the service provider network:

```
ip multicast-routing distributed
ip pim rp-address 192.168.0.254 mcast-service-provider-groups override
```

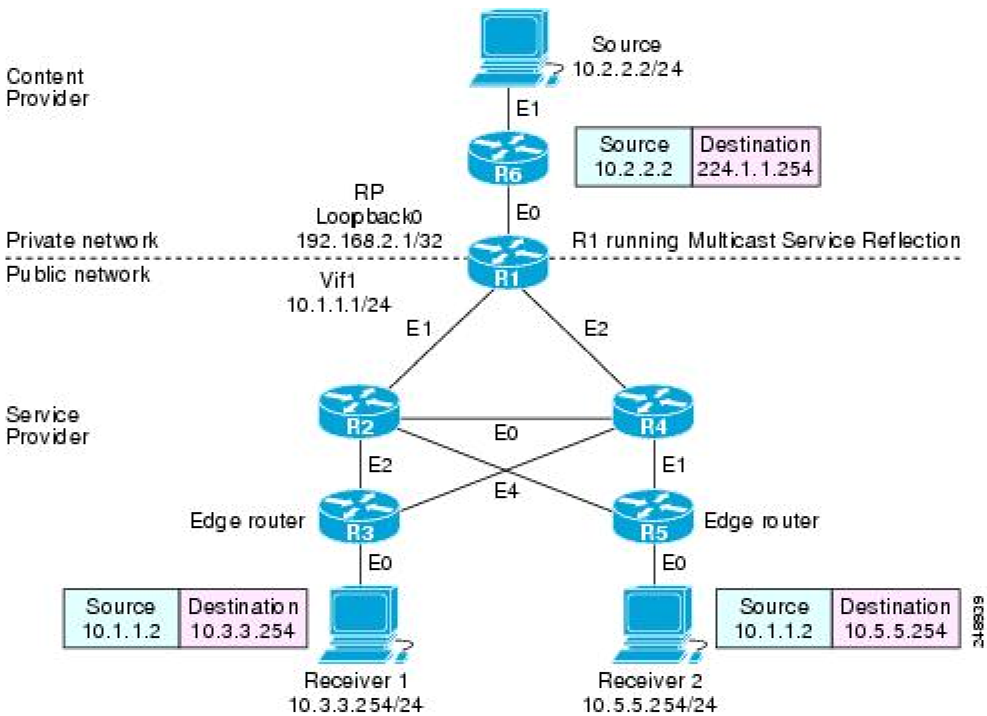
```
ip access-list standard mcast-service-provider-groups
permit 239.2.2.0 0.0.0.255
permit 239.3.3.0 0.0.0.255
!
```

Example: Multicast-to-Unicast Destination Splitting

The following example shows how to implement multicast service reflection (multicast-to-unicast destination splitting, where the multicast single stream is converted into two unique unicast streams) in a service provider network. Multicast-to-Unicast Destination Splitting allows service providers to translate externally received content provider multicast destination addresses to multiple unicast destination addresses that conform to the service provider's internal addressing policy.

This example uses the topology illustrated in the following figure.

Figure 28: Multicast Service Reflection (Multicast-to-Unicast Destination Splitting) in a Service Provider Network: Example Topology



In this example topology, a content provider is sending financial market information to a service provider, which in turn is sending that information to active receivers (brokerage houses). The service provider may be receiving market data from multiple content providers.

Router R1 is an edge router in the service provider's PIM domain.

Router R1 is acting as a RP for the 224.1.1.0/24 address range.

Router R1 has a Vif1 interface and is running the multicast service reflection application.

Routers R2, R3, R4 and R5 are not PIM enabled and are running unicast routing only in the service provider network.

Enter these commands on the router running the multicast service reflection application (R1):

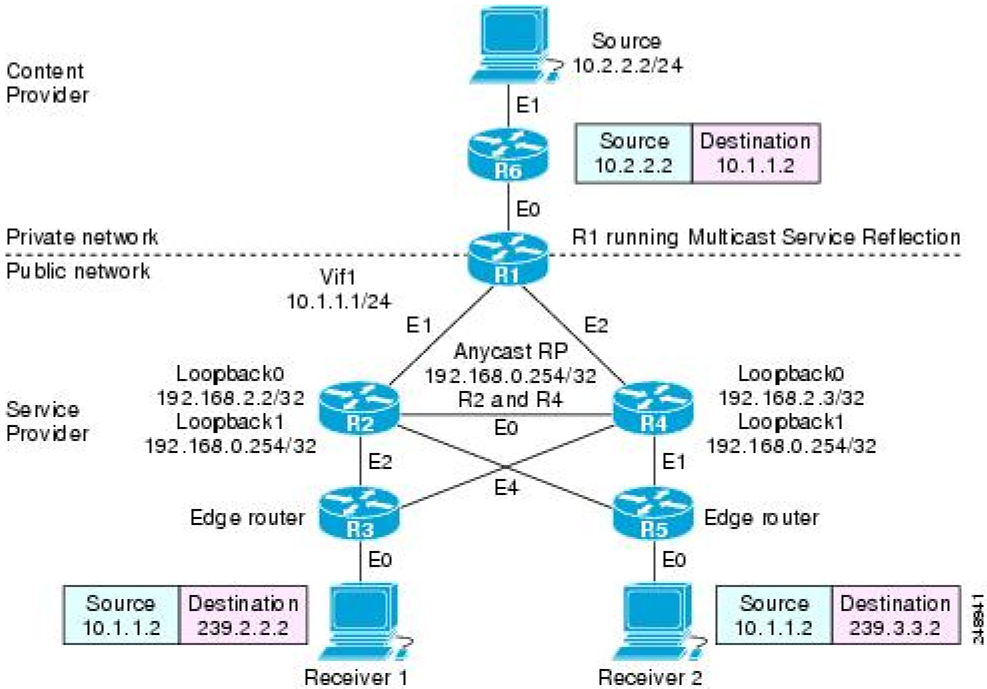
```
configure terminal
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
! Configure the loopback interface for the Service Provider RP
!
interface loopback 0
ip address 192.168.2.1 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.2.1 mcast-content-provider-groups override
ip access-list standard mcast-content-provider-groups
permit 224.1.1.0 0.0.0.255
!
! Configure the Vif1 virtual interface for multicast service reflection
!
interface Vif1
ip address 10.1.1.1 255.255.255.0
ip pim sparse-mode
ip service reflect Ethernet 0 destination 224.1.1.0 to 10.3.3.0 mask-len 24 source 10.1.1.2
ip service reflect Ethernet 0 destination 224.1.1.0 to 10.5.5.0 mask-len 24 source 10.1.1.2
ip igmp static-group 224.1.1.0
ip igmp static-group 224.1.1.1
ip igmp static-group 224.1.1.2
ip igmp static-group 224.1.1.3
.
.
.
ip igmp static-group 224.1.1.255
!
end
```

Example: Unicast-to-Multicast Destination Splitting

The following example shows how to implement multicast service reflection (unicast-to-multicast destination splitting, where the unicast single stream is converted into two unique multicast streams) in a service provider network. Unicast-to-Multicast Destination Splitting allows service providers to translate externally received content provider unicast destination addresses to multiple multicast destination addresses that conform to the service provider's internal addressing policy.

This example uses the topology illustrated in the following figure.

Figure 29: Multicast Service Reflection (Unicast-to-Multicast Destination Splitting) in a Service Provider Network: Example Topology



In this example topology, a content provider is sending financial market information to a service provider, which in turn is sending that information to active receivers (brokerage houses). The service provider may be receiving market data from multiple content providers.

Router R1 is an edge router in the service provider's PIM domain.

Router R1 has a Vif1 interface and is running the multicast service reflection application.

Routers R2 and R4 have multiple loopback interfaces and are acting as anycast RPs for the 239.2.2.0 and 239.3.3.0 address ranges.

Router R3 and R5 are other edge routers in the service provider's PIM domain.

Enter these commands on the router running the multicast service reflection application (R1):

```
configure terminal
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
ip pim rp-address 192.168.2.1 mcast-content-provider-groups override
ip pim rp-address 192.168.0.254 mcast-service-provider-groups override
ip access-list standard mcast-content-provider-groups
permit 224.1.1.0 0.0.0.255
ip access-list standard mcast-service-provider-groups
permit 239.2.2.0 0.0.0.255
ip access-list standard mcast-service-provider-groups
permit 239.3.3.0 0.0.0.255
!
! Configure the Vif1 virtual interface for multicast service reflection
!
interface Vif1
ip address 10.1.1.1 255.255.255.0
```

```
ip pim sparse-mode
ip service reflect Ethernet 0 destination 10.1.1.2 to 239.3.3.2 mask-len 32 source 10.1.1.2
ip service reflect Ethernet 0 destination 10.1.1.2 to 239.2.2.2 mask-len 32 source 10.1.1.2
ip route 10.1.1.2 255.255.255.255 vif1
```

Enter these commands on the R2 router that is the anycast RP in the service provider network:

```
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
interface loopback 0
ip address 192.168.2.2 255.255.255.255
ip pim sparse-mode
!
interface loopback 1
description --- Anycast RP ---
ip address 192.168.0.254 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.0.254 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups
permit 239.2.2.0 0.0.0.255
ip access-list standard mcast-service-provider-groups
permit 239.3.3.0 0.0.0.255
!
ip msdp peer 192.168.2.3 connect-source Loopback0
ip msdp originator-id Loopback0
```

Enter these commands on the R4 router that is the anycast RP in the service provider network:

```
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
interface loopback 0
ip address 192.168.2.3 255.255.255.255
ip pim sparse-mode
!
interface loopback 1
description --- Anycast RP ---
ip address 192.168.0.254 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.0.254 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups
permit 239.2.2.0 0.0.0.255
ip access-list standard mcast-service-provider-groups
permit 239.3.3.0 0.0.0.255
ip msdp peer 192.168.2.2 connect-source Loopback0
ip msdp originator-id Loopback0
```

Enter these commands on all of the other routers in the service provider network:

```
ip multicast-routing distributed
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
ip pim rp-address 192.168.0.254 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups
permit 239.2.2.0 0.0.0.255
ip access-list standard mcast-service-provider-groups
```

```
permit 239.3.3.0 0.0.0.255
end
```

Verifying Multicast Service Reflection Configuration

Use the following show commands to verify Multicast Service Reflection configuration:

- **show ip cef**
- **show ip mroute**
- **show ip mfib**
- **show platform hardware qfp active interface**
- **show ip multicast**
- **show platform hardware qfp active feature uni-sr**

Use the **show ip cef** command to display a summary of the Cisco Express Forwarding (CEF) Information Base (FIB). This command is applicable only to unicast-to-multicast scenarios.

```
interface Vif1
ip address 10.1.1.1 255.255.255.0
ip pim sparse-mode
ip service reflect Ethernet 0 destination 10.1.1.2 to 239.3.3.2 mask-len 32 source 10.1.1.2
end
ip route 10.1.1.2 255.255.255.255 vif1
```

For the above configuration, the **show ip cef** command displays the following output:

```
router# show ip cef
10.1.1.2/32 attached Vif1
```

Use the **show ip mroute** command to display the contents of the multicast routing (mroute) table:

```
router# show ip mroute
IP Multicast Routing Table Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C
 - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT, M
 - MSDP created entry,
E - Extranet, X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement, U - URD,

I - Received Source Specific Host Report, Z - Multicast Tunnel, z - MDT-data group sender,
 Y - Joined MDT-data group,
y - Sending to MDT-data group, G - Received BGP C-Mroute, g - Sent BGP C-Mroute, N - Received
 BGP Shared-Tree Prune,
n - BGP C-Mroute suppressed, Q - Received BGP S-A Route, q - Sent BGP S-A Route, V - RD &
Vector, v - Vector,
p - PIM Joins on route, x - VxLAN group, c - PFP-SA cache created entry, * - determined by
 Assert,
# - iif-starg configured on rpf intf, e - encap-helper tunnel flag, l - LISP decap ref count
 contributor
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join, t - LISP
 transit group
Timers: Uptime/Expires Interface state: Interface, Next-Hop or VCD, State/Mode (*, 239.0.0.0),
 00:04:49/stopped, RP 192.168.0.254, flags: SJCF
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
GigabitEthernet3, Forward/Sparse, 00:04:49/00:02:39, flags: (
10.1.1.3, 239.0.0.0), 00:00:05/00:02:54, flags: FT
Incoming interface: Vif1, RPF nbr 0.0.0.0
```

```
Outgoing interface list:
GigabitEthernet3, Forward/Sparse, 00:00:05/00:03:24, flags:
```

Use the **show ip mfib** command to display the forwarding entries and interfaces in the Multicast Forwarding Information Base (MFIB):

```
router# show ip mfib 239.3.3.0
(10.1.1.3,239.3.3.0) Flags: HW
SW Forwarding: 0/0/0/0, Other: 1/0/1
HW Forwarding: 1379885/2999/96/2249, Other: 0/0/0
Vif1 Flags: A NS
GigabitEthernet4 Flags: F NS
Pkts: 0/0/0 Rate: 0 pps
```

Use the **show platform hardware qfp active interface** command to display the interface status. This command is applicable only to unicast-to-multicast scenarios.

```
router# show platform hardware qfp active interface if-name vif1
Protocol 1 - ipv4_output
FIA handle - CP:0x56518a67abc8 DP:0xe6642780
IPV4_VFR_REFRAG (M)
IPV4_UC_SR_REPLICA_LOOKUP
IPV4_OUTPUT_L2_REWRITE (M)
IPV4_OUTPUT_FRAG (M)
IPV4_OUTPUT_DROP_POLICY (M)
DEF_IF_DROP_FIA (M)
```

Use the **show ip multicast** command to display information about IP multicast global configuration parameters:

```
router# show ip multicast
Multicast Routing: enabled
Multicast Multipath: disabled
Multicast Route limit: No limit
Limit for number of sources per group: 10
Limit for number of OIFs in this MVRF: 8000
The pim is turned off in this MVRF as the configured OIFs limit per MVRF has reached.
Limit for number of OIFs in the router: 8000
Multicast Triggered RPF check: enabled
Multicast Fallback group mode: Dense
```

Use the **show platform hardware qfp active feature uni-sr** command to display the status of the unicast-to-multicast destination translation or splitting. This command is applicable only to unicast-to-multicast scenarios.

```
router# show platform hardware qfp active feature uni-sr
Vif1:
unicast service reflect info:
vif name: Vif1
vif if_handle: 2013
ingress name: GigabitEthernet5
ingress if_handle: 10
replica count: 1
replica rule HW addr: 0x00000000e94e5c10
hash val: 5
prefix: 66.0.0.7/32
replica node info:
    translated source: 10.1.1.3
    translated destination: 239.3.3.0/32
    replica rule HW addr: 0x00000000ead98880
    match: octets 164427264 packets 1712784
```

Troubleshooting and Debugging

Use the **debug ip multicast service-reflect** command to debug multicast destination reflection configuration:

```
debug ip multicast service-reflect
IP multicast service reflect debugging is on
    int vif1 router
    ip service reflect destination 66.0.0.7 to 239.0.0.0 mask-len 32 source 10.1.1.3
    *May 19 12:53:33.566: MSR(0) [default]: Sync SR rule (0.0.0.0, 0.0.0.0) sgrp idx: 0 grp
 idx: 0, pim op: 0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multicast Service Reflection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for Multicast Service Reflection

Feature Name	Releases	Feature Information
Multicast Service Reflection	Cisco IOS XE 3.4S	The Cisco Multicast Service Reflection feature allows you to translate externally received multicast or unicast destination addresses to multicast or unicast addresses that conform to their organization’s internal addressing policy. Using this feature, users do not need to redistribute unicast routes from external sources at the translation boundary into their network infrastructure for Reverse Path Forwarding (RPF) to work properly. In addition, users can receive identical feeds from two ingress points in the network and route them independently. The following command was introduced or modified: **ip service reflect.**
Unicast-to-Multicast Destination Translation and Splitting	Cisco IOS XE Cupertino 17.9.1a	This feature introduces unicast-to-multicast destination translation and splitting configurations. The **show platform hardware qfp active feature uni-sr**, and **debug ip multicast service-reflect** commands were introduced.

CHAPTER 28

Multicast only Fast Re-Route

Multicast only Fast Re-Route (MoFRR) is an IP solution that minimizes packet loss in a network when there is a link or node failure. It works by making simple enhancements to multicast routing protocols like Protocol Independent Multicast (PIM).

MoFRR transmits a multicast join message from a receiver toward a source on a primary path, while also transmitting a secondary multicast join message from the receiver toward the source on a backup path. Data packets are received from both the primary path and the secondary paths. The redundant packets are discarded at topology merge points due to Reverse Path Forwarding (RPF) checks. When a failure is detected on the primary path, the repair is made by changing the interface on which packets are accepted to the secondary interface.Because the repair is local, it is fast--greatly improving convergence times in the event of node or link failures on the primary path.

Prerequisites for MoFRR

- Before performing the tasks in this module, you should be familiar with the concepts described in " IP Multicast Technology Overview " module.
- The tasks in this module assume that IP multicasting has been enabled and that PIM interfaces have been configured using the tasks described in the " Configuring Basic IP Multicast " module.

Restrictions for MoFRR

- The MoFRR feature is disabled by default and must be enabled using the CLI.
- The Equal Cost Multipath Protocol (ECMP) feature is a requirement in order for the MoFRR feature to function.
- MoFRR works only for Specific Multicast (SM) S, G, and Source Specific Multicast (SSM) routes.

- MoFRR is applicable to only IPv4 Multicast, not IPv6 Multicast.
- MoFRR does not support extranet routes.
- MoFRR works where the Reverse Path Forwarding (RPF) lookups are done in a single VRF.
- Both primary and secondary paths should exist in the same multicast topology.
- MoFRR is supported on images supporting IPv4 MFIB only.

Information About MoFRR

Overview of MoFRR

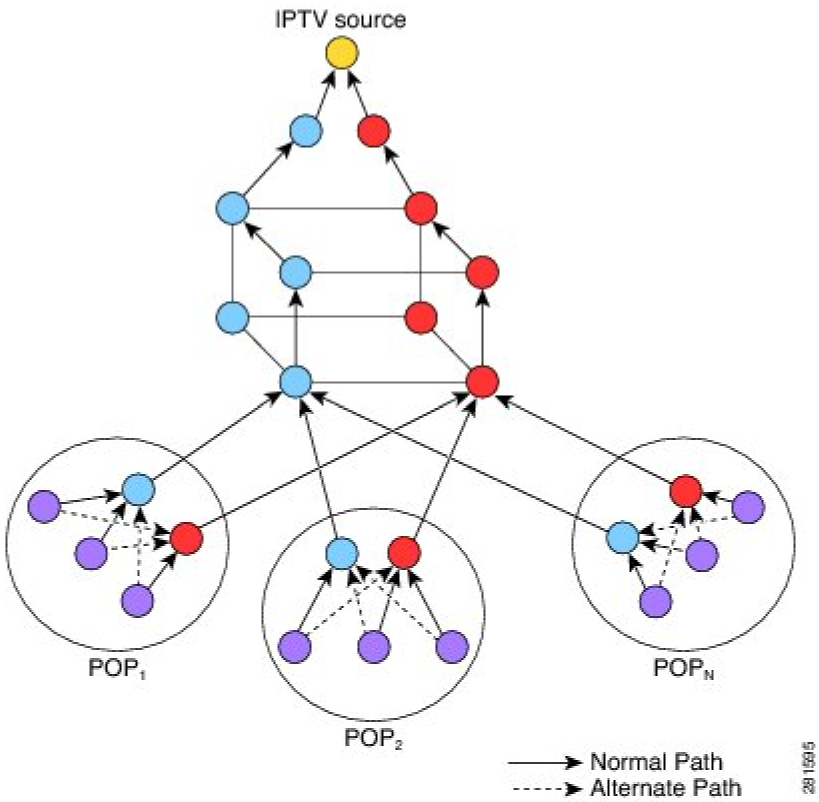
The MoFRR feature provides the ability to minimize packet loss in a network when there is a link or node failure by enhancing, but not changing, multicast routing protocols such as PIM. With MoFRR, multicast routing protocols do not have to wait or depend on unicast routing protocols to detect network failures.

The MoFRR feature can be divided into two planes, red and blue, that are fully disjoint from each other all the way into the points of presence (POPs) as shown in the figure.

This two-plane design eliminates single points of failure in the core network. The upstream full-line arrows indicate the normal path taken when the PIM joins the flow from the POPs toward the source of the network.

MoFRR adds the broken-arrow path where the provider edge (PE) routers send an alternate PIM join to their neighbor toward the source. Each PE router then receives two copies of the same stream, one from the blue plane and one from the red plane. As a result of multicast RPF checks, the following occurs:

- The multicast stream received over the primary path (in the reverse direction of the full-line arrows) is accepted and forwarded to the downstream links.
- The copy of the stream received on the alternate path (in the reverse direction of the broken-line arrows) is discarded



When a routing failure occurs, for example due to a link failure in the blue path, the red upstream router in the red plane becomes the primary upstream router to reach the source. This link to the router then becomes the RPF interface, and the copy of the multicast stream being received on the link is accepted and forwarded to the downstream links.

MoFRR achieves faster convergence by prebuilding the alternate multicast tree and receiving the traffic on that alternate path. The example discussed above is a simple case where there are two paths from each PE device toward the source, one along the blue plane and one along the red plane. MoFRR switchover as a result of routing convergence is expected to be in the order of -200 milliseconds.

How to Configure MoFRR

Enabling MoFRR

Perform this task to configure MoFRR.

Multiple ACL configurations are not allowed. Multicast routes are enabled for MoFRR based on the first match in the ACL.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing [vrf vrf-name] [distributed]**

4. **interface type number [name-tag]**
5. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
6. **ip pim** {**dense-mode**[**proxy-register**{**list** *access-list* | **route-map** *map-name*}] | **passive** | **sparse-mode** | **sparse-dense-mode**}
7. **exit**
8. Repeat Steps 4 through 7 for each interface to be configured.
9. **ip multicast [vrf vrf-name] rpf mofrr {access-list-number | access-list-name} [sticky]**
10. **ip access-list { standard | extended } {** *access-list-name* | *access-list-number* **}**
11. [**sequence-number**] **permit source** [**source-wildcard**[
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing [vrf vrf-name] [distributed]** **Example:** `Device(config)# ip multicast-routing vrf vrf1`	Enables multicast routing. Depending on your release, the **distributed** keyword may not be supported for this command. • In this example, multicast routing is enabled on a vrf instance named vrf1.
Step 4	**interface type number [name-tag]** **Example:** `Device(config)# interface loopback 4`	Selects an interface that is connected to hosts on which PIM can be enabled. • In this example, loopback interface 4 is selected.
Step 5	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** `Device(config-if)# ip address 209.165.200.225 255.255.255.224`	Sets a primary or secondary IP address for the interface. • In this example, 209.165.200.225 is set as the primary address for loopback interface 4.
Step 6	**ip pim** {**dense-mode**[**proxy-register**{**list** *access-list* | **route-map** *map-name*}] | **passive** | **sparse-mode** | **sparse-dense-mode**} **Example:** `Device(config-if)# ip pim sparse-dense-mode`	Enables PIM sparse-dense mode on an interface.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	Repeat Steps 4 through 7 for each interface to be configured.	--
Step 9	**ip multicast [vrf vrf-name] rpf mofrr {access-list-number | access-list-name} [sticky]** **Example:** `Device(config)# ip multicast rpf mofrr 150`	Enables MoFRR for a multicast routing entry that is specific to a source and a group (S, G) matching the ACL. • In this example, MoFRR is enabled for the S, G matching the ACL numbered 150.
Step 10	**ip access-list { standard | extended } {** *access-list-name* **|** *access-list-number* **}** **Example:** `Device(config)# ip access-list extended 150`	Defines a standard or extended IP access list or object group access control list (OGACL) by name or number. • In this example, an ACL numbered 150 is defined. **Note** MoFRR accepts extended ACLs only. It does not accept standard ACLs.
Step 11	[**sequence-number**] **permit source** [**source-wildcard**[**Example:** `Device(config-ext-nacl)# permit 192.168.34.0 0.0.0.255`	Sets conditions to allow a packet to pass a numbered IP access list. • In this example, packets from source address 192.168.34.0 are allowed to pass the ACL.
Step 12	**end** **Example:** `Device(config-ext-nacl)# end`	Exits standard named access list configuration mode and returns to privileged EXEC mode.

Verifying That MoFRR Is Enabled

Perform these steps to verify the configuration of MoFRR.

SUMMARY STEPS

1. **enable**
2. **show ip rpf** [**vrf vrf-name**] **source-address** [**group-address**] [**rd route-distinguisher**]} [**metric**]
3. **show ip mroute** [**vrf** *vrf-name*] [[**active** [*kbps*] [**interface** *type number*] | **bidirectional** | **count** [**terse**] | **dense** | **interface** *type number* | **proxy** | **pruned** | **sparse** | **ssm** | **static** | **summary**] | [*group-address* [*source-address*]] [**count** [*terse*] | **interface** *type number* | **proxy** | **pruned** | **summary**] | [*source-address group-address*] [**count** [**terse**] | **interface** *type number* | **proxy** | **pruned** | **summary**] | [*group-address*] **active** [*kbps*] [**interface** *type number* | **verbose**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip rpf** [**vrf vrf-name**] **source-address** [**group-address**] [**rd route-distinguisher**]} [**metric**] **Example:** `Device# show ip rpf 10.1.1.100`	Displays the information that IP multicast routing uses to perform the Reverse Path Forwarding (RPF) check for a multicast source. **Note** The MoFRR keyword will be displayed in the command output for MoFRR-enabled routes.
Step 3	**show ip mroute** [**vrf** *vrf-name*] [[**active** [*kbps*] [**interface** *type number*] | **bidirectional** | **count** [**terse**] | **dense** | **interface** *type number* | **proxy** | **pruned** | **sparse** | **ssm** | **static** | **summary**] | [*group-address* [*source-address*]] [**count** [*terse*] | **interface** *type number* | **proxy** | **pruned** | **summary**] | [*source-address group-address*] [**count** [**terse**] | **interface** *type number* | **proxy** | **pruned** | **summary**] | [*group-address*] **active** [*kbps*] [**interface** *type number* | **verbose**]] **Example:** `Device# show ip mroute`	Displays the contents of the multicast routing (mroute) table. **Note** The MoFRR keyword will be displayed in the command output for MoFRR-enabled routes.

Configuration Examples for MoFRR

Example Enabling MoFRR

This example shows MoFRR being enabled for the S, G matching ACL 125.

```
Device> enable
Device# configure terminal
Device(config)# ip multicast-routing vrf2
Device(config)# interface fastethernet 0/0
Device(config-if)# ip address 209.165.200.225 0.0.0.0
Device(config-if)# ip pim sparse-dense-mode
Device(config-if)# exit
Device(config)# ip multicast rpf mofrr 125
Device(config)# ip access-list
extended 125
Device(config-ext-nacl)# permit 209.165.201.1 255.255.255.224
Device(config-ext-nacl)# end
```

Example Verifying That MoFRR Is Enabled

The smaple output in the following example shows that MoFRR is enabled for the 209.165.200.225 multicast source IP address. The relevant command output is shown in bold.

```
device> enable
Device# show ip rpf 209.165.200.225
RPF information for ? (209.165.200.225) MoFRR Enabled
  RPF interface: Ethernet1/4
  RPF neighbor: ? (209.165.201.1)
  RPF route/mask: 255.255.255.224
  RPF type: unicast (ospf 200)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv4 unicast base
  Secondary RPF interface: Ethernet1/3
  Secondary RPF neighbor: ? (209.165.202.129)
```

For a detailed explanation of the output, see the **show ip rpf** command in the *Cisco Ip Multicast Command Reference*.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*
Overview of the IP multicast technology area	IP Multicast Technology Overview module
Concepts, tasks, and examples for configuring an IP multicast network using PIM	Configuring a Basic IP Multicast module

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	--

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MoFRR

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for MoFRR

Feature Name	Releases	Feature Information
MoFRR	Cisco IOS XE Release 3.2S 15.2(3)T 15.1(2)SY	The MoFRR feature provides the ability to minimize packet loss in a network when there is a link or node failure by enhancing, but not changing, multicast routing protocols such as PIM. With MoFRR, multicast routing protocols do not have to wait or depend on unicast routing protocols to detect network failures. The following commands were introduced or modified: **ip access-list, ip multicast rpf mofrr, ip multicast-routing, permit (IP), show ip mroute, show ip rpf** .

CHAPTER 29

Multicast Forwarding Information Base Overview

The Multicast Forwarding Information Base (MFIB) architecture provides modularity and separation between the multicast control plane (Protocol Independent Multicast [PIM] and Internet Group Management Protocol [IGMP]) and the multicast forwarding plane (MFIB). This architecture is used in Cisco IOS IPv6 multicast implementations. With the introduction of the IPv4 MFIB infrastructure, the Cisco IOS IPv4 multicast implementation has been enhanced, making the MFIB forwarding model the only forwarding engine used.

Information About the Multicast Forwarding Information Base

Benefits of the MFIB Architecture

- Simplifies multicast operation through the separation of the control and forwarding planes.
- Protects mission critical multicast applications by enabling new services such as multicast high availability (HA).
- Eliminates the need for the route cache maintenance associated with demand caching schemes such as multicast fast switching.

Types of Multicast Tables

The following tables are used for general multicast routing and forwarding:

- IGMP--Contains local IGMP memberships on the router.
- Multicast Route (Mroute)--Contains (*, G) and (S, G) multicast states on the router (including PIM mode, incoming interfaces, and outgoing interfaces).
- Multicast Source Discovery Protocol (MSDP)--Contains all Source-Active (SA) messages.
- Multicast Routing Information Base (MRIB)--Contains (*, G), (S, G), and (*, G/m) MRIB entries.
- MFIB--Contains (*, G), (S, G), and (*, G/m) MFIB entries.

Multicast tables can be further defined by the following contexts:

- Global--Non-VRF context.
- VRF--Layer-3 VPN context.
- IPv4--IPv4 address family context.
- IPv6--IPv6 address family context.

Types of Multicast Entries

- (*, G)--Shared tree entries used by PIM sparse mode (PIM-SM) and bidirectional PIM (bidir-PIM).
- (S, G)--Source tree entries used by PIM-SM and Source Specific Multicast (PIM-SSM).
- (*, G/mask)--Shared tree entries used by the bidir-PIM and the MFIB.

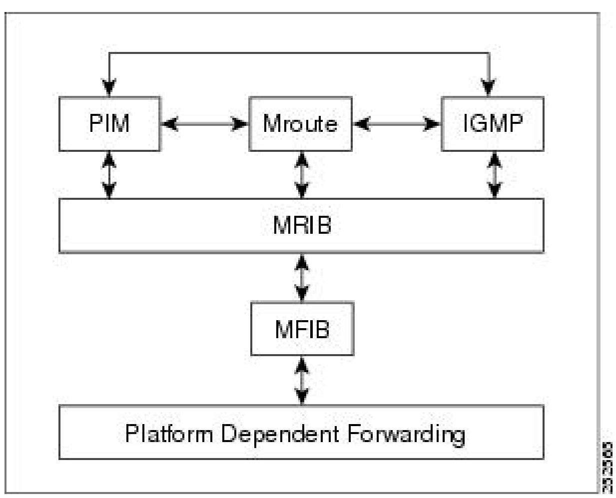
Note For more information about (*, G/mask) entries, see the Introduction of New Multicast Forwarding Entries, on page 355 section.

MFIB Components

The following sections describe the components that make up the MFIB architecture:

The figure illustrates the components that make up the MFIB architecture.

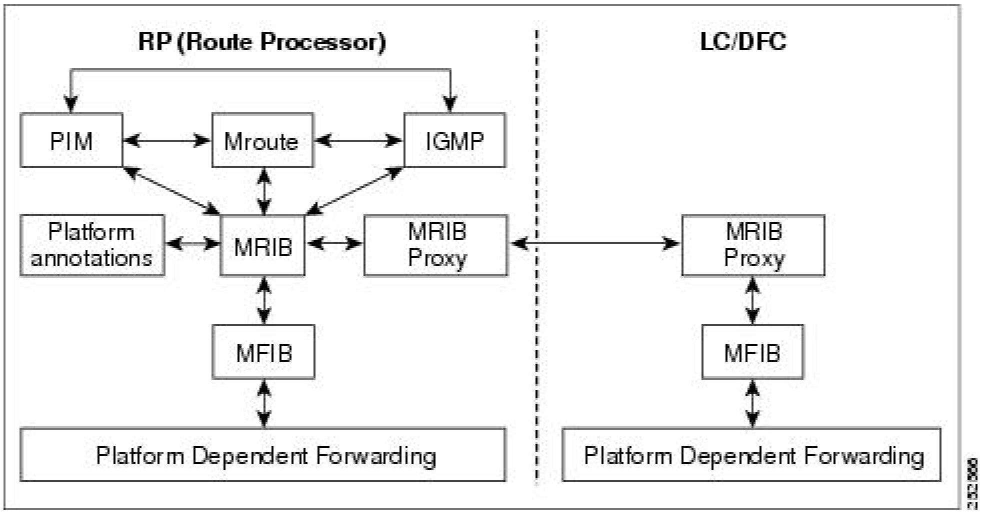
Figure 30: IPv4 MFIB Architecture



Note When you enter the **show ip mrib client** command on a multicast router in an IP multicast network, PIM, the mroute table, and IGMP will appear as one client to the MRIB. For more information, see the Cisco IOS IP Multicast Command Reference.

The figure illustrates the IPv4 MFIB distributed architecture.

Figure 31: IPv4 MFIB Distributed Architecture



MFIB

The MFIB is a multicast routing protocol independent forwarding engine; that is, it does not depend on PIM or any other multicast routing protocol. It is responsible for:

- Forwarding multicast packets
- Registering with the MRIB to learn the entry and interface flags set by the control plane
- Handling data-driven events that must be sent to the control plane
- Maintaining counts, rates, and bytes of received, dropped, and forwarded multicast packets

Distributed MFIB

Distributed MFIB (dMFIB) is used to switch multicast packets on distributed platforms. dMFIB may also contain platform-specific information on replication across line cards. The basic MFIB routines that implement the core of the forwarding logic are common to all forwarding environments.

dMFIB implements the following functions:

- Distributes a copy of the MFIB to the line cards.
- Relays data-driven protocol events generated in the line cards to PIM.
- Provides an MFIB platform application program interface (API) to propagate MFIB changes to platform-specific code responsible for programming the hardware acceleration engine. This API also includes entry points to switch a packet in software (necessary if the packet is triggering a data-driven event) and to upload traffic statistics to the software.
- Provides hooks to allow clients residing on the Route Processor (RP) to read traffic statistics on demand. (dMFIB does not periodically upload these statistics to the RP.)

The combination of dMFIB and MRIB subsystem (MRIB proxy) also allows the router to have a “customized” copy of the MFIB database in each line card and to transport MFIB-related platform-specific information from the RP to the line cards.

MRIB

The MRIB is the communication channel between MRIB clients. Examples of MRIB clients are PIM, IGMP, the multicast routing (mroute) table, and the MFIB.

MRIB communication is based on the setting and clearing of entry and interface flags. MRIB entries are keyed on source, group, and group mask; and appear as (*, G), (S, G), and (*, G/m) multicast entries in the output of the **show ip mrib route** commands. In addition, every MRIB entry will have a list of interfaces associated with it and each interface will have flags set that describe its forwarding state.

The MRIB does not interpret any entry or interface flags. The flags are significant only to MRIB clients.

Note The MRIB uses different tables for different contexts. MRIB tables are separated by address family to distinguish between IPv4 and IPv6 multicast entries. Each table can further be divided within a VRF or global context.

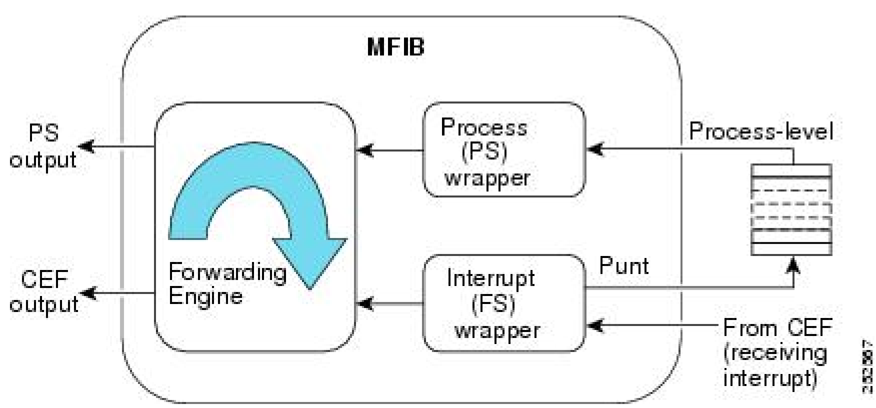
Multicast Control Plane

The multicast control plane is responsible for building and maintaining multicast distribution trees. It consists of PIM, IGMP, and the mroute table, which are MRIB clients in the MFIB architecture. Any changes, additions, and deletions to the mroute table (learned from either PIM or IGMP) are communicated across the MRIB and then distributed to the MFIB for multicast forwarding. Any events related to packet reception that require updates to the control plane are handled between the MRIB and MFIB. Such events include liveness checking, shortest path tree (SPT) switchover, and PIM asserts.

Multicast Packet Forwarding Using the MFIB

The core forwarding engine used by the MFIB is shared by both interrupt-level (fast switching) and process-level forwarding (process switching) as shown in the figure. Multicast packets received with a forwarding entry in the MFIB will be fast-switched by the MFIB, and multicast packets received without a forwarding entry that require the creation of a new forwarding entry will be process-switched by the MFIB.

Figure 32: Multicast Forwarding Engine



MFIB and MRIB Entry and Interface Flags

The table lists the significant MFIB and MRIB entry and interface flags used by IPv4 multicast.

Table 7: Significant MFIB and MRIB Flags

Entry Flag	Table	Description
C	MFIB/MRIB	Connected--Indicates that the MFIB will inform the multicast control plane when it receives traffic from a directly connected source. This flag is used for multicast groups running in PIM sparse mode (PIM-SM) or PIM dense mode (PIM-DM). For PIM-SM, it triggers PIM registration. For PIM-DM, it triggers dense mode flooding. **Note** MFIB entries with Source Specific Multicast (PIM-SSM) and bidirectional PIM (bidir-PIM) environments will not have the C flag set.
DDE	MFIB/MRIB	Data Driven Event--Set by the forwarding plane when an entry is created due to receiving traffic. This flag is used only for HA operations. When there is a RP switchover, entries with this flag set are replayed by the MFIB and signaled to PIM.
ET	MFIB/MRIB	Data Rate Exceeds a Threshold--Set by the forwarding plane when an entry surpasses the data multicast distribution tree (MDT) threshold in a Multicast VPN (MVPN) environment (configured using the **mdt data** command). This flag is used by PIM to initiate the switchover of the data MDT to the default MDT, and vice versa.
IA	MFIB/MRIB	Inherit A Flag--(*, G) entries with the IA flag set indicate that the accept check be performed using its (*, G/mask) parent entry. In other words, the accept check is used to inherit interfaces with the A flag set in the (*, G/m) parent entry. **Note** The IA flag is used for bidir-PIM entries.
K	MFIB/MRIB	Keepalive--Set by PIM to indicate that the entry has been processed and should be stored in the MFIB.
S	MFIB/MRIB	Signal--Indicates the MFIB will notify the multicast control plane when traffic is received on any interface for this entry that does not have the NS flag set.
Interface Flag	Table	Description
A	MFIB/MRIB	Accept--Indicates that multicast data can be accepted on this interface. For example, for PIM-SM and PIM-SSM, the A flag would appear on the Reverse Path Forwarding (RPF) interface set in the mroute table. **Note** The A flag in the MFIB is cleared if MFIB forwarding has been disabled on the interface using the **no ip mfib forwarding input** command.

Entry Flag	Table	Description
F	MFIB/MRIB	Forward--Indicates that multicast data can be forwarded out this interface. For example, the interfaces that are in the outgoing interface list in the mroute table will have this flag set. **Note** The F flag in the MFIB is cleared if the MFIB forwarding has been disabled on the interface using the **no ip mfib forwarding output**command.
IC	MFIB/MRIB	Internal Copy--Indicates that a copy of the packet will be processed by the control plane. The IC flag applies to: • Static IGMP joins--Indicates that the **ip igmp join-group** interface command is configured. • Auto-RP groups (224.0.1.39 and 224.0.1.40)--Indicates that the router is participating in Auto-RP. • Linkscope multicast groups (224.0.0.0/24)--Indicates that the router is listening to linkscope multicast groups, which include PIM hellos, PIM joins and prunes, IGMPv2 /v3 reports, and IGP hello packets (Enhanced Interior Gateway Protocol [EIGRP], Open Shortest Path First [OSPF], and Routing Information Protocol Version 2 [RIPv2]).
NP	MFIB	Not Platform Switched--Indicates that this interface is not being hardware switched. The NP flag is an MFIB specific flag.
NS	MFIB/MRIB	Negate Signal--Indicates that the MFIB will notify the multicast control plane when traffic is received on the specified interface, if the S flag is not set. The NS flag is used for: • SPT switchover in PIM-SM--The NS flag is set on the (*, G) accept interface towards the RP to trigger SPT switchover. • Asserts--The NS-flag is set on (*, G) and (S, G) forward interfaces to trigger PIM asserts. • Liveness checking for active sources in PIM-SM--The NS flag is set on the (S, G) accept interface toward the source to check for active sources. • Proxy registers that enable a PIM-DM domain to register within a PIM-SM domain--The NS flag is set on the (S, G) accept interface where the **ip pim dense-mode proxy-register** command is configured. **Note** For PIM-SSM, the accept interface entries will not have the NS flag set. PIM-SSM neither performs SPT-switchover nor liveness checking. **Note** For PIM-SM, entries that have the **ip pim spt-threshold infinity** command configured globally will not have the NS flag set on their accept interfaces because SPT switchover will be disabled.

Entry Flag	Table	Description
RA	MFIB	MRIB Accept--The RA flag is an MFIB-specific flag. The MFIB sets this flag when the MRIB sets the A flag on an interface.
RF	MFIB	MRIB Forward--The RF flag is an MFIB-specific flag. The MFIB sets this flag when the MRIB sets the F flag on an interface.

Introduction of New Multicast Forwarding Entries

The MFIB architecture introduces (*, G/mask) entries to describe a group range present in a router's local group-to-RP mapping cache (static, Auto-RP, Bootstrap Router [BSR]).

(*, G/mask) entries are used by the MFIB to:

- Create (S, G) entries if they are not already present in the MFIB table (for PIM-SM)
- Create (*, G) entries along source-only branches (for bidir-PIM)
- Forward multicast traffic along shared-tree branches (for bidir-PIM)

Note (*, G/mask) entries are present until the group-to-RP mapping cache either times out or is cleared.

Introduction of PIM Tunnel Interfaces

The MFIB architecture introduces PIM tunnel interfaces. PIM tunnel interfaces are used by the MFIB for the PIM-SM registration process. Two types of PIM tunnel interfaces are used by the MFIB:

- A PIM encapsulation tunnel (PIM Encap Tunnel)
- A PIM decapsulation tunnel (PIM Decap Tunnel)

The PIM Encap Tunnel interface is dynamically created whenever a group-to-RP mapping is learned (via Auto-RP, BSR, or static RP configuration). The PIM Encap Tunnel interface is used to encapsulate multicast packets sent by first-hop designated routers (DRs) that have directly connected sources.

Similar to the PIM Encap Tunnel, the PIM Decap Tunnel interface is dynamically created--with the exception that it is created on the RP only whenever a group-to-rp mapping is learned. The PIM Decap Tunnel interface is used by the RP to decapsulate PIM registers.

Note PIM tunnels will not appear in the running configuration. To display information about PIM Tunnel interfaces, use the **show ip pim tunnel** command.

The following syslog message will appear when a PIM tunnel interface is created:

```
* %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel<interface_number>, changed state
to up
```

MFIB Statistics Support

In the MFIB forwarding model, the MFIB maintains multicast state packet and byte counts and packet rates. The MFIB calculates these statistics as it forwards traffic. There is no periodic polling of these statistics by the control plane, nor does the MFIB periodically upload these statistics to the control plane. The MFIB has an API to these statistics allowing the control plane to query multicast counters when requested from the command-line interface (CLI) for the **show ip mroute count**command and for MIB statistics objects.

Where to Go Next

Proceed to the “ Verifying IPv4 Multicast Forwarding Using the MFIB ” module.

Additional References

Related Documents

Related Topic	Document Title
Multicast verification tasks and examples using the MFIB	“ Verifying IPv4 Multicast Forwarding Using the MFIB ” module
IP multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for the Multicast Forwarding Information Base

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 8: Feature Information for Multicast Forwarding Information Base Overview

Feature Name	Releases	Feature Information
IPv4 Multicast Support of the MFIB	15.0(1)M	The MFIB architecture provides modularity and separation between the multicast control plane (PIM and IGMP) and the multicast forwarding plane (MFIB). This architecture is used in Cisco IOS IPv6 and Cisco IOS XR multicast implementations. With the introduction of the IPv4 MFIB infrastructure, the Cisco IOS IPv4 multicast implementation has been enhanced, making the MFIB forwarding model the only forwarding engine used. The following commands were introduced or modified: **clear ip mfib counters**, **debug ip mcache**, **debug ip mfib adjacency**, **debug ip mfib db**, **debug ip mfib fs**, **debug ip mfib init**, **debug ip mfib interface**, **debug ip mfib mrib**, **debug ip mfib pak**, **debug ip mfib platform**, **debug ip mfib ppr**, **debug ip mfib ps**, **debug ip mfib signal**, **debug ip mfib table**, **debug ip mpacket**, **debug ip mrib**, **ip mfib**, **ip mfib cef**, **ip mfib forwarding**, **ip mroute-cache**, **ip multicast cache-headers**, **ip multicast rate-limit**, **ip multicast ttl-threshold**, **ip pim register-rate-limit**, **show ip mcache**, **show ip mfib**, **show ip mfib active**, **show ip mfib count**, **show ip mfib interface**, **show ip mfib route**, **show ip mfib status**, **show ip mfib summary**, **show ip pim interface**, **show ip pim tunne**l.

CHAPTER 30

Verifying IPv4 Multicast Forwarding Using the MFIB

This module describes how to verify IPv4 multicast forwarding using the Multicast Forwarding Information Base (MFIB) in multicast networks operating in Protocol Independent Multicast (PIM) sparse mode (PIM-SM), Source Specific Multicast (PIM-SSM) mode, or bidirectional PIM (bidir-PIM) mode.

Prerequisites for Verifying IPv4 Multicast Forwarding Using the MFIB

- Before performing the tasks in this module, you should be familiar with concepts described in the " Multicast Forwarding Information Base Overview " and " IP Multicast Technology Overview " modules.
- The tasks in this module assume that IP multicast has been enabled and that PIM-SM, PIM-SSM, or bidir-PIM have been configured using the relevant tasks described in the " Configuring Basic IP Multicast " module.

Restrictions for Verifying IPv4 Multicast Forwarding Using the MFIB

- You must be running a software image that supports the IPv4 MFIB infrastructure.

Information About Verifying IPv4 Multicast Forwarding Using the MFIB

Guidelines for Verifying IPv4 Multicast Forwarding Using the MFIB

When you verify IPv4 multicast forwarding using the MFIB in PIM network environments, a useful approach is to begin the verification process on the last-hop designated router (DR), and then continue the verification process on the routers along the SPT for PIM-SM or PIM-SSM (or on the shared tree for bidir-PIM) until the first-hop DR has been reached. The goal of the verification is to ensure that IP multicast traffic is being forwarded properly through an IP multicast network.

Common Commands for Verifying IPv4 Multicast Forwarding Using the MFIB

The table describes the common commands used to verify multicast forwarding using the MFIB.

Table 9: Common IP Multicast Commands for Verifying Multicast Forwarding

Command	Description and Purpose
show ip igmp groups	Displays the multicast groups with receivers that are directly connected to the router and that were learned through Internet Group Management Protocol (IGMP). • Use this command to confirm that the IGMP cache is being properly populated on the last-hop DR for the groups that receivers on the LAN have joined.
show ip mfib	Displays the multicast forwarding entries within the MFIB.
show ip mfib platform	Displays the platform specific multicast forwarding entries within the MFIB (this command is available on select platforms).
show ip mrib route	Displays entries in the Multicast Routing Information Base (MRIB) table set by PIM, IGMP, or the MFIB.
show ip mroute	Displays the contents of the multicast routing (mroute) table.
show ip pim rp mapping	Displays all group-to-Rendezvous Point (RP) mappings of which the router is aware (either configured or learned from Auto-RP or bootstrap router [BSR]). **Note** The **show ip pim rp mapping**command does not apply to routers in a PIM-SSM network because PIM-SSM does not use rendezvous points (RPs).

Common Mroute Flags

When you verify multicast forwarding, it is helpful to start by looking at the control plane using the **show ip mroute** command. The table describes some of the common flags that you will observe in the output of the **show ip mroute** command when verifying multicast forwarding.

Table 10: Common Mroute Flags

Flag	Description
Mode Flags (All Routers)	
S	Sparse—Entry is operating in sparse mode.
s	SSM—GroupEntry is operating in SSM mode.
B	B Bidir—GroupEntry is operating in bidirectional mode.
Last-Hop DR Flags	
C	Connected—Indicates that an IGMPv2 report for the multicast group was received.
I	Received Source Specific Host Report—Indicates that an IGMPv3 report for the multicast group was received.
L	Local—Indicates that the router itself is a member of the multicast group. Examples are groups that are joined locally by the **ip igmp join-group** command, the **ip sap listen** commands, and the well-known Auto-RP groups, 224.0.1.39 and 224.0.1.40. **Note** Locally joined groups are process switched.
J	Joined SPT—Indicates that the SPT threshold is set to 0 kbps and the next (S, G) packet received down the shared tree will trigger an (S, G) join in the direction of the source. **Note** If the SPT threshold is set to infinity (using the **ip pim spt-threshold infinity** command), the J flag will not be set and all (S, G) packets will stay on the shared tree.
First-Hop DR Flags	
F	Register Flag—Indicates that the router is a candidate to register for the multicast group.
(S, G) Forwarding Flag (Routers Along SPT)	

Flag	Description
T	SPT-bit Set—Indicates that packets have been received on the SPT.
Pruned Flag	
P	Pruned—Entry is in a prune state. Multicast traffic for the multicast group will be dropped by the router.

Common MRIB Flags

When you verify multicast forwarding, it is helpful to confirm the communication between the control plane and the MFIB by examining the MRIB using the **show ip mrib route**command. The table describes some of the common flags that you will encounter in the output of the **show ip mrib route**command when verifying multicast forwarding.

Table 11: Common MRIB Flags

Flag	Description
Entry Flags	
C	Connected--When set, this flag should also appear in the MFIB. For more information, see the C Flag, on page 363 description in the Common MFIB Flags, on page 363 section.
IA	Inherited Accept--When set, this flag should also appear in the MFIB. For more information, see the IA Flag, on page 365 description in the Common MFIB Flags, on page 363 section.
Interface Flags	
A	Accept--When set, this flag should also appear in the MFIB. For more information, see the A Flag, on page 366 description in the Common MFIB Flags, on page 363 section.
F	Forward--When set, this flag should also appear in the MFIB. For more information, see the F Flag, on page 367 description in the Common MFIB Flags, on page 363 section.
NS	Negate Signal--When set, this flag should also appear in the MFIB. For more information about this flag, see the NS Flag, on page 368 description in Common MFIB Flags, on page 363 section.

Common MFIB Flags

When you verify multicast forwarding, it is important to examine the MFIB using the **show ip mfib** command to ensure that multicast traffic is being forwarded as expected. This section describes some of the common flags that you will observe in the output of the **show ip mfib** command when verifying multicast forwarding.

C Flag

The table describes the C flag.

Table 12: C Flag Description

Entry Flag	Description
C	Connected--Indicates that the MFIB will inform the multicast control plane when it receives traffic from a directly connected source. This flag is used for multicast groups running in PIM-SM or PIM-DM. For PIM-SM, it triggers PIM registration. For PIM-DM, it triggers dense mode flooding. **Note** PIM-SSM and bidir-PIM MFIB entries will not have the C flag set.

C Flag Sample Output

The following is sample output from the **show ip mfib** command. In this example, the output has been filtered to display only entries that have the C flag set.

```
RP# show ip mfib | inc Flags: C
 (*,224.0.0.0/4) Flags: C
 (*,239.1.1.1) Flags: C
 (*,224.0.1.39) Flags: C
 (*,224.0.1.40) Flags: C
```

Well-Known Groups

(*, 224.0.0.0/4) Flags: C

This entry indicates that a directly connected check is being performed for the multicast range 224.0.0.0/4. The assumption is that this range is in the group-to-RP mapping cache. If it is not in the group-to-RP mapping cache, this entry will not appear. (*, G/m) parent entries, such as this entry, are used when a match for a (*, G) or (S, G) entry is not found. When traffic from a directly connected source matches a parent entry, the appropriate (*, G) and (S, G) entries in the MFIB, MRIB, and mroute tables will be created.

(*, 224.0.1.39) Flags: C

This entry indicates that a directly connected check is being performed for the Auto-RP Announce multicast group. When traffic from a directly connected source matches this entry and no corresponding (S, G) entry is found, the appropriate (S, G) entry will be created in the MFIB, MRIB, and mroute tables.

Note (*, 224.0.1.39) appears in routers that are configured as an RP for Auto-RP using the **ip pim send-rp-announce** command. The C flag will always be set for this entry, whether the multicast group is running in PIM-DM or PIM-SM.

(*, 224.0.1.40) Flags: C

This entry indicates that a directly connected check is being performed for the Auto-RP Discovery multicast group. When traffic from a directly connected source matches this entry and no corresponding (S, G) entry is found, the appropriate (S, G) entry will be created in the MFIB, MRIB, and mroute table.

Note (*, 224.0.1.40) appears on routers that are configured as a Mapping Agent using the **ip pim send-rp-discovery** command. The C flag will always be set for this entry, whether the multicast group is running in PIM-DM or PIM-SM.

Standard Multicast Group Entry

(*, 239.1.1.1) Flags: C

This entry indicates that a directly connected check is being performed for the multicast group 239.1.1.1.

Note 239.1.1.1 was arbitrarily chosen for this example to represent a standard multicast group entry in the **show ip mfib** output; in practice, the multicast group entries that will display in the output will depend upon your multicast network environment and application.

For this example, the (*, 224.0.0.0/4) entry will not be used because (*, 239.1.1.1) is more specific. When traffic from a directly connected source matches the (*, 239.1.1.1) entry and no (S, G) entry match is found, the MFIB will create the appropriate (S, G) entry then inform the multicast control plane to do the same in the mroute table. If the source is sending for the first time, the multicast control plane will then perform PIM registration or dense mode flooding based on the mode running for the multicast group.

K Flag

The table describes the K flag.

Table 13: K Flag Description

Entry Flag	Description
K	Keepalive--Set by PIM to indicate that the entry has been processed and should be stored in the MFIB.

K Flag Sample Output

The K flag is set to indicate that the control plane (PIM/IGMP/TRANS) owns this entry. When the K flag is set the entry stays in the MFIB until the control plane removes it.

If all flags on an entry (or interface) are removed, MFIB deletes the entry. Therefore, the K flag is used to ensure that MFIB keeps the entry in the absence of any other entry flags (or interfaces with flags on the entry).

The following is sample output from the **show ip mfib**command. In this example, the output has been filtered to display only entries that have the K flag set.

Note The K flag is displayed only when the **verbose** keyword is also specified.

RP# **show ip mfib verbose | inc Flags: K**

```
ET - Data Rate Exceeds Threshold, K - Keepalive Forwarding Counts: Pkt Count/Pkts per
second/Avg Pkt Size/Kbits per second
 (*,224.0.0.0/4) Flags: K
 (*,224.0.1.40) Flags: C K
 (*,232.0.0.0/8) Flags: K
 (*,239.0.0.0/8) Flags: K
 (*,239.1.1.1) Flags: IA K
```

IA Flag

The table describes the IA flag.

Table 14: IA Flag Description

Entry Flag	Description
IA	Inherit A Flag--(*, G) entries with the IA flag set indicate that the accept check be performed using its (*, G/mask) parent entry. In other words, the accept check is used to inherit interfaces with the A flag set in the (*, G/m) parent entry. **Note** The IA flag is used for bidir-PIM entries.

IA Flag Sample Output

In the following output from the **show ip mfib** and **show ip pim rp-mapping**commands, the multicast group 239.195.1.1 is running bidir-PIM and there are two entries: (*, 239.195.1.1) and (*, 239.195.0.0/16). The (*, 239.195.1.1) entry indicates that there is an interested receiver in the network. The parent entry, (*, 239.195.0.0/16), indicates that there is a bidir-PIM group-to-RP mapping. The (*, 239.195.1.1) entry will be used for forwarding multicast traffic for the multicast group 239.195.1.1. The (*, 239.195.1.1) entry will also have the IA flag set, indicating it will inherit the following interfaces from its parent entry for performing accept checks: Serial interface 4/0, Serial interface 2/0, GigabitEthernet interface 0/0/0, and Null interface 0.

Note The portions of output relevant to the IA flag are highlighted in bold.

```
Router# show ip mfib 239.195.1.1
 (*,239.195.1.1) Flags: IA
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial4/0 Flags: F
     Pkts: 0/0
   GigabitEthernet0/0/0 Flags: F
     Pkts: 0/0
Router# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial4/0
 Flags: A
```

```
 F
     Pkts: 0/0
   Serial2/0
Flags: A
   GigabitEthernet0/0/0
 Flags: A
   Null0
 Flags: A
Router# show ip pim rp mapping
     PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 00:49:10, expires: 00:02:19
```

A Flag

The table describes the A flag.

Table 15: A Flag Description

I/O Flag	Description
A	Accept--Indicates that multicast data can be accepted on this interface. For example, for PIM-SM and PIM-SSM, the A flag would appear on the Reverse Path Forwarding (RPF) interface set in the mroute table. **Note** The A flag in the MFIB is cleared if MFIB forwarding has been disabled on the interface using the **no ip mfib forwarding input** command.

A Flag Sample Output

Interfaces with the A flag set in the MFIB correspond to the incoming interfaces for their respective mroute entries, as shown in the following output for the multicast group 239.1.1.1:

Note The portions of sample output relevant to the A flag are highlighted in bold.

```
Router# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 366/0/28/0, Other: 0/0/0
   Serial4/0 Flags: A
NS
   GigabitEthernet0/0/0 Flags: F NS
     Pkts: 366/0
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 107/10/28/2, Other: 1/1/0
   Serial2/0 Flags: A
   GigabitEthernet0/0/0 Flags: F NS
     Pkts: 106/1
Router# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:00:40/stopped, RP 192.168.6.6, flags: SJC
  Incoming interface: Serial4/0
, RPF nbr 192.168.67.6
  Outgoing interface list:
```

```
    GigabitEthernet0/0/0, Forward/Sparse, 00:00:40/00:02:59
(192.168.1.2, 239.1.1.1), 00:00:03/00:02:56, flags: JT
  Incoming interface: Serial2/0
, RPF nbr 192.168.37.3
  Outgoing interface list:
    GigabitEthernet0/0/0, Forward/Sparse, 00:00:03/00:02:59
```

F Flag

The table describes the F flag.

Table 16: F Flag Description

I/O Flag	Description
F	Forward--Indicates that multicast data can be forwarded out this interface. For example, the interfaces that are in the outgoing interface list in the mroute table will have this flag set. **Note** The F flag in the MFIB is cleared if the MFIB forwarding has been disabled on the interface using the **no ip mfib forwarding output**command.

F Flag Sample Output

Interfaces with the F flag set in the MFIB correspond to interfaces in the outgoing interface list for their respective mroute entries, as shown in the following output for the multicast group 239.1.1.1:

Note The portions of sample output relevant to the F flag are highlighted in bold.

```
Router# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 366/0/28/0, Other: 0/0/0
   Serial4/0 Flags: A NS
   GigabitEthernet0/0 Flags: F
 NS
     Pkts: 366/0
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 107/10/28/2, Other: 1/1/0
   Serial2/0 Flags: A
   GigabitEthernet0/0/0 Flags: F
 NS
     Pkts: 106/1
Router# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:00:40/stopped, RP 192.168.6.6, flags: SJC
  Incoming interface: Serial4/0, RPF nbr 192.168.67.6
  Outgoing interface list:
  GigabitEthernet0/0
, Forward/Sparse, 00:00:40/00:02:59
(192.168.1.2, 239.1.1.1), 00:00:03/00:02:56, flags: JT
  Incoming interface: Serial2/0, RPF nbr 192.168.37.3
  Outgoing interface list:
    GigabitEthernet0/0/0
, Forward/Sparse, 00:00:03/00:02:59
```

NS Flag

The table describes the NS flag.

Table 17: NS Flag Description

I/O Flag	Description
NS	Negate Signal--Indicates the MFIB will notify the multicast control plane when traffic is received on the specified interface, if the S flag is not set. The NS flag is used for: • SPT switchover in PIM-SM--The NS flag is set on the (*, G) accept interface toward the RP to trigger SPT switchover. • Asserts--The NS flag is set on (*, G) and (S, G) forward interfaces to trigger PIM asserts. • Liveness checking for active sources in PIM-SM--The NS flag is set on the (S, G) accept interface toward the source to check for active sources. • Proxy-registers that enable a PIM-DM domain to register within a PIM-SM domain--The NS flag is set on the (S, G) accept interface where the **ip pim dense-mode proxy-register** command is configured. **Note** For PIM-SSM, the accept interface entries will not have the NS flag set. PIM-SSM neither performs SPT-switchover nor liveness checking. **Note** For PIM-SM, entries that have **ip pim spt-threshold infinity** configured globally will not have the NS flag set on their accept interfaces because SPT switchover will be disabled.

IC Flag

The table describes the IC flag.

Table 18: IC Flag Description

I/O Flag	Description
IC	Internal Copy--Indicates that a copy of the packet will be processed by the control plane. The IC flag applies to: • Static IGMP joins--Indicates that the **ip igmp join-group** interface command is configured. • Auto-RP groups (224.0.1.39 and 224.0.1.40)--Indicates that the router is participating in Auto-RP. • Linkscope multicast groups (224.0.0.0/24)--Indicates that the router is listening to linkscope multicast groups, which include PIM hellos, PIM joins and prunes, IGMPv2 /v3 reports, and Interior Gateway Protocol hello packets (Enhanced Interior Gateway Routing Protocol [EIGRP], Open Shortest Path First [OSPF], and Routing Information Protocol version 2 [RIPv2]).

IC Flag Sample Output

> **Note** The configuration lines and portions of sample output relevant to the IC flag are highlighted in bold.

Static IGMP Join

The following example configures a static IGMP join for multicast group 239.1.1.1 under GigabitEthernet interface 0/0/0:

```
interface GigabitEthernet0/0/0
 ip address 192.168.7.7 255.255.255.0
 ip pim sparse-mode
 ip igmp join-group 239.1.1.1
```

The following sample output from the **show ip mfib** command verifies that the IC flag is set for GigabitEthernet interface 0/0/0:

```
Router# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 366/0/28/0, Other: 0/0/0
   Serial4/0 Flags: A NS
   GigabitEthernet0/0/0
 Flags: F IC
 NS
     Pkts: 366/0
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 3978/10/28/2, Other: 1/1/0
   Serial2/0 Flags: A
   GigabitEthernet0/0/0
Flags: F IC
 NS
     Pkts: 3977/1
```

> **Note** The **ip igmp static-group** command will not set the IC flag.

Auto-RP Groups 224.0.1.39 and 224.0.1.40

The following output from the **show ip igmp group**and **show ip mfib** command confirms that this router is both an RP and Mapping Agent and has the IC flag set to process switch Auto-RP multicast packets.

> **Note** All routers, including the RP, will join the multicast group 224.0.1.40. In addition to the multicast group 224.0.1.40, Mapping Agents will also join 224.0.1.39.

```
Router# show ip igmp group
IGMP Connected Group Membership
Group Address    Interface              Uptime    Expires   Last Reporter   Group Accounted
224.0.1.39        Serial2/0
               02:57:51  stopped   192.168.26.6
224.0.1.39        Serial1/0
```

```
                02:57:51  stopped   192.168.67.6
224.0.1.39       GigabitEthernet0/0/0
    02:57:51  00:02:11  192.168.16.6
224.0.1.39       Loopback0
                02:57:51  00:02:07  192.168.6.6
224.0.1.40       Loopback0
                02:57:51  00:02:11  192.168.6.6
239.1.1.1        GigabitEthernet0/0/0     02:58:51  00:02:13  192.168.16.6
224.1.1.1        GigabitEthernet0/0/0     02:58:51  00:02:13  192.168.16.6
Router# show ip mfib 224.0.1.39
 (*,224.0.1.39) Flags: C
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Loopback0 Flags: F IC NS
     Pkts: 0/0
   Serial2/0 Flags: F IC NS
     Pkts: 0/0
   Serial1/0 Flags: F IC NS
     Pkts: 0/0
   GigabitEthernet0/0/0 Flags: F IC NS
     Pkts: 0/0
 (192.168.6.6,224.0.1.39) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Loopback0 Flags: A IC
   Serial2/0 Flags: F IC NS
     Pkts: 0/0
   Serial1/0 Flags: F IC NS
     Pkts: 0/0
   GigabitEthernet0/0/0 Flags: F IC NS
     Pkts: 0/0
Router# show ip mfib 224.0.1.40
 (*,224.0.1.40) Flags: C
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Loopback0 Flags: F IC NS
     Pkts: 0/0
   Serial2/0 Flags: F NS
     Pkts: 0/0
   Serial1/0 Flags: F NS
     Pkts: 0/0
 (192.168.6.6,224.0.1.40) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Loopback0 Flags: A IC
   Serial2/0 Flags: F NS
     Pkts: 0/0
   Serial1/0 Flags: F NS
     Pkts: 0/0
```

Linkscope Multicast Groups 224.0.0.0/24

The following output from the **show ip mfib linkscope** command confirms that the IC flag is set to process multicast control packets:

```
Router# show ip mfib linkscope
.
.
.
(*,224.0.0.1) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Loopback0 Flags: IC
   Serial4/0 Flags: IC
   Serial3/0 Flags: IC
   Serial2/0 Flags: IC
   GigabitEthernet1/0/0 Flags: IC
   GigabitEthernet0/0/0 Flags: IC
```

```
(*,224.0.0.2) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  Loopback0 Flags: IC
  Serial4/0 Flags: IC
  Serial3/0 Flags: IC
  Serial2/0 Flags: IC
  GigabitEthernet1/0/0 Flags: IC
  GigabitEthernet0/0/0 Flags: IC
(*,224.0.0.13) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  Loopback0 Flags: IC
  Serial4/0 Flags: IC
  Serial3/0 Flags: IC
  Serial2/0 Flags: IC
  GigabitEthernet1/0/0 Flags: IC
  GigabitEthernet0/0/0 Flags: IC
(*,224.0.0.22) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  Loopback0 Flags: IC
  Serial4/0 Flags: IC
  Serial3/0 Flags: IC
  Serial2/0 Flags: IC
  GigabitEthernet1/0/0 Flags: IC
  GigabitEthernet0/0/0 Flags: IC
```

PIM Tunnel Interfaces

PIM tunnel interfaces are used by MFIB for the PIM-SM registration process. Two types of PIM tunnel interfaces are used by the MFIB:

- A PIM encapsulation tunnel (PIM Encap Tunnel)
- A PIM decapsulation tunnel (PIM Decap Tunnel)

The PIM Encap Tunnel is dynamically created whenever a group-to-RP mapping is learned (via Auto-RP, BSR, or static RP configuration). The PIM Encap Tunnel is used to encapsulate multicast packets sent by first-hop DRs that have directly connected sources.

Similar to the PIM Encap Tunnel, the PIM Decap Tunnel interface is dynamically created--with the exception that it is created on the RP only whenever a group-to-RP mapping is learned. The PIM Decap Tunnel interface is used by the RP to decapsulate PIM registers.

Note PIM tunnels will not appear in the running configuration.

The following syslog message will appear when a PIM tunnel interface is created:

```
* %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel<interface_number>, changed state
to up
```

How to Verify IPv4 Multicast Forwarding Using the MFIB

Verifying IPv4 Multicast Forwarding Using the MFIB for PIM-SM PIM-SSM and Bidir-PIM

Perform this optional task to verify multicast forwarding using the MFIB in PIM-SM, PIM-SSM, and bidir-PIM networks.

When you verify IPv4 multicast forwarding using the MFIB in PIM network environments, a useful approach is to begin the verification process on the last-hop DR, and then continue the verification process on the routers along the SPT for PIM-SM or PIM-SSM (or on the shared tree for bidir-PIM) until the first-hop DR has been reached. The goal of the verification is to ensure that IP multicast traffic is being forwarded properly through an IP multicast network.

Before you begin

The tasks in this module assume that IP multicast has been enabled and that PIM-SM, PIM-SSM, or bidir-PIM have been configured.

Note You must be running a Cisco software image that supports the IPv4 MFIB infrastructure.

>

SUMMARY STEPS

1. **enable**
2. **show ip mroute**
3. **show ip mrib route**
4. **show ip mfib**
5. show ip pim rp mapping
6. show ip igmp groups

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **show ip mroute**

Displays the contents of the mroute table.

```
Router# show ip mroute
```

Step 3 **show ip mrib route**

Displays the MRIB table.

```
Router# show ip mrib route
```

Step 4 **show ip mfib**

Displays the forwarding entries and interfaces in the MFIB.

```
Router# show ip mfib
```

Step 5 show ip pim rp mapping

Displays all group-to-RP mappings of which the router is aware (either configured or learned from Auto-RP or BSR). Use this command to confirm which router is acting as the RP.

Note The **show ip pim rp mapping**command does not apply to routers in a PIM-SSM network because PIM-SSM does not use RPs.

Sample Output from an RP

The following is sample output from the **show ip pim rp mapping** command. The output confirms that the router in this example is the RP.

Example:

```
RP# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 01:04:03, expires: 00:02:53
```

Sample Output from a Non-RP

The following is sample output from the **show ip pim rp mapping** command. The output confirms that this router is not the RP.

Example:

```
Non-RP# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 00:40:55, expires: 00:02:45
```

Step 6 show ip igmp groups

Displays the multicast groups with receivers that are directly connected to the router and that were learned through IGMP.

Use this command to confirm that the IGMP cache is being properly populated on the last-hop DR for the groups that receivers on the LAN have joined.

```
Router# show ip igmp groups
```

Verifying PIM Tunnel Interfaces for PIM-SM

Perform this optional task verify to verify the PIM tunnel interfaces that are used by the MFIB for the PIM-SM registration process. This task can be performed if you suspect that there may be problems related to PIM-SM registration.

SUMMARY STEPS

1. **enable**
2. **show ip pim rp mapping**
3. **show ip pim tunnel**
4. **show ip mfib**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **show ip pim rp mapping**

Displays all group-to-RP mappings of which the router is aware (either configured or learned from Auto-RP or BSR). Use this command to confirm which router is acting as the RP.

Sample Output from an RP

The following is sample output from the **show ip pim rp mapping** command. The output confirms that the router in this example is the RP.

Example:

```
RP# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 01:04:03, expires: 00:02:53
```

Sample Output from a Non-RP

The following is sample output from the **show ip pim rp mapping** command. The output confirms that this router is not the RP.

Example:

```
Non-RP# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 00:40:55, expires: 00:02:45
```

Step 3 **show ip pim tunnel**

Displays the PIM tunnel interfaces used by the MFIB for the PIM-SM registration process.

Sample Output from an RP (show ip pim tunnel)

The following is output from the **show ip pim tunnel** command. The output is used to verify the PIM Encap and Decap Tunnel on the RP.

Example:

```
RP# show ip pim tunnel
Tunnel0
  Type  : PIM Encap
  RP    : 192.168.6.6*
  Source: 192.168.6.6
Tunnel1
  Type  : PIM Decap
  RP    : 192.168.6.6*
  Source: -
```

Note The asterisk (*) indicates that the router is the RP. The RP will always have a PIM Encap and Decap Tunnel interface.

Sample Output from a Non-RP (show ip pim tunnel) The following is output from the **show ip pim tunnel** command. The output is used to confirm that a PIM Encap Tunnel has been created on a non-RP router.

Example:

```
Non-RP# show ip pim tunnel
Tunnel0
  Type  : PIM Encap
  RP    : 192.168.6.6
  Source: 192.168.67.7
```

Step 4 **show ip mfib**

Displays the forwarding entries and interfaces in the MFIB.

or

show ip mrib route

Displays the MRIB table.

Use either the **show ip mfib** command or the **show ip mrib route** command to verify that the entries registering for PIM-SM have the F flag set for the PIM Encap Tunnel.

Configuration Examples for Verifying IPv4 Multicast Forwarding Using the MFIB

Examples Verifying IPv4 Multicast Forwarding Using the MFIB for PIM-SM

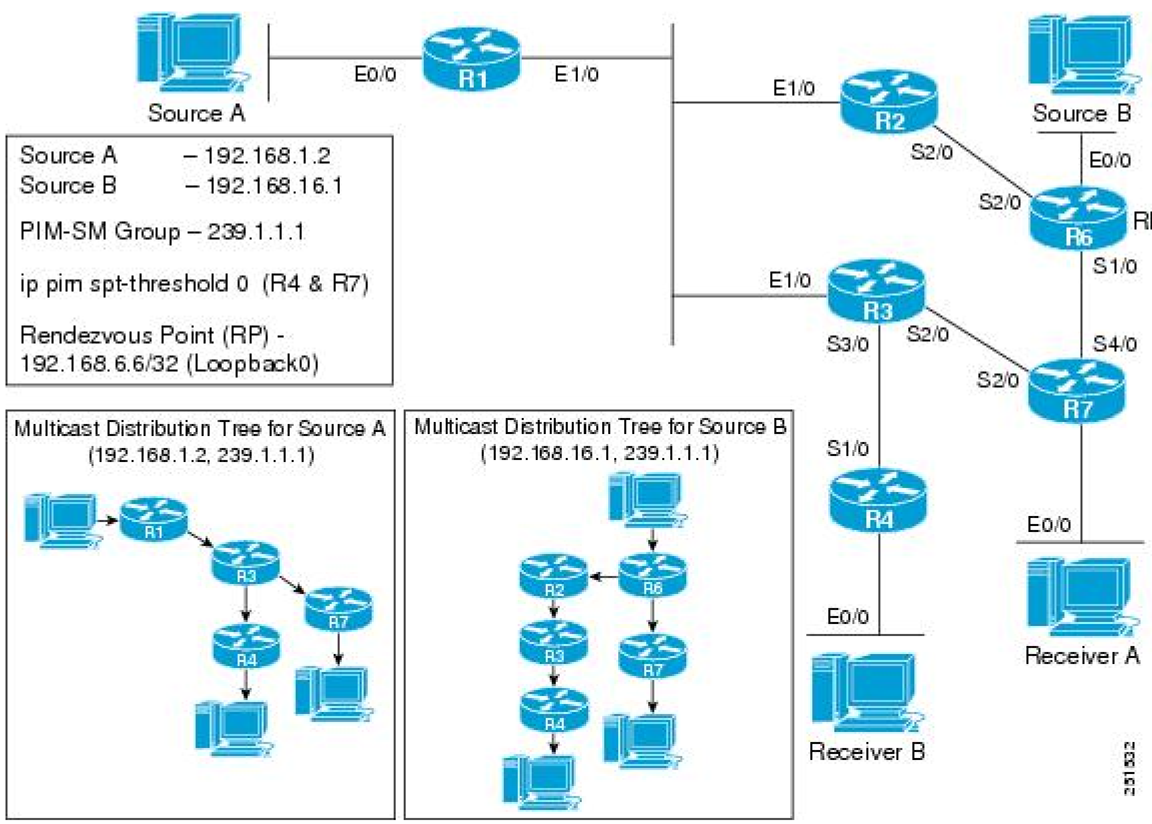
This section contains the following examples for verifying multicast forwarding using the MFIB for PIM-SM networks:

Note The examples in this section were created in a test environment to provide a conceptual view of the multicast environment. The IP addresses, interfaces, and other values are shown as examples only. They do not show real-world deployment values.

PIM-SM Example Active Sources and Interested Receivers - SPT Switchover

The following example shows how to verify multicast forwarding using the MFIB for PIM-SM in a network environment where there are active sources with interested receivers. This verification example is based on the topology shown in the figure.

Figure 33: PIM-SM Example Topology: Active Sources and Interested Receivers (SPT Switchover)



In this verification example, the following conditions apply:

- All routers have the SPT switchover set to the default (**ip pim spt-threshold 0**).
- Because the SPT threshold is set to 0, all last-hop DRs with interested receivers will perform an SPT switchover when multicast traffic is received on the shared tree.
- During the PIM-SM registration process between the first-hop DR and the RP, a PIM tunnel is used. First-hop DRs will have a PIM Encap Tunnel and the RP will have both a PIM Encap and Decap Tunnel. After the PIM-SM registration process completes, PIM tunnels will not be used for multicast forwarding. For more information, see the Verifying PIM Tunnel Interfaces for PIM-SM, on page 374 section.

R1 (First-Hop DR)

```
R1# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R1# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:53:08/stopped, RP 192.168.6.6, flags: SPF
  Incoming interface: Ethernet1/0, RPF nbr 192.168.123.2
  Outgoing interface list: Null
(192.168.1.2, 239.1.1.1), 00:53:08/00:03:12, flags: FT
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse, 00:38:25/00:03:07
R1# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 192.168.123.2 Flags: C
  Ethernet1/0 Flags: A
(192.168.1.2,239.1.1.1) RPF nbr: 0.0.0.0 Flags:
  Ethernet0/0 Flags: A
  Ethernet1/0 Flags: F NS
R1# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 0/0/0/0, Other: 23058/0/23058
   Ethernet1/0 Flags: A
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 23059/10/28/2, Other: 8826/0/8826
   Ethernet0/0 Flags: A
   Ethernet1/0 Flags: F NS
     Pkts: 23058/0
```

R2 (Router Along the SPT)

```
R2# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R2# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:39:05/00:02:53, RP 192.168.6.6, flags: S
  Incoming interface: Serial2/0, RPF nbr 192.168.26.6
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse, 00:39:05/00:02:53
```

```
(192.168.16.1, 239.1.1.1), 00:03:31/00:02:54, flags: T
  Incoming interface: Serial2/0, RPF nbr 192.168.26.6
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse, 00:03:31/00:02:56
(192.168.1.2, 239.1.1.1), 00:39:05/00:02:42, flags: PT
  Incoming interface: Ethernet1/0, RPF nbr 192.168.123.1
  Outgoing interface list: Null
R2# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 192.168.26.6 Flags: C
  Ethernet1/0 Flags: F NS
  Serial2/0 Flags: A
(192.168.1.2,239.1.1.1) RPF nbr: 192.168.123.1 Flags:
  Ethernet1/0 Flags: A
(192.168.16.1,239.1.1.1) RPF nbr: 192.168.26.6 Flags:
  Serial2/0 Flags: A
  Ethernet1/0 Flags: F NS
R2# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 21343/0/28/0, Other: 0/0/0
   Serial2/0 Flags: A
   Ethernet1/0 Flags: F NS
     Pkts: 21343/0
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 21643/0/28/0, Other: 1812/1/1811
   Ethernet1/0 Flags: A
 (192.168.16.1,239.1.1.1) Flags:
   SW Forwarding: 2112/10/28/2, Other: 0/0/0
   Serial2/0 Flags: A
   Ethernet1/0 Flags: F NS
     Pkts: 2112/0
```

R3 (Router Along the SPT)

```
R3# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R3# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:40:32/00:03:19, RP 192.168.6.6, flags: S
  Incoming interface: Ethernet1/0, RPF nbr 192.168.123.2
  Outgoing interface list:
    Serial3/0, Forward/Sparse, 00:40:32/00:03:19
(192.168.16.1, 239.1.1.1), 00:04:58/00:02:29, flags: T
  Incoming interface: Ethernet1/0, RPF nbr 192.168.123.2
  Outgoing interface list:
    Serial3/0, Forward/Sparse, 00:04:58/00:03:26
(192.168.1.2, 239.1.1.1), 00:04:58/00:02:26, flags: T
  Incoming interface: Ethernet1/0, RPF nbr 192.168.123.1
  Outgoing interface list:
    Serial2/0, Forward/Sparse, 00:04:28/00:02:57
    Serial3/0, Forward/Sparse, 00:04:58/00:03:27
R3# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 192.168.123.2 Flags: C
  Serial3/0 Flags: F NS
  Ethernet1/0 Flags: A
(192.168.1.2,239.1.1.1) RPF nbr: 192.168.123.1 Flags:
  Ethernet1/0 Flags: A
  Serial2/0 Flags: F NS
  Serial3/0 Flags: F NS
(192.168.16.1,239.1.1.1) RPF nbr: 192.168.123.2 Flags:
```

```
  Ethernet1/0 Flags: A
  Serial3/0 Flags: F NS
R3# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 42686/0/28/0, Other: 0/0/0
   Ethernet1/0 Flags: A
   Serial3/0 Flags: F NS
     Pkts: 42686/0
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 2984/10/28/2, Other: 0/0/0
   Ethernet1/0 Flags: A
   Serial3/0 Flags: F NS
     Pkts: 2984/0
   Serial2/0 Flags: F NS
     Pkts: 2684/0
 (192.168.16.1,239.1.1.1) Flags:
   SW Forwarding: 2984/10/28/2, Other: 0/0/0
   Ethernet1/0 Flags: A
   Serial3/0 Flags: F NS
     Pkts: 2984/0
```

R4 (Last-Hop DR for Receiver B)

```
R4# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R4# show ip igmp groups 239.1.1.1
IGMP Connected Group Membership
Group Address    Interface       Uptime    Expires    Last Reporter   Group Accounted
239.1.1.1        Ethernet0/0     00:06:39  00:02:56   192.168.4.1
R4# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:42:12/stopped, RP 192.168.6.6, flags: SJC
  Incoming interface: Serial1/0, RPF nbr 192.168.34.3
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 00:42:12/00:02:02
(192.168.16.1, 239.1.1.1), 00:06:37/00:02:16, flags: JT
  Incoming interface: Serial1/0, RPF nbr 192.168.34.3
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 00:06:37/00:02:02
(192.168.1.2, 239.1.1.1), 00:06:37/00:02:19, flags: JT
  Incoming interface: Serial1/0, RPF nbr 192.168.34.3
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 00:06:37/00:02:02
R4# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 192.168.34.3 Flags: C
  Serial1/0 Flags: A NS
  Ethernet0/0 Flags: F NS
(192.168.1.2,239.1.1.1) RPF nbr: 192.168.34.3 Flags:
  Serial1/0 Flags: A
  Ethernet0/0 Flags: F NS
(192.168.16.1,239.1.1.1) RPF nbr: 192.168.34.3 Flags:
  Serial1/0 Flags: A
  Ethernet0/0 Flags: F NS
R4# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 42684/0/28/0, Other: 0/0/0
   Serial1/0 Flags: A NS
   Ethernet0/0 Flags: F NS
     Pkts: 42684/0
```

```
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 3980/10/28/2, Other: 0/0/0
   Serial1/0 Flags: A
   Ethernet0/0 Flags: F NS
     Pkts: 3979/1
 (192.168.16.1,239.1.1.1) Flags:
   SW Forwarding: 3980/10/28/2, Other: 0/0/0
   Serial1/0 Flags: A
   Ethernet0/0 Flags: F NS
     Pkts: 3979/1
```

R6 (RP and First-Hop DR for Source B)

```
R6# show ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:10:53, expires: 00:02:06
R6# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:58:12/00:03:25, RP 192.168.6.6, flags: SF
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/0, Forward/Sparse, 00:43:25/00:03:22
    Serial2/0, Forward/Sparse, 00:43:29/00:03:25
(192.168.1.2, 239.1.1.1), 00:58:12/00:02:47, flags: PT
  Incoming interface: Serial2/0, RPF nbr 192.168.26.2
  Outgoing interface list: Null
(192.168.16.1, 239.1.1.1), 00:58:12/00:03:17, flags: FT
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/0, Forward/Sparse, 00:43:25/00:03:22
    Serial2/0, Forward/Sparse, 00:43:29/00:03:27
R6# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 0.0.0.0 Flags: C
  Serial1/0 Flags: F NS
  Serial2/0 Flags: F NS
  Tunnel1 Flags: A
(192.168.1.2,239.1.1.1) RPF nbr: 192.168.26.2 Flags:
  Serial2/0 Flags: A NS
(192.168.16.1,239.1.1.1) RPF nbr: 0.0.0.0 Flags:
  Ethernet0/0 Flags: A
  Serial1/0 Flags: F NS
  Serial2/0 Flags: F NS
R6# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Tunnel1 Flags: A
   Serial2/0 Flags: F NS
     Pkts: 0/0
   Serial1/0 Flags: F NS
     Pkts: 0/0
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 21604/0/28/0, Other: 39/1/38
   Serial2/0 Flags: A NS
 (192.168.16.1,239.1.1.1) Flags:
   SW Forwarding: 26099/10/28/2, Other: 8827/0/8827
   Ethernet0/0 Flags: A
   Serial2/0 Flags: F NS
     Pkts: 26098/0
```

```
    Serial1/0 Flags: F NS
      Pkts: 26058/0
```

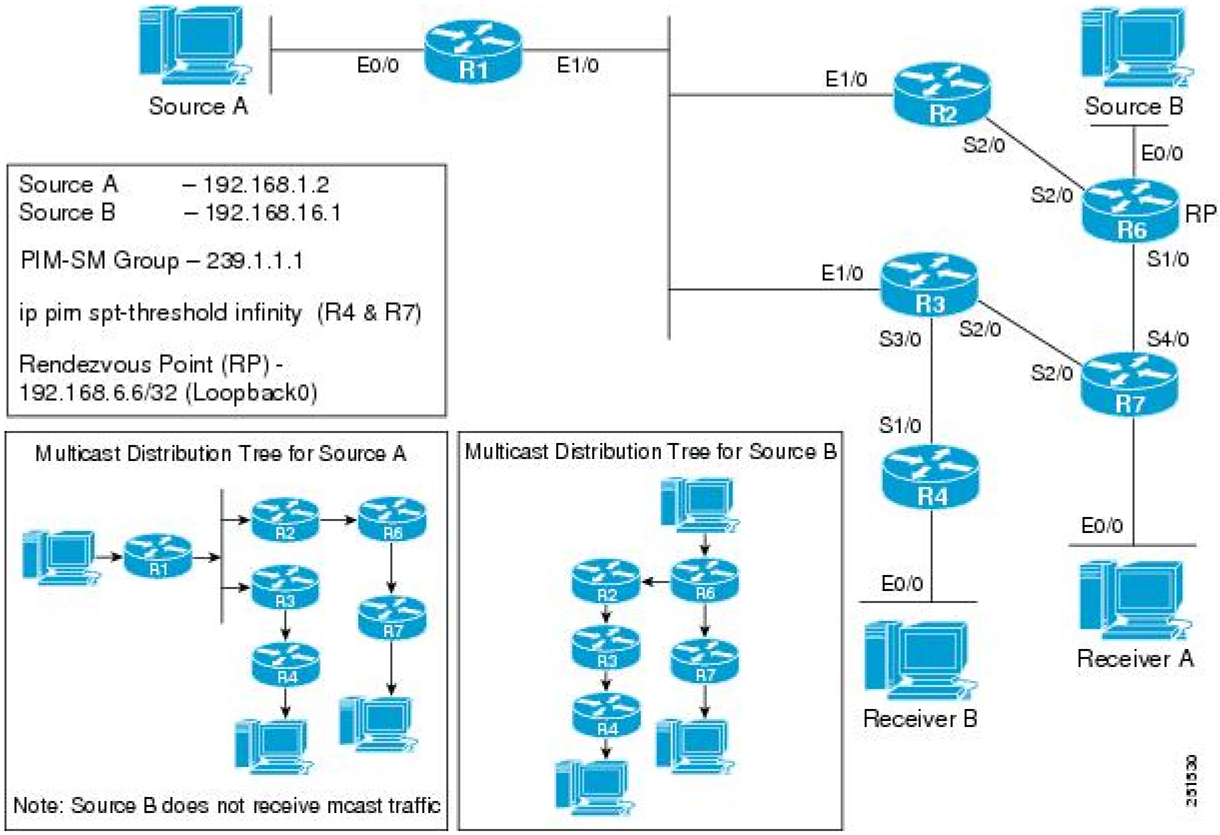
R7 (Last-Hop DR for Receiver A)

```
R7# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R7# show ip igmp groups 239.1.1.1
IGMP Connected Group Membership
Group Address    Interface       Uptime    Expires    Last Reporter   Group Accounted
239.1.1.1        Ethernet0/0     00:08:47  00:02:56   192.168.7.1
R7# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:44:45/stopped, RP 192.168.6.6, flags: SJC
  Incoming interface: Serial4/0, RPF nbr 192.168.67.6
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 00:44:45/00:02:47
(192.168.1.2, 239.1.1.1), 00:08:45/00:02:13, flags: JT
  Incoming interface: Serial2/0, RPF nbr 192.168.37.3
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 00:08:45/00:02:47
(192.168.16.1, 239.1.1.1), 00:08:45/00:02:10, flags: JT
  Incoming interface: Serial4/0, RPF nbr 192.168.67.6
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 00:08:45/00:02:47
R7# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 192.168.67.6 Flags: C
  Serial4/0 Flags: A NS
  Ethernet0/0 Flags: F NS
(192.168.1.2,239.1.1.1) RPF nbr: 192.168.37.3 Flags:
  Serial2/0 Flags: A
  Ethernet0/0 Flags: F NS
(192.168.16.1,239.1.1.1) RPF nbr: 192.168.67.6 Flags:
  Serial4/0 Flags: A
  Ethernet0/0 Flags: F NS
R7# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 43204/0/28/0, Other: 0/0/0
   Serial4/0 Flags: A NS
   Ethernet0/0 Flags: F NS
     Pkts: 43204/0
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 5255/10/28/2, Other: 1/1/0
   Serial2/0 Flags: A
   Ethernet0/0 Flags: F NS
     Pkts: 5254/1
 (192.168.16.1,239.1.1.1) Flags:
   SW Forwarding: 5255/10/28/2, Other: 0/0/0
   Serial4/0 Flags: A
   Ethernet0/0 Flags: F NS
     Pkts: 5254/1
```

PIM-SM Example Active Sources and Interested Receivers - SPT Threshold Set to Infinity

The following example shows how to verify multicast forwarding using the MFIB for PIM-SM in a network environment where there are active sources with interested receivers. This verification example is based on the topology shown in the figure.

Figure 34: PIM-SM Example Topology: Active Sources and Interested Receivers (SPT Threshold Set to Infinity)



For this verification example, the following conditions apply:

- Last-hop DRs R4 and R7 have the SPT threshold set to infinity (configured with the **ip pim spt-threshold infinity** command).

Note When the SPT threshold is set to infinity, multicast traffic is configured to stay on the shared tree. Last-hop DRs will not perform an SPT switchover.

- During the PIM-SM registration process between the first-hop DR and the RP, a PIM tunnel is used. First-hop DRs will have a PIM Encap Tunnel and the RP will have both a PIM Encap and Decap Tunnel. After the PIM-SM registration process completes, PIM tunnels will not be used for multicast forwarding. For more information, see the Verifying PIM Tunnel Interfaces for PIM-SM, on page 374 section.

R1 (First-Hop DR for Source A)

```
R1# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
```

```
         Uptime: 03:09:53, expires: 00:02:14
R1# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:37:29/stopped, RP 192.168.6.6, flags: SPF
  Incoming interface: Ethernet1/0, RPF nbr 192.168.123.2
  Outgoing interface list: Null
(192.168.1.2, 239.1.1.1), 00:37:29/00:02:53, flags: FT
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse, 00:22:46/00:03:19
R1# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 192.168.123.2 Flags: C
  Ethernet1/0 Flags: A
(192.168.1.2,239.1.1.1) RPF nbr: 0.0.0.0 Flags:
  Ethernet0/0 Flags: A
  Ethernet1/0 Flags: F NS
R1# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 0/0/0/0, Other: 13688/0/13688
   Ethernet1/0 Flags: A
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 13689/10/28/2, Other: 8826/0/8826
   Ethernet0/0 Flags: A
   Ethernet1/0 Flags: F NS
     Pkts: 13688/0
```

R2 (Router Along SPT for Source A and Shared Tree for Source B)

```
R2# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R2# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:23:28/00:02:44, RP 192.168.6.6, flags: S
  Incoming interface: Serial2/0, RPF nbr 192.168.26.6
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse, 00:23:28/00:02:44
(192.168.1.2, 239.1.1.1), 00:23:28/00:02:54, flags: T
  Incoming interface: Ethernet1/0, RPF nbr 192.168.123.1
  Outgoing interface list:
    Serial2/0, Forward/Sparse, 00:23:28/00:02:40
R2# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 192.168.26.6 Flags: C
  Ethernet1/0 Flags: F NS
  Serial2/0 Flags: A
(192.168.1.2,239.1.1.1) RPF nbr: 192.168.123.1 Flags:
  Ethernet1/0 Flags: A
  Serial2/0 Flags: F NS
R2# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 14084/10/28/2, Other: 0/0/0
   Serial2/0 Flags: A
   Ethernet1/0 Flags: F NS
     Pkts: 14084/0
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 14083/10/28/2, Other: 1/1/0
   Ethernet1/0 Flags: A
   Serial2/0 Flags: F NS
     Pkts: 14083/0
```

R3 (Router Along the Shared Tree)

```
R3# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R3# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:23:57/00:03:10, RP 192.168.6.6, flags: S
  Incoming interface: Ethernet1/0, RPF nbr 192.168.123.2
  Outgoing interface list:
    Serial3/0, Forward/Sparse, 00:23:57/00:03:10
R3# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 192.168.123.2 Flags: C
  Serial3/0 Flags: F NS
  Ethernet1/0 Flags: A
R3# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 28742/20/28/4, Other: 0/0/0
   Ethernet1/0 Flags: A
   Serial3/0 Flags: F NS
     Pkts: 28742/0
```

R4 (Last-Hop DR for Receiver B)

```
R4# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R4# show ip igmp groups 239.1.1.1
IGMP Connected Group Membership
Group Address    Interface       Uptime    Expires    Last Reporter   Group Accounted
239.1.1.1        Ethernet0/0     00:24:37  00:02:56   192.168.4.1
R4# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:24:35/00:02:35, RP 192.168.6.6, flags: SC
  Incoming interface: Serial1/0, RPF nbr 192.168.34.3
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 00:24:35/00:02:35
R4# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 192.168.34.3 Flags: C
  Ethernet0/0 Flags: F NS
  Serial1/0 Flags: A
R4# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 29517/20/28/4, Other: 0/0/0
   Serial1/0 Flags: A
   Ethernet0/0 Flags: F NS
     Pkts: 29517/0
```

R6 (RP and First-Hop DR for Source B)

```
R6# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)
```

```
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:10:53, expires: 00:02:06
R6# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:39:44/00:03:09, RP 192.168.6.6, flags: SF
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/0, Forward/Sparse, 00:24:57/00:03:09
    Serial2/0, Forward/Sparse, 00:25:01/00:03:09
(192.168.1.2, 239.1.1.1), 00:39:44/00:03:18, flags: T
  Incoming interface: Serial2/0, RPF nbr 192.168.26.2
  Outgoing interface list:
    Serial1/0, Forward/Sparse, 00:24:57/00:03:09
(192.168.16.1, 239.1.1.1), 00:39:44/00:02:35, flags: FT
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/0, Forward/Sparse, 00:24:57/00:03:09
    Serial2/0, Forward/Sparse, 00:25:01/00:03:09
R6# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 0.0.0.0 Flags: C
  Serial1/0 Flags: F NS
  Serial2/0 Flags: F NS
  Tunnel1 Flags: A
(192.168.1.2,239.1.1.1) RPF nbr: 192.168.26.2 Flags:
  Serial2/0 Flags: A
  Serial1/0 Flags: F NS
(192.168.16.1,239.1.1.1) RPF nbr: 0.0.0.0 Flags:
  Ethernet0/0 Flags: A
  Serial1/0 Flags: F NS
  Serial2/0 Flags: F NS
R6# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Tunnel1 Flags: A
   Serial2/0 Flags: F NS
     Pkts: 0/0
   Serial1/0 Flags: F NS
     Pkts: 0/0
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 14978/10/28/2, Other: 39/1/38
   Serial2/0 Flags: A
   Serial1/0 Flags: F NS
     Pkts: 14978/0
 (192.168.16.1,239.1.1.1) Flags:
   SW Forwarding: 15019/10/28/2, Other: 8827/0/8827
   Ethernet0/0 Flags: A
   Serial2/0 Flags: F NS
     Pkts: 15018/0
   Serial1/0 Flags: F NS
     Pkts: 14978/0
R6# show ip pim tunnel

Tunnel0
  Type  : PIM Encap
  RP    : 192.168.6.6*
  Source: 192.168.6.6
Tunnel1*
  Type  : PIM Decap
  RP    : 192.168.6.6*
  Source: -
```

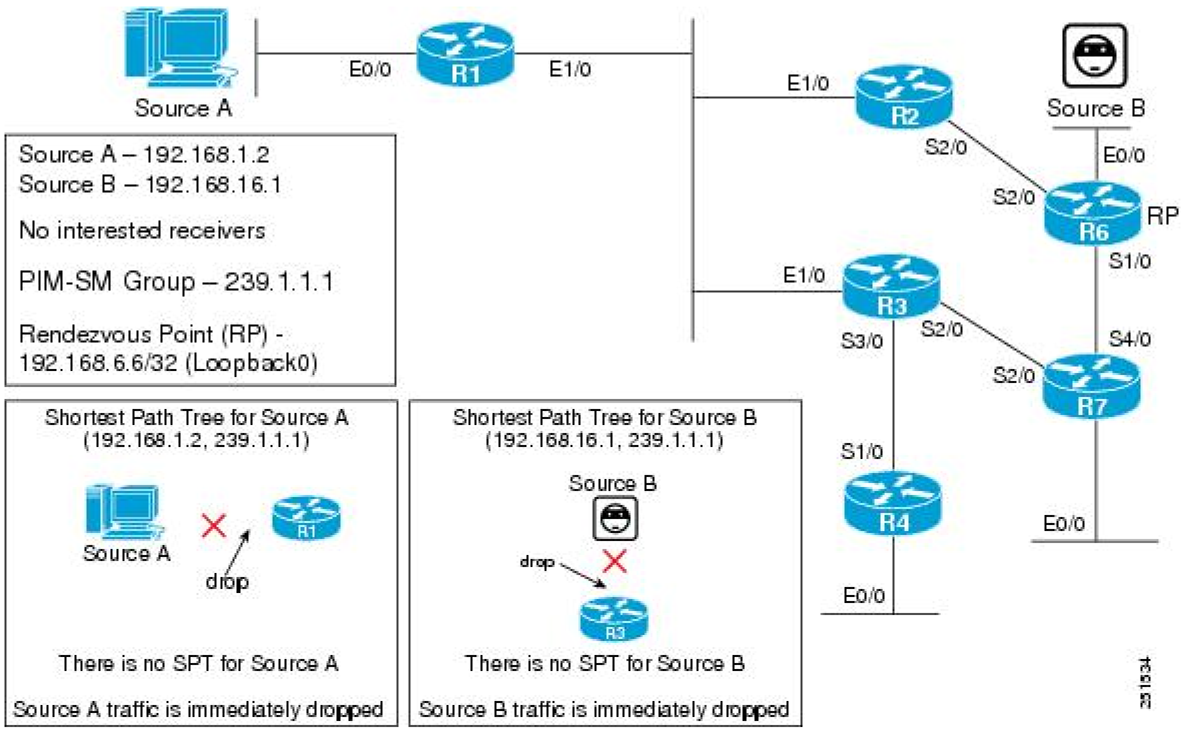
R7 (Last-Hop DR for Receiver A)

```
R7# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R7# show ip igmp groups 239.1.1.1
IGMP Connected Group Membership
Group Address    Interface       Uptime    Expires    Last Reporter   Group Accounted
239.1.1.1        Ethernet0/0     00:25:39  00:02:56   192.168.7.1
R7# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:25:37/00:02:58, RP 192.168.6.6, flags: SC
  Incoming interface: Serial4/0, RPF nbr 192.168.67.6
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 00:25:37/00:02:58
R7# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 192.168.67.6 Flags: C
  Ethernet0/0 Flags: F NS
  Serial4/0 Flags: A
R7# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 30756/20/28/4, Other: 0/0/0
   Serial4/0 Flags: A
   Ethernet0/0 Flags: F NS
     Pkts: 30756/0
```

PIM-SM Example Source Traffic Only with No Receivers

The following example shows how to verify multicast forwarding using the MFIB for PIM-SM in a network environment where sources are sending traffic without interested receivers. This verification example is based on the topology shown in the figure.

Figure 35: PIM-SM Example Topology: Source Traffic Only with No Receivers



In this verification example, the following conditions apply:

- Source A and Source B are sending traffic for multicast group 239.1.1.1 to first-hop DRs R1 and R6, respectively.
- When R1 and R6 receive the source traffic, they will then check their group-to-RP mapping cache for multicast group 239.1.1.1 to determine the RP. In this case, R6 is the RP.
- After determining the RP, R1 and R6 will then create state and send PIM registers for (Source A, 239.1.1.1) and (Source B, 239.1.1.1) toward the RP.
- Because there are no interested receivers, the RP will send a register stop to R1 and R6 (itself).
- R1 and R6 are the only routers that will have (S, G) state for 239.1.1.1.
- Routers that are not the RP or directly connected to an active source will not create state for (*, 239.1.1.1).

R1 (First-Hop DR for Source A)

```
R1# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R1# show ip mroute 239.1.1.1
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(*, 239.1.1.1), 00:02:06/stopped, RP 192.168.6.6, flags: SPF
  Incoming interface: Ethernet1/0, RPF nbr 192.168.123.2
  Outgoing interface list: Null
(192.168.1.2, 239.1.1.1), 00:02:06/00:02:53, flags: PFT
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list: Null
R1# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 192.168.123.2 Flags: C
  Ethernet1/0 Flags: A
(192.168.1.2,239.1.1.1) RPF nbr: 0.0.0.0 Flags:
  Ethernet0/0 Flags: A
R1# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Ethernet1/0 Flags: A
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 1/0/28/0, Other: 1267/0/1267
   Ethernet0/0 Flags: A
```

R6 (RP and First-Hop DR for Source B)

```
R6# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:10:53, expires: 00:02:06
R6# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:02:48/stopped, RP 192.168.6.6, flags: SPF
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null
(192.168.1.2, 239.1.1.1), 00:02:42/00:02:17, flags: P
  Incoming interface: Serial2/0, RPF nbr 192.168.26.2
  Outgoing interface list: Null
(192.168.16.1, 239.1.1.1), 00:02:48/00:02:11, flags: PFT
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list: Null
R6# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 0.0.0.0 Flags: C
  Tunnel1 Flags: A
(192.168.1.2,239.1.1.1) RPF nbr: 192.168.26.2 Flags:
  Serial2/0 Flags: NS
  Tunnel1 Flags: A
(192.168.16.1,239.1.1.1) RPF nbr: 0.0.0.0 Flags:
  Ethernet0/0 Flags: A
R6# show ip mfib 239.1.1.1
 (*,239.1.1.1) Flags: C
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Tunnel1 Flags: A
 (192.168.1.2,239.1.1.1) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Tunnel1 Flags: A
   Serial2/0 Flags: NS
 (192.168.16.1,239.1.1.1) Flags:
   SW Forwarding: 1/0/28/0, Other: 1688/0/1688
   Ethernet0/0 Flags: A
R6# show ip pim tunnel
Tunnel0
  Type  : PIM Encap
  RP    : 192.168.6.6*
  Source: 192.168.6.6
```

```
Tunnel1*
  Type   : PIM Decap
  RP     : 192.168.6.6*
  Source: -
```

In this scenario, R2, R3, R4, and R7 have no interested receivers; therefore, they are not on the multicast forwarding path and will not have multicast state. The output for the **show ip mroute**, **show ip mrib route**, and **show ip mfib route** commands would appear only on R2, R3, R4, and R7, as in this example (taken from R2):

R2 (Router Not Along the Multicast Forwarding Path)

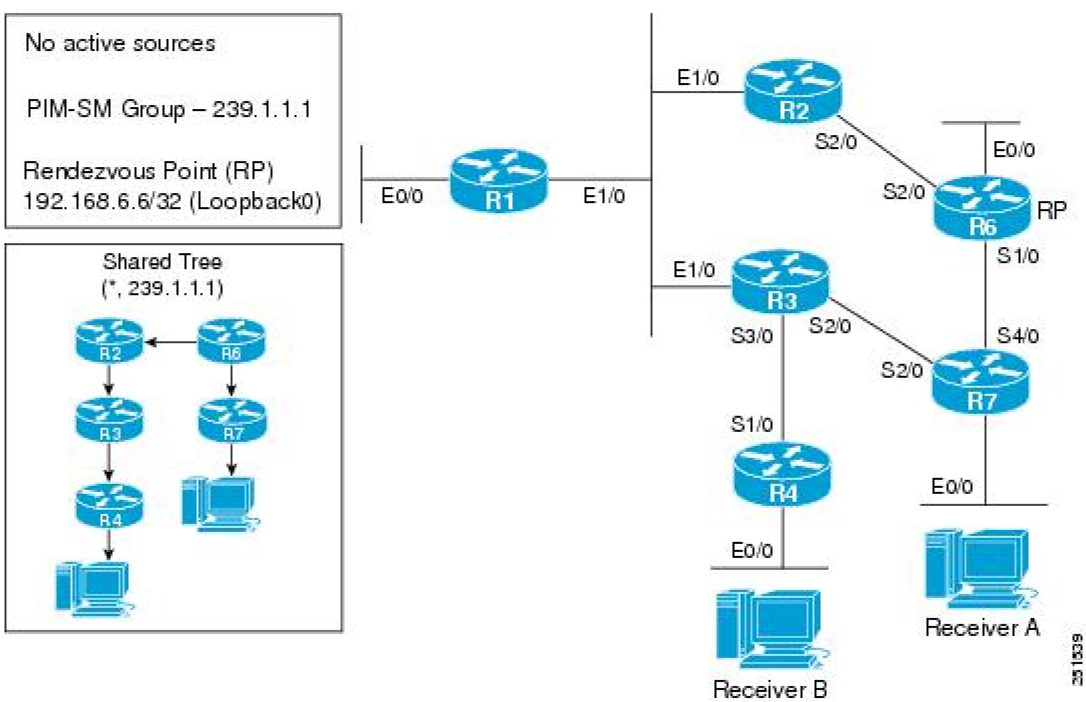
```
R2# show ip mroute 239.1.1.1
Group 239.1.1.1 not found
R2# show ip mrib route 239.1.1.1
No matching routes in MRIB route-DB
R2# show ip mfib 239.1.1.1
Group 239.1.1.1 not found
```

Note The output for the **show ip mroute**, **show ip mrib route**, and **show ip mfib** commands would be the same for R2, R3, R4, and R7 for this scenario.

PIM-SM Example Interested Receivers with No Active Sources

The following example shows how to verify multicast forwarding using the MFIB for PIM-SM in a network environment where there are interested receivers with no active sources. This verification example is based on the topology shown in the figure.

Figure 36: PIM-SM Example Topology: Interested Receivers with No Active Sources



For this verification example, the following conditions apply:

- Last-hop DRs R4 and R7 also have the SPT threshold set to infinity (configured with the **ip pim spt-threshold infinity** command).

Note When the SPT threshold is set to infinity, multicast traffic is configured to stay on the shared tree. Last-hop DRs will not perform an SPT switchover.

- Receiver A and Receiver B are sending IGMP joins to R7 and R4, respectively, for multicast group 239.1.1.1.
- When R4 and R7 receive the IGMP joins, they will then check their group-to-RP mapping cache for multicast group 239.1.1.1 to determine the RP.
- After determining the RP, R4 and R7 will then create state and send PIM joins for (*, 239.1.1.1) toward the RP.

Note The unicast routing table is used to build the shared tree entry for (*, 239.1.1.1). Shared tree entries are always rooted at the RP. In this scenario, the shared tree from R4 to R6 is through R3 and R2 because R3's best unicast route (determined by the underlying IGP) is R2. The shared tree for R7 is directly upstream to R6.

- Routers that are not along the shared tree will not create state for (*, 239.1.1.1).

R4 (Last-Hop DR)

```
R4# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R4# show ip igmp groups 239.1.1.1
IGMP Connected Group Membership
Group Address    Interface       Uptime    Expires    Last Reporter   Group Accounted
239.1.1.1        Ethernet0/0     00:03:07  00:02:56   192.168.4.1
R4# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:03:05/00:02:47, RP 192.168.6.6, flags: SJC
  Incoming interface: Serial1/0, RPF nbr 192.168.34.3
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 00:03:05/00:02:47
R4# show ip mrib route 239.1.1.1
(*,239.1.1.1) RPF nbr: 192.168.34.3 Flags: C
  Ethernet0/0 Flags: F NS
  Serial1/0 Flags: A NS
R4# show ip mfib 239.1.1.1
(*,239.1.1.1) Flags: C
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial1/0 Flags: A NS
   Ethernet0/0 Flags: F NS
     Pkts: 0/0
```

R3 (Router Along the Shared Tree)

```
R4# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R3# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:03:40/00:02:47, RP 192.168.6.6, flags: S
  Incoming interface: Ethernet1/0, RPF nbr 192.168.123.2
  Outgoing interface list:
    Serial3/0, Forward/Sparse, 00:03:40/00:02:47
R3# show ip mrib route 239.1.1.1
 (*,239.1.1.1) RPF nbr: 192.168.123.2 Flags: C
  Serial3/0 Flags: F NS
  Ethernet1/0 Flags: A
R3# show ip mfib 239.1.1.1
(*,239.1.1.1) Flags: C
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Ethernet1/0 Flags: A
   Serial3/0 Flags: F NS
     Pkts: 0/0
```

R2 (Router Along the Shared Tree)

```
R2# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R2# show ip mroute 239.1.1.1
 (*, 239.1.1.1), 00:04:05/00:03:20, RP 192.168.6.6, flags: S
  Incoming interface: Serial2/0, RPF nbr 192.168.26.6
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse, 00:04:05/00:03:20
R2# show ip mrib route 239.1.1.1
 (*,239.1.1.1) RPF nbr: 192.168.26.6 Flags: C
  Ethernet1/0 Flags: F NS
  Serial2/0 Flags: A
R2# show ip mfib 239.1.1.1
(*,239.1.1.1) Flags: C
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial2/0 Flags: A
   Ethernet1/0 Flags: F NS
     Pkts: 0/0
```

R7 (Last-Hop DR)

```
R7# show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:09:53, expires: 00:02:14
R7# show ip igmp groups 239.1.1.1
```

```
IGMP Connected Group Membership
Group Address    Interface       Uptime    Expires    Last Reporter   Group Accounted
239.1.1.1        Ethernet0/0     00:04:33  00:02:56   192.168.7.1
R7# show ip mroute 239.1.1.1
 (*, 239.1.1.1), 00:04:31/00:02:36, RP 192.168.6.6, flags: SJC
  Incoming interface: Serial4/0, RPF nbr 192.168.67.6
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 00:04:31/00:02:36
R7# show ip mrib route 239.1.1.1
 (*,239.1.1.1) RPF nbr: 192.168.67.6 Flags: C
  Ethernet0/0 Flags: F NS
  Serial4/0 Flags: A NS
R7# show ip mfib 239.1.1.1
(*,239.1.1.1) Flags: C
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial4/0 Flags: A NS
   Ethernet0/0 Flags: F NS
     Pkts: 0/0
```

R6 (RP)

```
R6# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)
Group(s) 224.0.0.0/4
  RP 192.168.6.6 (?), v2v1
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 03:10:53, expires: 00:02:06
R6# show ip mroute 239.1.1.1
(*, 239.1.1.1), 00:05:01/00:03:27, RP 192.168.6.6, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/0, Forward/Sparse, 00:04:57/00:03:27
    Serial2/0, Forward/Sparse, 00:05:01/00:03:23
R6# show ip mrib route 239.1.1.1
 (*,239.1.1.1) RPF nbr: 0.0.0.0 Flags: C
  Serial1/0 Flags: F NS
  Serial2/0 Flags: F NS
  Tunnel1 Flags: A
R6# show ip mfib 239.1.1.1
(*,239.1.1.1) Flags: C
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Tunnel1 Flags: A
   Serial2/0 Flags: F NS
     Pkts: 0/0
   Serial1/0 Flags: F NS
     Pkts: 0/0
R6# show ip pim tunnel
Tunnel0
  Type  : PIM Encap
  RP    : 192.168.6.6*
  Source: 192.168.6.6
Tunnel1*
  Type  : PIM Decap
  RP    : 192.168.6.6*
  Source: -
```

R1 (Router Not Along the Multicast Forwarding Path)

```
R1# show ip mroute 239.1.1.1
```

```
Group 239.1.1.1 not found
R1# show ip mrib route 239.1.1.1
No matching routes in MRIB route-DB
R1# show ip mfib 239.1.1.1
Group 239.1.1.1 not found
```

Note R1 does not have any state for 239.1.1.1 because it does not have an interested receiver, is not along the shared tree path, and does not have a directly connected source.

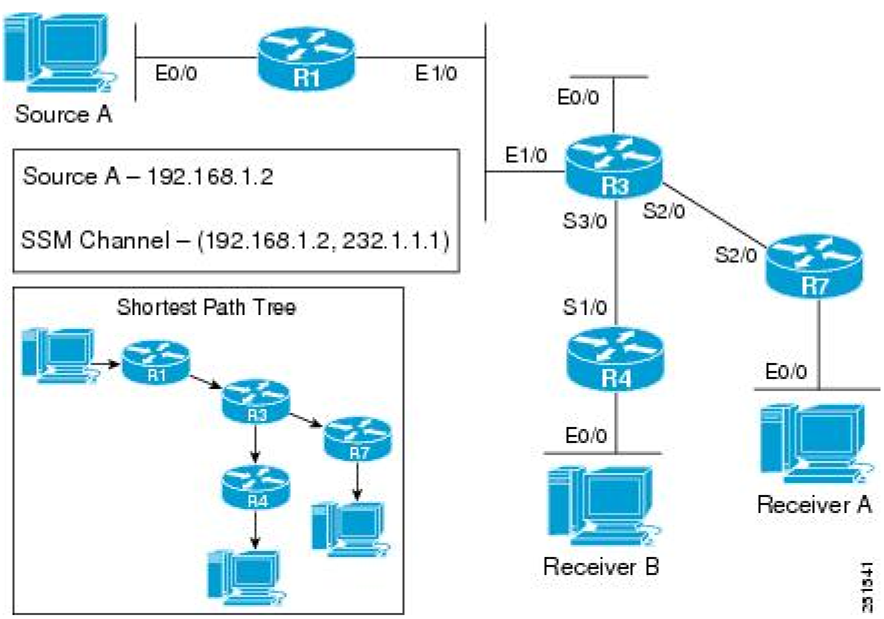
Examples Verifying IPv4 Multicast Forwarding Using the MFIB for PIM-SSM

Note The examples in this section were created in a test environment to provide a conceptual view of the multicast environment. The IP addresses, interfaces, and other values are shown as examples only. They do not show real-world deployment values.

PIM-SSM Example Interested Receivers With or Without Active Sources

The following example shows how to verify multicast forwarding using the MFIB for PIM-SSM in a network environment where there are interested receivers with or without active sources. This verification example is based on the topology shown in the figure.

Figure 37: PIM-SSM Example Topology: Interested Receivers With or Without Active Sources



For this verification example, the following conditions apply:

- All routers in the network have been configured to run PIM-SSM and have the **ip pim ssm default** command configured globally.
- Source A is sending multicast packets to SSM group 232.1.1.1.

- Receiver A and Receiver B are interested in receiving multicast from Source A, (192.168.1.2, 232.1.1.1).
- Receiver A and Receiver B are using IGMPv3.

R1 (First-Hop DR for Source A)

```
R1# show ip mroute 232.1.1.1
(192.168.1.2, 232.1.1.1), 00:07:18/00:03:02, flags: sT
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse, 00:07:18/00:03:02
R1# show ip mrib route 232.1.1.1
(192.168.1.2,232.1.1.1) RPF nbr: 0.0.0.0 Flags:
  Ethernet1/0 Flags: F NS
  Ethernet0/0 Flags: A
R1# show ip mfib 232.1.1.1
 (192.168.1.2,232.1.1.1) Flags:
   SW Forwarding: 3039/10/28/2, Other: 0/0/0
   Ethernet0/0 Flags: A
   Ethernet1/0 Flags: F NS
     Pkts: 3039/0
```

R3 (Router Along the SPT)

```
R3# show ip mroute 232.1.1.1
(192.168.1.2, 232.1.1.1), 00:08:00/00:03:13, flags: sT
  Incoming interface: Ethernet1/0, RPF nbr 192.168.123.1
  Outgoing interface list:
    Serial3/0, Forward/Sparse, 00:08:00/00:03:13
    Serial2/0, Forward/Sparse, 00:08:00/00:02:59
R3# show ip mrib route 232.1.1.1
(192.168.1.2,232.1.1.1) RPF nbr: 192.168.123.1 Flags:
  Serial3/0 Flags: F NS
  Serial2/0 Flags: F NS
  Ethernet1/0 Flags: A
R3# show ip mfib 232.1.1.1
 (192.168.1.2,232.1.1.1) Flags:
   SW Forwarding: 3514/10/28/2, Other: 0/0/0
   Ethernet1/0 Flags: A
   Serial3/0 Flags: F NS
     Pkts: 3514/0
   Serial2/0 Flags: F NS
     Pkts: 3514/0
```

R4 (Last-Hop DR for Receiver B)

```
R4# show ip igmp groups 232.1.1.1
IGMP Connected Group Membership
Group Address    Interface          Uptime    Expires   Last Reporter   Group Accounted
232.1.1.1        Ethernet0/0        00:12:46  stopped   192.168.4.1
R4# show ip mroute 232.1.1.1
(192.168.1.2, 232.1.1.1), 00:08:42/stopped, flags: sTI
  Incoming interface: Serial1/0, RPF nbr 192.168.34.3
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 00:08:42/00:00:17
R4# show ip mrib route 232.1.1.1
(192.168.1.2,232.1.1.1) RPF nbr: 192.168.34.3 Flags:
  Serial1/0 Flags: A
  Ethernet0/0 Flags: F NS
```

```
R4# show ip mfib 232.1.1.1
 (192.168.1.2,232.1.1.1) Flags:
   SW Forwarding: 3786/10/28/2, Other: 0/0/0
   Serial1/0 Flags: A
   Ethernet0/0 Flags: F NS
     Pkts: 3786/0
```

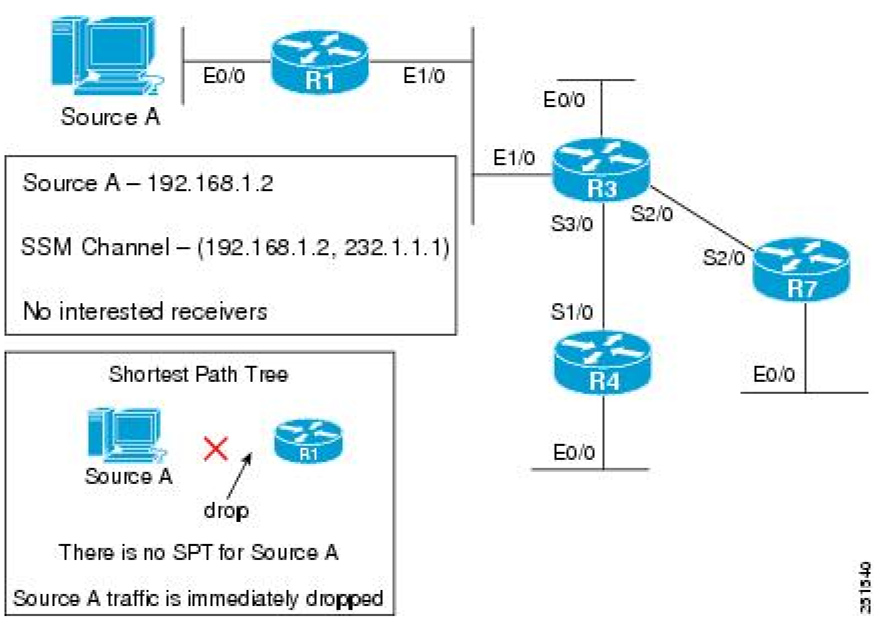
R7 (Last-Hop DR for Receiver A)

```
R7# show ip igmp groups 232.1.1.1
IGMP Connected Group Membership
Group Address    Interface          Uptime    Expires   Last Reporter   Group Accounted
232.1.1.1        Ethernet0/0        00:12:24  stopped   192.168.7.1
R7# show ip mroute 232.1.1.1
(192.168.1.2, 232.1.1.1), 00:09:37/stopped, flags: sTI
  Incoming interface: Serial2/0, RPF nbr 192.168.37.3
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 00:09:37/00:02:22
R7# show ip mrib route 232.1.1.1
(192.168.1.2,232.1.1.1) RPF nbr: 192.168.37.3 Flags:
  Serial2/0 Flags: A
  Ethernet0/0 Flags: F NS
R7# show ip mfib 232.1.1.1
 (192.168.1.2,232.1.1.1) Flags:
   SW Forwarding: 4182/10/28/2, Other: 0/0/0
   Serial2/0 Flags: A
   Ethernet0/0 Flags: F NS
     Pkts: 4182/0
```

PIM-SSM Example Source Traffic Only with No Active Receivers

The following example shows how to verify multicast forwarding using the MFIB for PIM-SSM in a network environment where there is an active source with no interested receivers. This verification example is based on the topology shown in the figure.

Figure 38: PIM-SSM Example Topology: Source Traffic Only with No Active Receivers



For this verification example, the following conditions apply:

- All routers in the network have been configured to run PIM-SSM and have the **ip pim ssm default** command configured globally.
- Source A is sending multicast packets to SSM group 232.1.1.1.
- Source B is not actively sending.
- There are no interested receivers in the network.

Routers that support the MFIB will not create state for SSM multicast groups until a join has been requested by an interested receiver, which means that any routers with active sources sending to an SSM group will not have multicast state. Because there are no interested receivers in this network, none of the routers will create state for (192.168.1.2, 232.1.1.1).

The following is output from the **show ip mroute**, **show ip mrib route**, and **show ip mfib** commands taken from R1:

R1

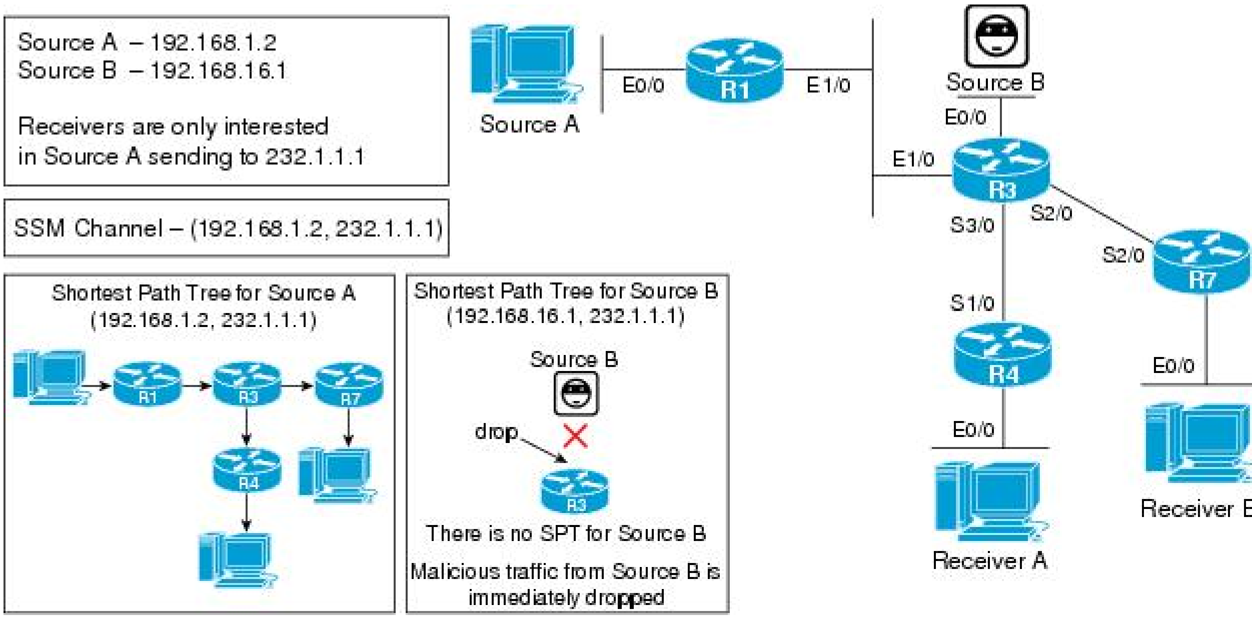
```
R1# show ip mroute 239.1.1.1
Group 239.1.1.1 not found
R1# show ip mrib route 239.1.1.1
No matching routes in MRIB route-DB
R1# show ip mfib 239.1.1.1
Group 239.1.1.1 not found
```

Note Because there are no interested receivers in this network, the output from the **show ip mroute**, **show ip mrib route**, and **show ip mfib** commands would be the same on R3, R4, and R7 in this example scenario.

PIM-SSM Example Unwanted Sources in the SSM Network

The following example shows how to verify multicast forwarding using the MFIB for PIM-SSM in a network environment where there is an unwanted source. This verification example is based on the topology shown in the figure.

Figure 39: PIM-SSM Example Topology: Unwanted Sources in the SSM Network



For this verification example, the following conditions apply:

- All routers in the network have been configured to run PIM-SSM and have the **ip pim ssm default** command configured globally.
- Receiver A and Receiver B are only interested in receiving multicast from Source A, (192.168.1.2, 232.1.1.1).
- Unwanted source, Source B, is sending traffic to 232.1.1.1.

Note Even though Source B is directly connected to R3, R3 will not create state for 232.1.1.1. Multicast traffic from Source B sending to SSM group 232.1.1.1, thus, will be immediately dropped by the router.

R3 (First-Hop DR for Unwanted Source B)

```
R3# show ip mroute 232.1.1.1 192.168.3.1
R3# show ip mrib route 232.1.1.1 192.168.3.1
No matching routes in MRIB route-DB
R3# show ip mfib 232.1.1.1 192.168.3.1
(192.168.3.1,232.1.1.1) entry not found
```

Note Likewise, R1, R4, and R7 will also have no multicast state for (192.168.3.1, 232.1.1.1) and any directly connected sources sending to 232.1.1.1 will be dropped.

Examples Verifying IPv4 Multicast Forwarding Using the MFIB for Bidir-PIM Networks

This section contains the following examples for verifying multicast forwarding using the MFIB for bidir-PIM networks:

Note The examples in this section were created in a test environment to provide a conceptual view of the multicast environment. The IP addresses, interfaces, and other values are shown as examples only. They do not show real-world deployment values.

Bidir-PIM Example Active Sources with Interested Receivers

The following example shows how to verify multicast forwarding using the MFIB for bidir-PIM in a network environment where there are active sources and interested receivers. This verification example is based on the topology shown in the figures.

Figure 40: Bidir-PIM Example Topology: Active Sources with Interested Receivers

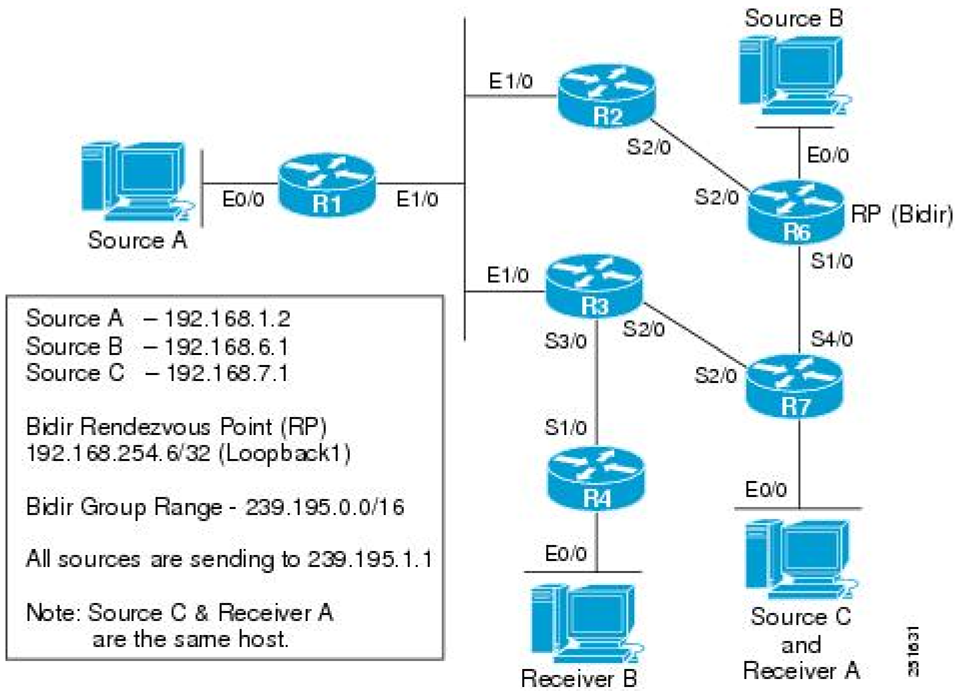
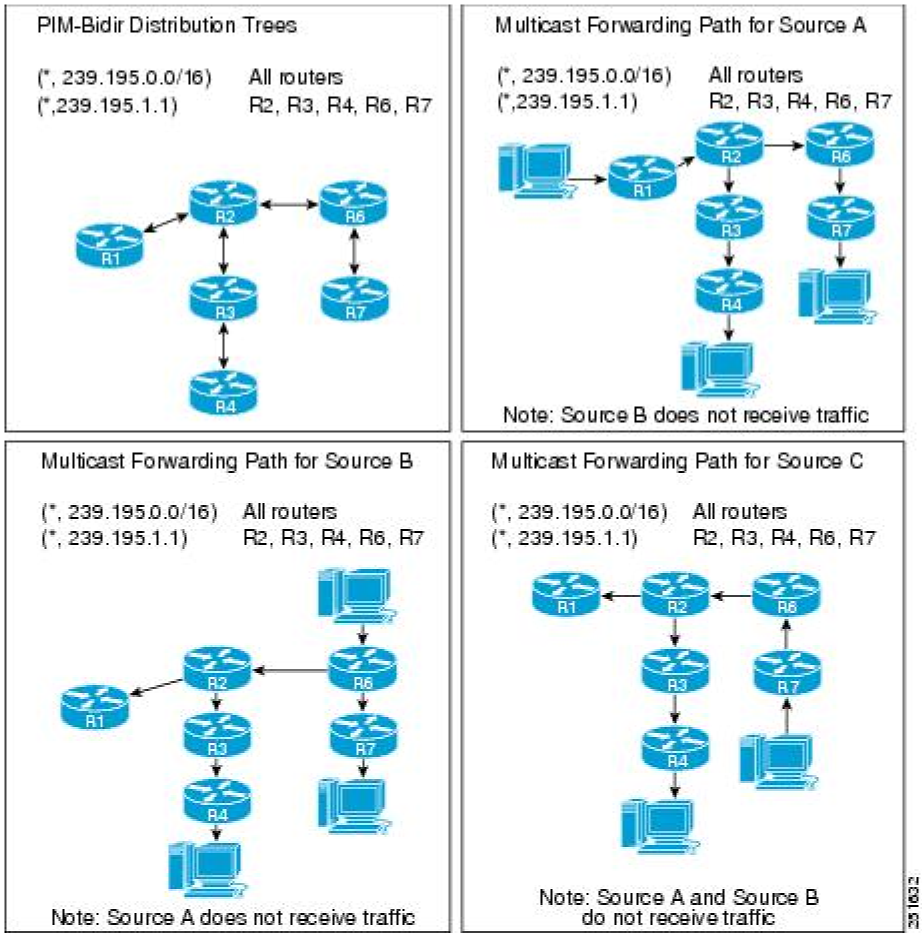


Figure 41: Bidir-PIM Distribution Trees and Multicast Forwarding Paths for the Active Sources with No Interested Receivers Example



For this verification example, the following conditions apply:

- Entries for (*, 239.195.0.0/16) are created by the control plane based on the PIM group-to-RP mappings on all routers.
- Entries for (*, 239.195.1.1) will only be created when IGMP joins are initiated by interested receivers joining this group. As a result, all routers along the shared tree between the RP and the last-hop DRs that have interested receivers will have state for (*, 239.195.1.1).

Note R1 will not have state for (*, 239.195.1.1) because it is not between the RP and the last-hop DRs.

- If both (*,239.195.0.0/26) and (*,239.195.1.1) entries are present in a router, the more specific entry, (*,239.195.1.1) will be used for forwarding.
- All source traffic for this scenario will go to the RP and then out the appropriate interfaces where there are interested receivers.
- Source traffic received by the RP is never sent back out the same interface it was received on.

- In general, multicast packet forwarding can be verified by observing the “SW Forwarding” counter in the **show ip mfib** output for the most specific entry available in the MFIB. If multicast is being forwarded, this counter will increment.

R1 (First-Hop DR for Source A)

```
R1# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d02h, expires: 00:02:09
R1# show ip pim interface df
* implies this system is the DF
Interface                RP                DF Winner        Metric     Uptime
Ethernet0/0              192.168.254.6    *192.168.1.1       75          1d02h
Ethernet1/0              192.168.254.6     192.168.123.2     65          1d02h
R1# show ip mroute 239.195.1.1
Group 239.195.1.1 not found
R1# show ip mrib route 239.195.1.1
No matching routes in MRIB route-DB
R1# show ip mfib 239.195.1.1
Group 239.195.1.1 not found
R1# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d02h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Ethernet1/0, RPF nbr: 192.168.123.2
  Incoming interface list:
    Ethernet0/0, Accepting/Sparse
    Ethernet1/0, Accepting/Sparse
R1# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.123.2 Flags:
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Ethernet1/0 Flags: A F
R1# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 4677/10/28/2, Other: 9355/0/9355
   Ethernet1/0 Flags: A F
     Pkts: 4677/0
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

R2 (Router Along the Multicast Forwarding Path)

```
R2# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d02h, expires: 00:02:45
R2# show ip pim interface df
* implies this system is the DF
Interface                RP                DF Winner        Metric     Uptime
Ethernet1/0              192.168.254.6    *192.168.123.2     65          1d02h
Serial2/0                192.168.254.6     192.168.26.6      0           1d02h
R2# show ip mroute 239.195.1.1
(*, 239.195.1.1), 02:13:50/00:02:36, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial2/0, RPF nbr 192.168.26.6
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse, 02:13:50/00:02:36
```

```
    Serial2/0, Bidir-Upstream/Sparse, 02:13:50/00:00:00
R2# show ip mrib route 239.195.1.1
(*,239.195.1.1) RPF nbr: 192.168.26.6 Flags: IA
  Ethernet1/0 Flags: F
  Serial2/0 Flags: F
R2# show ip mfib 239.195.1.1
 (*,239.195.1.1) Flags: IA
   SW Forwarding: 14693/30/28/6, Other: 0/0/0
   Serial2/0 Flags: F
     Pkts: 4897/0
   Ethernet1/0 Flags: F
     Pkts: 9796/0
R2# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d02h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial2/0, RPF nbr: 192.168.26.6
  Incoming interface list:
    Ethernet1/0, Accepting/Sparse
    Serial2/0, Accepting/Sparse
R2# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.26.6 Flags:
  Ethernet1/0 Flags: A
  Null0 Flags: A
  Serial2/0 Flags: A F
R2# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial2/0 Flags: A F
     Pkts: 0/0
   Ethernet1/0 Flags: A
   Null0 Flags: A
```

R3 (Router Along the Multicast Forwarding Path)

```
R3# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d02h, expires: 00:02:25
R3# show ip pim interface df
* implies this system is the DF
Interface                RP               DF Winner        Metric     Uptime
Ethernet0/0              192.168.254.6    *192.168.3.3      75         1d02h
Ethernet1/0              192.168.254.6     192.168.123.2    65         1d02h
Serial2/0                192.168.254.6     192.168.37.7     65         1d02h
Serial3/0                192.168.254.6    *192.168.34.3     75         1d02h
R3# show ip mroute 239.195.1.1
(*, 239.195.1.1), 02:14:09/00:03:08, RP 192.168.254.6, flags: B
  Bidir-Upstream: Ethernet1/0, RPF nbr 192.168.123.2
  Outgoing interface list:
    Serial3/0, Forward/Sparse, 02:14:09/00:03:08
    Ethernet1/0, Bidir-Upstream/Sparse, 02:14:09/00:00:00
R3# show ip mrib route 239.195.1.1
(*,239.195.1.1) RPF nbr: 192.168.123.2 Flags: IA
  Serial3/0 Flags: F
  Ethernet1/0 Flags: F
R3# show ip mfib 239.195.1.1
 (*,239.195.1.1) Flags: IA
   SW Forwarding: 15263/30/28/6, Other: 0/0/0
   Serial3/0 Flags: F
     Pkts: 15263/0
   Ethernet1/0 Flags: F
     Pkts: 0/0
```

```
R3# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d02h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Ethernet1/0, RPF nbr: 192.168.123.2
  Incoming interface list:
    Serial3/0, Accepting/Sparse
    Ethernet0/0, Accepting/Sparse
    Ethernet1/0, Accepting/Sparse
R3# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.123.2 Flags:
  Serial3/0 Flags: A
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Ethernet1/0 Flags: A F
R3# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial3/0 Flags: A
   Ethernet1/0 Flags: A F
     Pkts: 0/0
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

R4 (Last-Hop DR for Receiver B)

```
R4# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d02h, expires: 00:02:10
R4# show ip pim interface df
* implies this system is the DF
Interface                RP                DF Winner        Metric     Uptime
Ethernet0/0              192.168.254.6    *192.168.4.4      139         1d02h
Serial1/0                192.168.254.6     192.168.34.3     75          1d02h
R4# show ip igmp groups 239.195.1.1

IGMP Connected Group Membership
Group Address    Interface               Uptime    Expires   Last Reporter   Group Accounted
239.195.1.1      Ethernet0/0               02:14:25  00:02:25  192.168.4.1
R4# show ip mroute 239.195.1.1
(*, 239.195.1.1), 02:14:25/00:02:25, RP 192.168.254.6, flags: BC
  Bidir-Upstream: Serial1/0, RPF nbr 192.168.34.3
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 02:14:25/00:02:25
    Serial1/0, Bidir-Upstream/Sparse, 02:14:25/00:00:00
R4# show ip mrib route 239.195.1.1
(*,239.195.1.1) RPF nbr: 192.168.34.3 Flags: IA
  Ethernet0/0 Flags: F
  Serial1/0 Flags: F
R4# show ip mfib 239.195.1.1
 (*,239.195.1.1) Flags: IA
   SW Forwarding: 15729/30/28/6, Other: 0/0/0
   Serial1/0 Flags: F
     Pkts: 0/0
   Ethernet0/0 Flags: F
     Pkts: 15729/0
R4# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d02h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial1/0, RPF nbr: 192.168.34.3
  Incoming interface list:
    Ethernet0/0, Accepting/Sparse
    Serial1/0, Accepting/Sparse
```

```
R4# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.34.3 Flags:
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Serial1/0 Flags: A F
R4# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial1/0 Flags: A F
     Pkts: 0/0
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

R6 (RP and First-Hop DR for Source B)

```
R6# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.254.6 (?), elected via Auto-RP
         Uptime: 1d02h, expires: 00:02:55
R6# show ip pim interface df
* implies this system is the DF
Interface                RP               DF Winner        Metric     Uptime
Loopback0                192.168.254.6    *192.168.6.6      0          1d02h
Loopback1                192.168.254.6    *192.168.254.6    0          1d02h
Ethernet0/0              192.168.254.6    *192.168.16.6     0          1d02h
Serial1/0                192.168.254.6    *192.168.67.6     0          1d02h
Serial2/0                192.168.254.6    *192.168.26.6     0          1d02h
R6# show ip mroute 239.195.1.1
(*, 239.195.1.1), 02:14:43/00:02:49, RP 192.168.254.6, flags: B
  Bidir-Upstream: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/0, Forward/Sparse, 02:14:35/00:02:49
    Serial2/0, Forward/Sparse, 02:14:43/00:02:41
R6# show ip mrib route 239.195.1.1
(*,239.195.1.1) RPF nbr: 0.0.0.0 Flags: IA
  Serial1/0 Flags: F
  Serial2/0 Flags: F
R6# show ip mfib 239.195.1.1
 (*,239.195.1.1) Flags: IA
   SW Forwarding: 16269/30/28/6, Other: 0/0/0
   Serial2/0 Flags: F
     Pkts: 10846/0
   Serial1/0 Flags: F
     Pkts: 10846/0
R6# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d02h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Loopback1, RPF nbr: 192.168.254.6
  Incoming interface list:
    Serial2/0, Accepting/Sparse
    Serial1/0, Accepting/Sparse
    Ethernet0/0, Accepting/Sparse
    Loopback0, Accepting/Sparse
    Loopback1, Accepting/Sparse
R6# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.254.6 Flags:
  Serial2/0 Flags: A
  Serial1/0 Flags: A
  Ethernet0/0 Flags: A
  Loopback0 Flags: A
```

```
  Null0 Flags: A
  Loopback1 Flags: A F
R6# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Loopback1 Flags: A F
     Pkts: 0/0
   Loopback0 Flags: A
   Serial2/0 Flags: A
   Serial1/0 Flags: A
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

R7 (First-Hop DR for Source C and Last-Hop DR for Receiver A)

```
R7# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d02h, expires: 00:02:35
R7# show ip pim interface df
* implies this system is the DF
Interface                RP               DF Winner        Metric     Uptime
Ethernet0/0              192.168.254.6    *192.168.7.7      65         1d02h
Serial2/0                192.168.254.6    *192.168.37.7     65         1d02h
Serial4/0                192.168.254.6     192.168.67.6     0          1d02h
R7# show ip igmp groups 239.195.1.1
IGMP Connected Group Membership
Group Address    Interface              Uptime    Expires   Last Reporter   Group Accounted
239.195.1.1      Ethernet0/0              02:14:51  00:02:44  192.168.7.1
R7# show ip mroute 239.195.1.1
(*, 239.195.1.1), 02:14:51/00:02:43, RP 192.168.254.6, flags: BC
  Bidir-Upstream: Serial4/0, RPF nbr 192.168.67.6
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 02:14:51/00:02:43
    Serial4/0, Bidir-Upstream/Sparse, 02:14:51/00:00:00
R7# show ip mrib route 239.195.1.1
(*,239.195.1.1) RPF nbr: 192.168.67.6 Flags: IA
  Ethernet0/0 Flags: F
  Serial4/0 Flags: F
R7# show ip mfib 239.195.1.1
 (*,239.195.1.1) Flags: IA
   SW Forwarding: 16747/30/28/6, Other: 0/0/0
   Serial4/0 Flags: F
     Pkts: 5582/0
   Ethernet0/0 Flags: F
     Pkts: 11165/0
R7# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d02h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial4/0, RPF nbr: 192.168.67.6
  Incoming interface list:
    Serial2/0, Accepting/Sparse
    Ethernet0/0, Accepting/Sparse
    Serial4/0, Accepting/Sparse
R7# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.67.6 Flags:
  Serial2/0 Flags: A
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Serial4/0 Flags: A F
R7# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
```

```
SW Forwarding: 0/0/0/0, Other: 0/0/0
Serial4/0 Flags: A F
  Pkts: 0/0
Serial2/0 Flags: A
Ethernet0/0 Flags: A
Null0 Flags: A
```

Bidir-PIM Example Active Sources with No Interested Receivers

The following example shows how to verify multicast forwarding using the MFIB for bidir-PIM in a network environment where there are active sources with no interested receivers. This verification example is based on the topology shown in the figures.

Figure 42: Bidir-PIM Example Topology: Active Sources with No Interested Receivers

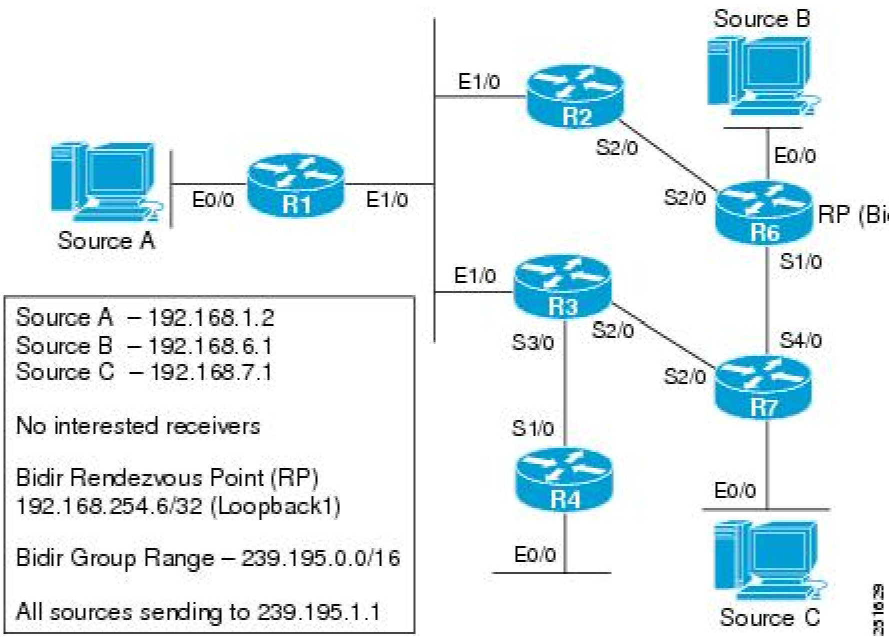
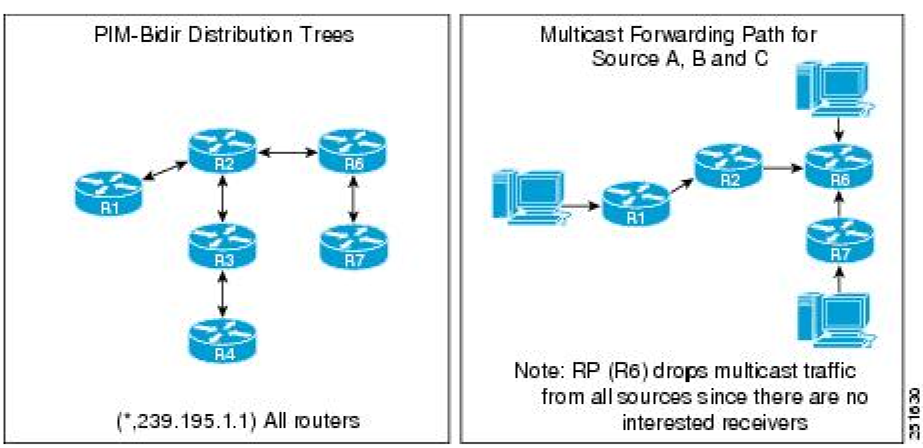


Figure 43: Bidir-PIM Distribution Trees and Multicast Forwarding Path for Active Sources with No Interested Receivers Example



For this verification example, the following conditions apply:

- Entries for (*, 239.195.0.0/16) are created by the control plane based on the PIM group-to-RP mappings on all routers.
- Because there are no interested receivers, (*, 239.195.0.0/16) will be the only state in the network on all routers.
- All source traffic for this example will go to the RP and then be dropped because there are no interested receivers in the network. In addition, source traffic received by the RP is never sent back out the same interface it was received on.
- In general, multicast packet forwarding can be verified by observing the "SW Forwarding" counter in the **show ip mfib** output for the most specific entry available in the MFIB. If multicast is being forwarded, this counter will increment.
- In this scenario, all traffic stops at the RP because there are no interested receivers. In addition, traffic received by the RP will be forwarded only to the bidir-PIM RP interface to be dropped.

R1 (First-Hop DR for Source A)

```
R1# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d03h, expires: 00:02:43
R1# show ip pim interface df
* implies this system is the DF
Interface                RP                DF Winner        Metric     Uptime
Ethernet0/0              192.168.254.6    *192.168.1.1      75          1d03h
Ethernet1/0              192.168.254.6     192.168.123.2    65          1d03h
R1# show ip mroute 239.195.1.1
Group 239.195.1.1 not found
R1# show ip mrib route 239.195.1.1
No matching routes in MRIB route-DB
R1# show ip mfib 239.195.1.1
Group 239.195.1.1 not found
R1# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d03h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Ethernet1/0, RPF nbr: 192.168.123.2
  Incoming interface list:
    Ethernet0/0, Accepting/Sparse
    Ethernet1/0, Accepting/Sparse
R1# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.123.2 Flags:
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Ethernet1/0 Flags: A F
R1# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 34754/10/28/2, Other: 58228/0/58228
   Ethernet1/0 Flags: A F
     Pkts: 34754/0
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

R2

```
R2# show ip pim rp mapping
```

```
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d03h, expires: 00:02:28
R2# show ip pim interface df
* implies this system is the DF
Interface                RP                DF Winner        Metric     Uptime
Ethernet1/0              192.168.254.6    *192.168.123.2    65          1d03h
Serial2/0                192.168.254.6     192.168.26.6     0           1d03h
R2# show ip mroute 239.195.1.1
Group 239.195.1.1 not found
R2# show ip mrib route 239.195.1.1
No matching routes in MRIB route-DB
R2# show ip mfib 239.195.1.1
Group 239.195.1.1 not found
R2# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d03h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial2/0, RPF nbr: 192.168.26.6
  Incoming interface list:
    Ethernet1/0, Accepting/Sparse
    Serial2/0, Accepting/Sparse
R2# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.26.6 Flags:
  Ethernet1/0 Flags: A
  Null0 Flags: A
  Serial2/0 Flags: A F
R2# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 4211/10/28/2, Other: 0/0/0
   Serial2/0 Flags: A F
     Pkts: 4211/0
   Ethernet1/0 Flags: A
   Null0 Flags: A
```

R3

```
R3# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d03h, expires: 00:02:09
R3# show ip pim interface df
* implies this system is the DF
Interface                RP                DF Winner        Metric     Uptime
Ethernet0/0              192.168.254.6    *192.168.3.3      75          1d03h
Ethernet1/0              192.168.254.6     192.168.123.2    65          1d03h
Serial2/0                192.168.254.6     192.168.37.7     65          1d03h
Serial3/0                192.168.254.6    *192.168.34.3     75          1d03h
R3# show ip igmp groups 239.195.1.1

IGMP Connected Group Membership
Group Address    Interface              Uptime    Expires   Last Reporter   Group Accounted
R3# show ip mroute 239.195.1.1
Group 239.195.1.1 not found
R3# show ip mrib route 239.195.1.1
No matching routes in MRIB route-DB
R3# show ip mfib 239.195.1.1
Group 239.195.1.1 not found
R3# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d03h/-, RP 192.168.254.6, flags: B
```

```
  Bidir-Upstream: Ethernet1/0, RPF nbr: 192.168.123.2
  Incoming interface list:
    Serial3/0, Accepting/Sparse
    Ethernet0/0, Accepting/Sparse
    Ethernet1/0, Accepting/Sparse
R3# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.123.2 Flags:
  Serial3/0 Flags: A
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Ethernet1/0 Flags: A F
R3# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 3935/0/3935
   Serial3/0 Flags: A
   Ethernet1/0 Flags: A F
     Pkts: 0/0
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

R4

```
R4# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d03h, expires: 00:02:54
R4# show ip pim interface df
* implies this system is the DF
Interface                RP                DF Winner        Metric     Uptime
Ethernet0/0              192.168.254.6    *192.168.4.4       139        1d03h
Serial1/0                192.168.254.6     192.168.34.3      75         1d03h
R4# show ip igmp groups 239.195.1.1

IGMP Connected Group Membership
Group Address    Interface              Uptime    Expires   Last Reporter   Group Accounted
R4# show ip mroute 239.195.1.1
Group 239.195.1.1 not found
R4# show ip mrib route 239.195.1.1
No matching routes in MRIB route-DB
R4# show ip mfib 239.195.1.1
Group 239.195.1.1 not found
R4# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d03h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial1/0, RPF nbr: 192.168.34.3
  Incoming interface list:
    Ethernet0/0, Accepting/Sparse
    Serial1/0, Accepting/Sparse
R4# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.34.3 Flags:
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Serial1/0 Flags: A F
R4# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial1/0 Flags: A F
     Pkts: 0/0
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

R6 (RP and First-Hop DR for Source B)

```
R6# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.254.6 (?), elected via Auto-RP
         Uptime: 1d03h, expires: 00:01:59
R6# show ip pim interface df
* implies this system is the DF
Interface                RP               DF Winner        Metric     Uptime
Loopback0                192.168.254.6    *192.168.6.6      0          1d03h
Loopback1                192.168.254.6    *192.168.254.6    0          1d03h
Ethernet0/0              192.168.254.6    *192.168.16.6     0          1d03h
Serial1/0                192.168.254.6    *192.168.67.6     0          1d03h
Serial2/0                192.168.254.6    *192.168.26.6     0          1d03h
R6# show ip mroute 239.195.1.1
Group 239.195.1.1 not found
R6# show ip mrib route 239.195.1.1
No matching routes in MRIB route-DB
R6# show ip mfib 239.195.1.1
Group 239.195.1.1 not found
R6# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d03h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Loopback1, RPF nbr: 192.168.254.6
  Incoming interface list:
    Serial2/0, Accepting/Sparse
    Serial1/0, Accepting/Sparse
    Ethernet0/0, Accepting/Sparse
    Loopback0, Accepting/Sparse
    Loopback1, Accepting/Sparse
R6# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.254.6 Flags:
  Serial2/0 Flags: A
  Serial1/0 Flags: A
  Ethernet0/0 Flags: A
  Loopback0 Flags: A
  Null0 Flags: A
  Loopback1 Flags: A F
R6# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 13951/30/28/6, Other: 0/0/0
   Loopback1 Flags: A F
     Pkts: 13951/0
   Loopback0 Flags: A
   Serial2/0 Flags: A
   Serial1/0 Flags: A
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

R7 (First-Hop DR for Source C)

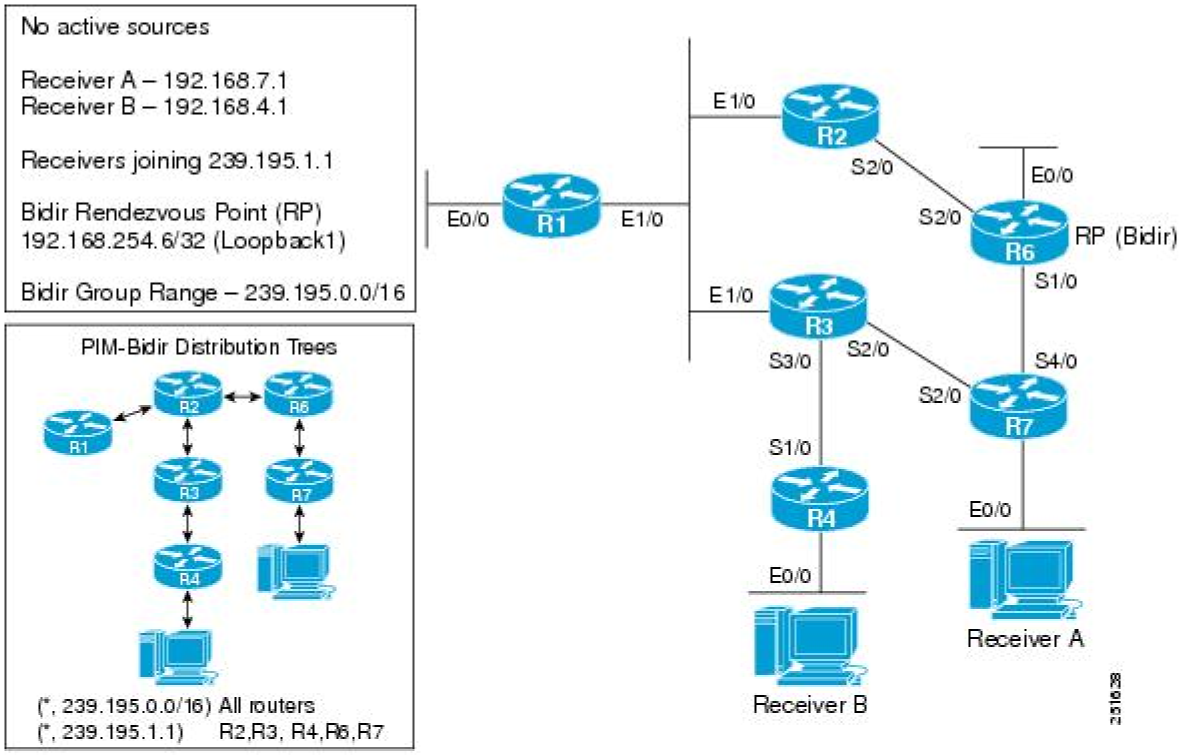
```
R7# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d03h, expires: 00:02:22
R7# show ip pim interface df
* implies this system is the DF
```

```
Interface                RP               DF Winner        Metric     Uptime
Ethernet0/0              192.168.254.6    *192.168.7.7      65          1d03h
Serial2/0                192.168.254.6    *192.168.37.7     65          1d03h
Serial4/0                192.168.254.6     192.168.67.6     0           1d03h
R7# show ip igmp groups 239.195.1.1
IGMP Connected Group Membership
Group Address    Interface              Uptime    Expires   Last Reporter   Group Accounted
R7# show ip mroute 239.195.1.1
Group 239.195.1.1 not found
R7# show ip mrib route 239.195.1.1
No matching routes in MRIB route-DB
R7# show ip mfib 239.195.1.1
Group 239.195.1.1 not found
R7# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d03h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial4/0, RPF nbr: 192.168.67.6
  Incoming interface list:
    Serial2/0, Accepting/Sparse
    Ethernet0/0, Accepting/Sparse
    Serial4/0, Accepting/Sparse
R7# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.67.6 Flags:
  Serial2/0 Flags: A
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Serial4/0 Flags: A F
R7# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 4917/10/28/2, Other: 0/0/0
   Serial4/0 Flags: A F
     Pkts: 4917/0
   Serial2/0 Flags: A
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

Bidir-PIM Example No Active Sources with Interested Receivers

The following example shows how to verify multicast forwarding using the MFIB for bidir-PIM in a network environment where there are no active sources with interested receivers. This verification example is based on the topology shown in the figure.

Figure 44: Bidir-PIM Example Topology: No Active Sources with Interested Receivers



For this verification example, the following conditions apply:

- Entries for (*, 239.195.0.0/16) are created by the control plane based on the PIM group-to-RP mappings on all routers.
- Entries for (*, 239.195.1.1) will be created only when IGMP joins are initiated by interested receivers joining this group. As a result, all routers along the shared tree between the RP and the last-hop DRs that have interested receivers will have state for (*, 239.195.1.1).

Note R1 will not have state for (*, 239.195.1.1) because it is not between the RP and the last-hop DRs.

- In general, multicast packet forwarding can be verified by observing the “SW Forwarding” counter in **show ip mfib** command output for the most specific entry available in the MFIB. If multicast is being forwarded, this counter will increment; however, because there are no active sources in this scenario, this counter will not increment.

R1

```
R1# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
```

```
         Uptime: 1d01h, expires: 00:02:07
R1# show ip pim interface df
* implies this system is the DF
Interface                RP                DF Winner        Metric     Uptime
Ethernet0/0              192.168.254.6    *192.168.1.1       75         1d01h
Ethernet1/0              192.168.254.6     192.168.123.2    65         1d01h
R1# show ip mroute 239.195.1.1
Group 239.195.1.1 not found
R1# show ip mrib route 239.195.1.1
No matching routes in MRIB route-DB
R1# show ip mfib 239.195.1.1
Group 239.195.1.1 not found
R1# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d01h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Ethernet1/0, RPF nbr: 192.168.123.2
  Incoming interface list:
    Ethernet0/0, Accepting/Sparse
    Ethernet1/0, Accepting/Sparse
R1# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.123.2 Flags:
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Ethernet1/0 Flags: A F
R1# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Ethernet1/0 Flags: A F
     Pkts: 0/0
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

R2

```
R2# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d01h, expires: 00:02:32
R2# show ip pim interface df
* implies this system is the DF
Interface                RP                DF Winner        Metric     Uptime
Ethernet1/0              192.168.254.6    *192.168.123.2    65         1d01h
Serial2/0                192.168.254.6     192.168.26.6     0          1d01h
R2# show ip mroute 239.195.1.1
(*, 239.195.1.1), 01:30:22/00:02:50, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial2/0, RPF nbr 192.168.26.6
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse, 01:30:22/00:02:50
    Serial2/0, Bidir-Upstream/Sparse, 01:30:22/00:00:00
R2# show ip mrib route 239.195.1.1
(*,239.195.1.1) RPF nbr: 192.168.26.6 Flags: IA
  Ethernet1/0 Flags: F
  Serial2/0 Flags: F
R2# show ip mfib 239.195.1.1
 (*,239.195.1.1) Flags: IA
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial2/0 Flags: F
     Pkts: 0/0
   Ethernet1/0 Flags: F
     Pkts: 0/0
R2# show ip mroute 239.195.0.0/16
```

```
(*,239.195.0.0/16), 1d01h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial2/0, RPF nbr: 192.168.26.6
  Incoming interface list:
    Ethernet1/0, Accepting/Sparse
    Serial2/0, Accepting/Sparse
R2# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.26.6 Flags:
  Ethernet1/0 Flags: A
  Null0 Flags: A
  Serial2/0 Flags: A F
R2# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial2/0 Flags: A F
     Pkts: 0/0
   Ethernet1/0 Flags: A
   Null0 Flags: A
```

R3

```
R3# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d01h, expires: 00:02:17
R3# show ip pim interface df
* implies this system is the DF
Interface               RP                DF Winner        Metric     Uptime
Ethernet0/0             192.168.254.6    *192.168.3.3      75          1d01h
Ethernet1/0             192.168.254.6     192.168.123.2    65          1d01h
Serial2/0               192.168.254.6     192.168.37.7     65          1d01h
Serial3/0               192.168.254.6    *192.168.34.3     75          1d01h
R3# show ip mroute 239.195.1.1
(*, 239.195.1.1), 01:30:36/00:03:21, RP 192.168.254.6, flags: B
  Bidir-Upstream: Ethernet1/0, RPF nbr 192.168.123.2
  Outgoing interface list:
    Serial3/0, Forward/Sparse, 01:30:36/00:03:21
    Ethernet1/0, Bidir-Upstream/Sparse, 01:30:36/00:00:00
R3# show ip mrib route 239.195.1.1
(*,239.195.1.1) RPF nbr: 192.168.123.2 Flags: IA
  Serial3/0 Flags: F
  Ethernet1/0 Flags: F
R3# show ip mfib 239.195.1.1
 (*,239.195.1.1) Flags: IA
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial3/0 Flags: F
     Pkts: 0/0
   Ethernet1/0 Flags: F
     Pkts: 0/0
R3# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d01h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Ethernet1/0, RPF nbr: 192.168.123.2
  Incoming interface list:
    Serial3/0, Accepting/Sparse
    Ethernet0/0, Accepting/Sparse
    Ethernet1/0, Accepting/Sparse
R3# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.123.2 Flags:
  Serial3/0 Flags: A
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Ethernet1/0 Flags: A F
```

```
R3# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial3/0 Flags: A
   Ethernet1/0 Flags: A F
     Pkts: 0/0
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

R4 (Last-Hop DR for Receiver B)

```
R4# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d01h, expires: 00:02:02
R4# show ip pim interface df
* implies this system is the DF
Interface               RP               DF Winner        Metric     Uptime
Ethernet0/0             192.168.254.6    *192.168.4.4      139        1d01h
Serial1/0               192.168.254.6     192.168.34.3     75         1d01h
R4# show ip igmp groups 239.195.1.1
IGMP Connected Group Membership
Group Address    Interface              Uptime    Expires   Last Reporter   Group Accounted
239.195.1.1      Ethernet0/0              01:30:51  00:02:56  192.168.4.1
R4# show ip mroute 239.195.1.1
(*, 239.195.1.1), 01:30:51/00:02:56, RP 192.168.254.6, flags: BC
  Bidir-Upstream: Serial1/0, RPF nbr 192.168.34.3
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 01:30:51/00:02:56
    Serial1/0, Bidir-Upstream/Sparse, 01:30:51/00:00:00
R4# show ip mrib route 239.195.1.1
(*,239.195.1.1) RPF nbr: 192.168.34.3 Flags: IA
  Ethernet0/0 Flags: F
  Serial1/0 Flags: F
R4# show ip mfib 239.195.1.1
 (*,239.195.1.1) Flags: IA
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial1/0 Flags: F
     Pkts: 0/0
   Ethernet0/0 Flags: F
     Pkts: 0/0
R4# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d01h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial1/0, RPF nbr: 192.168.34.3
  Incoming interface list:
    Ethernet0/0, Accepting/Sparse
    Serial1/0, Accepting/Sparse
R4# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.34.3 Flags:
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Serial1/0 Flags: A F
R4# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial1/0 Flags: A F
     Pkts: 0/0
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

R6 (RP)

```
R6# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.254.6 (?), elected via Auto-RP
         Uptime: 1d01h, expires: 00:02:30
R6# show ip pim interface df
* implies this system is the DF
Interface                RP               DF Winner        Metric     Uptime
Loopback0                192.168.254.6    *192.168.6.6      0          1d01h
Loopback1                192.168.254.6    *192.168.254.6    0          1d01h
Ethernet0/0              192.168.254.6    *192.168.16.6     0          1d01h
Serial1/0                192.168.254.6    *192.168.67.6     0          1d01h
Serial2/0                192.168.254.6    *192.168.26.6     0          1d01h
R6# show ip mroute 239.195.1.1
(*, 239.195.1.1), 01:31:08/00:03:00, RP 192.168.254.6, flags: B
  Bidir-Upstream: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/0, Forward/Sparse, 01:31:00/00:03:00
    Serial2/0, Forward/Sparse, 01:31:08/00:02:57
R6# show ip mrib route 239.195.1.1
(*,239.195.1.1) RPF nbr: 0.0.0.0 Flags: IA
  Serial1/0 Flags: F
  Serial2/0 Flags: F
R6# show ip mfib 239.195.1.1
 (*,239.195.1.1) Flags: IA
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial2/0 Flags: F
     Pkts: 0/0
   Serial1/0 Flags: F
     Pkts: 0/0
R6# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d01h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Loopback1, RPF nbr: 192.168.254.6
  Incoming interface list:
    Serial2/0, Accepting/Sparse
    Serial1/0, Accepting/Sparse
    Ethernet0/0, Accepting/Sparse
    Loopback0, Accepting/Sparse
    Loopback1, Accepting/Sparse
R6# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.254.6 Flags:
  Serial2/0 Flags: A
  Serial1/0 Flags: A
  Ethernet0/0 Flags: A
  Loopback0 Flags: A
  Null0 Flags: A
  Loopback1 Flags: A F
R6# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Loopback1 Flags: A F
     Pkts: 0/0
   Loopback0 Flags: A
   Serial2/0 Flags: A
   Serial1/0 Flags: A
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

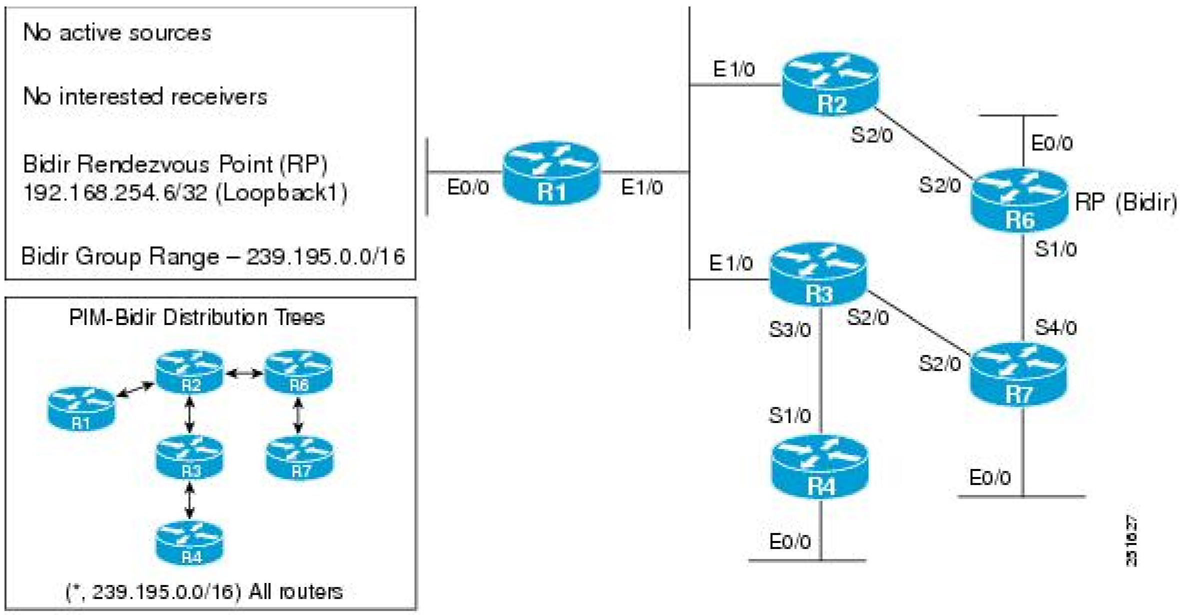
R7 (Last-Hop DR for Receiver A)

```
R7# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 1d01h, expires: 00:02:33
R7# show ip pim interface df
* implies this system is the DF
Interface                RP               DF Winner        Metric     Uptime
Ethernet0/0              192.168.254.6    *192.168.7.7      65         1d01h
Serial2/0                192.168.254.6    *192.168.37.7     65         1d01h
Serial4/0                192.168.254.6     192.168.67.6     0          1d01h
R7# show ip igmp groups 239.195.1.1
IGMP Connected Group Membership
Group Address    Interface              Uptime    Expires   Last Reporter   Group Accounted
239.195.1.1      Ethernet0/0              01:31:14  00:02:22  192.168.7.1
R7# show ip mroute 239.195.1.1
(*, 239.195.1.1), 01:31:14/00:02:22, RP 192.168.254.6, flags: BC
  Bidir-Upstream: Serial4/0, RPF nbr 192.168.67.6
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 01:31:14/00:02:22
    Serial4/0, Bidir-Upstream/Sparse, 01:31:14/00:00:00
R7# show ip mrib route 239.195.1.1
(*,239.195.1.1) RPF nbr: 192.168.67.6 Flags: IA
  Ethernet0/0 Flags: F
  Serial4/0 Flags: F
R7# show ip mfib 239.195.1.1
 (*,239.195.1.1) Flags: IA
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial4/0 Flags: F
     Pkts: 0/0
   Ethernet0/0 Flags: F
     Pkts: 0/0
R7# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 1d01h/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial4/0, RPF nbr: 192.168.67.6
  Incoming interface list:
    Serial2/0, Accepting/Sparse
    Ethernet0/0, Accepting/Sparse
    Serial4/0, Accepting/Sparse
R7# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.67.6 Flags:
  Serial2/0 Flags: A
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Serial4/0 Flags: A F
R7# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial4/0 Flags: A F
     Pkts: 0/0
   Serial2/0 Flags: A
   Ethernet0/0 Flags: A
   Null0 Flags: A
```

Bidir-PIM Example No Active Sources with No Interested Receivers

The following example shows how to verify multicast forwarding using the MFIB for bidir-PIM in a network environment where there are no active sources and no interested receivers. This verification example is based on the topology shown in the figure.

Figure 45: Bidir-PIM Example Topology: No Active Sources with No Interested Receivers



For this verification example, the following conditions apply:

- Entries for (*, 239.195.0.0/16) are created by the control plane based on the PIM group-to-RP mappings on all routers.
- Entries for any group within the range 239.195.0.0/16 will be created only when IGMP joins are initiated by interested receivers joining that particular group. For example, if a multicast receiver joins the group 239.195.1.1, an entry for (*, 239.195.1.1) will be created on all routers along the shared tree.
- Because there are no interested receivers, (*, 239.195.0.0/16) will be the only state in the network on all routers.
- In general, multicast packet forwarding can be verified by observing the “SW Forwarding” counter in **show ip mfib** command output for the most specific entry available in the MFIB. If multicast is being forwarded, this counter will increment; however, because there are no active sources in this scenario, this counter will not increment.

R1

```
R1# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 22:06:01, expires: 00:02:30
R1# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 22:06:01/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Ethernet1/0, RPF nbr: 192.168.123.2
  Incoming interface list:
    Ethernet0/0, Accepting/Sparse
    Ethernet1/0, Accepting/Sparse
R1# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.123.2 Flags:
```

```
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Ethernet1/0 Flags: A F
R1# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Ethernet1/0 Flags: A F
     Pkts: 0/0
   Ethernet0/0 Flags: A
   Null0 Flags: A
R1# show ip pim interface df
* implies this system is the DF
Interface                RP               DF Winner        Metric     Uptime
Ethernet0/0              192.168.254.6    *192.168.1.1      75         22:06:01
Ethernet1/0              192.168.254.6     192.168.123.2    65         22:06:01
```

R2

```
R2# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 22:09:00, expires: 00:02:30
R2# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 22:09:00/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial2/0, RPF nbr: 192.168.26.6
  Incoming interface list:
    Ethernet1/0, Accepting/Sparse
    Serial2/0, Accepting/Sparse
R2# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.26.6 Flags:
  Ethernet1/0 Flags: A
  Null0 Flags: A
  Serial2/0 Flags: A F
R2# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial2/0 Flags: A F
     Pkts: 0/0
   Ethernet1/0 Flags: A
   Null0 Flags: A
R2# show ip pim interface df
* implies this system is the DF
Interface                RP               DF Winner        Metric     Uptime
Ethernet1/0              192.168.254.6    *192.168.123.2    65         22:09:00
Serial2/0                192.168.254.6     192.168.26.6     0          22:09:00
```

R3

```
R3# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 22:09:20, expires: 00:02:12
R3# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 22:09:20/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Ethernet1/0, RPF nbr: 192.168.123.2
  Incoming interface list:
    Serial3/0, Accepting/Sparse
```

```
    Ethernet1/0, Accepting/Sparse
R3# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.123.2 Flags:
  Serial3/0 Flags: A
  Null0 Flags: A
  Ethernet1/0 Flags: A F
R3# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial3/0 Flags: A
   Ethernet1/0 Flags: A F
     Pkts: 0/0
   Null0 Flags: A
R3# show ip pim interface df
* implies this system is the DF
Interface                RP                DF Winner        Metric     Uptime
Ethernet1/0              192.168.254.6      192.168.123.2    65         22:09:20
Serial2/0                192.168.254.6      192.168.37.7     65         22:09:20
Serial3/0                192.168.254.6     *192.168.34.3     75         22:09:20
```

R4

```
R4# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 22:09:47, expires: 00:02:42
R4# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 22:09:47/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial1/0, RPF nbr: 192.168.34.3
  Incoming interface list:
    Ethernet0/0, Accepting/Sparse
    Serial1/0, Accepting/Sparse
R4# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.34.3 Flags:
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Serial1/0 Flags: A F
R4# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial1/0 Flags: A F
     Pkts: 0/0
   Ethernet0/0 Flags: A
   Null0 Flags: A
R4# show ip pim interface df
* implies this system is the DF
Interface                RP                DF Winner        Metric     Uptime
Ethernet0/0              192.168.254.6     *192.168.4.4      139        22:09:47
Serial1/0                192.168.254.6      192.168.34.3     75         22:09:47
```

R6 (RP)

```
R6# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.254.6 (?), elected via Auto-RP
```

```
         Uptime: 22:11:08, expires: 00:02:48
R6# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 22:11:08/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Loopback1, RPF nbr: 192.168.254.6
  Incoming interface list:
    Serial2/0, Accepting/Sparse
    Serial1/0, Accepting/Sparse
    Ethernet0/0, Accepting/Sparse
    Loopback1, Accepting/Sparse
R6# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.254.6 Flags:
  Serial2/0 Flags: A
  Serial1/0 Flags: A
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Loopback1 Flags: A F
R6# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Loopback1 Flags: A F
     Pkts: 0/0
   Serial2/0 Flags: A
   Serial1/0 Flags: A
   Ethernet0/0 Flags: A
   Null0 Flags: A
R6# show ip pim interface df
* implies this system is the DF
Interface               RP                DF Winner        Metric     Uptime
Loopback1               192.168.254.6    *192.168.254.6    0           22:11:08
Ethernet0/0             192.168.254.6    *192.168.16.6     0           22:11:08
Serial1/0               192.168.254.6    *192.168.67.6     0           22:11:08
Serial2/0               192.168.254.6    *192.168.26.6     0           22:11:08
```

R7

```
R7# show ip pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.195.0.0/16
  RP 192.168.254.6 (?), v2v1, bidir
    Info source: 192.168.6.6 (?), elected via Auto-RP
         Uptime: 22:10:23, expires: 00:02:04
R7# show ip mroute 239.195.0.0/16
(*,239.195.0.0/16), 22:10:23/-, RP 192.168.254.6, flags: B
  Bidir-Upstream: Serial4/0, RPF nbr: 192.168.67.6
  Incoming interface list:
    Serial2/0, Accepting/Sparse
    Ethernet0/0, Accepting/Sparse
    Serial4/0, Accepting/Sparse
R7# show ip mrib route 239.195.0.0/16
(*,239.195.0.0/16) RPF nbr: 192.168.67.6 Flags:
  Serial2/0 Flags: A
  Ethernet0/0 Flags: A
  Null0 Flags: A
  Serial4/0 Flags: A F
R7# show ip mfib 239.195.0.0/16
 (*,239.195.0.0/16) Flags:
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   Serial4/0 Flags: A F
     Pkts: 0/0
   Serial2/0 Flags: A
   Ethernet0/0 Flags: A
   Null0 Flags: A
R7# show ip pim interface df
```

```
* implies this system is the DF
Interface            RP              DF Winner       Metric     Uptime
Ethernet0/0          192.168.254.6   *192.168.7.7     65          22:10:23
Serial2/0            192.168.254.6   *192.168.37.7    65          22:10:23
Serial4/0            192.168.254.6    192.168.67.6    0           22:10:23
```

Additional References

Related Documents

Related Topic	Document Title
MFIB overview concepts and MFIB/MRIB entry and interface flag descriptions	" Multicast Forwarding Information Base Overview "
Overview of the IP multicast technology area	" IP Multicast Technology Overview "
Concepts, tasks, and examples for configuring an IP multicast network using PIM	" Configuring Basic IP Multicast "
IP multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Verifying IPv4 Multicast Forwarding Using the MFIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 19: Feature Information for Verifying IPv4 Multicast Forwarding Using the MFIB

Feature Name	Releases	Feature Information
IPv4 Multicast Support of the MFIB	15.0(1)M 12.2(33)SRE	The MFIB architecture provides modularity and separation between the multicast control plane (PIM and IGMP) and the multicast forwarding plane (MFIB). This architecture is used in Cisco IOS IPv6 and Cisco IOS XR multicast implementations. With the introduction of the IPv4 MFIB infrastructure, the Cisco IOS IPv4 multicast implementation has been enhanced, making the MFIB forwarding model the only forwarding engine used. The following commands were introduced or modified: **clear ip mfib counters**, **debug ip mcache**, **debug ip mfib adjacency**, **debug ip mfib db**, **debug ip mfib fs**, **debug ip mfib init**, **debug ip mfib interface**, **debug ip mfib mrib**, **debug ip mfib pak**, **debug ip mfib platform**, **debug ip mfib ppr**, **debug ip mfib ps**, **debug ip mfib signal**, **debug ip mfib table**, **debug ip mpacket**, **debug ip mrib**, **ip mfib**, **ip mfib cef**, **ip mfib forwarding**, **ip mroute-cache**, **ip multicast cache-headers**, **ip multicast rate-limit**, **ip multicast ttl-threshold**, **ip pim register-rate-limit**, **show ip mcache**, **show ip mfib**, **show ip mfib active**, **show ip mfib count**, **show ip mfib interface**, **show ip mfib route**, **show ip mfib status**, **show ip mfib summary**, **show ip pim interface**, **show ip pim tunne**l.

CHAPTER 31

Distributed MFIB for IPv6 Multicast

Distributed MFIB (dMFIB) is used to switch multicast IPv6 packets on distributed platforms. The basic MFIB routines that implement the core of the forwarding logic are common to all forwarding environments.

Information About Distributed MFIB for IPv6 Multicast

Distributed MFIB

Distributed Multicast Forwarding Information Base (MFIB) is used to switch multicast IPv6 packets on distributed platforms. Distributed MFIB may also contain platform-specific information on replication across line cards. The basic MFIB routines that implement the core of the forwarding logic are common to all forwarding environments.

dMFIB implements the following functions:

- Distributes a copy of the MFIB to the line cards.
- Relays data-driven protocol events generated in the line cards to PIM.
- Provides an MFIB platform application program interface (API) to propagate MFIB changes to platform-specific code responsible for programming the hardware acceleration engine. This API also includes entry points to switch a packet in software (necessary if the packet is triggering a data-driven event) and to upload traffic statistics to the software.
- Provides hooks to allow clients residing on the RP to read traffic statistics on demand. Distributed MFIB does not periodically upload these statistics to the RP.

The combination of distributed MFIB and MRIB subsystems allows the device to have a "customized" copy of the MFIB database in each line card and to transport MFIB-related platform-specific information from the RP to the line cards.

How to Disable MFIB on a Distributed Platform

Before you begin

SUMMARY STEPS

1.

DETAILED STEPS

Example:

Example:

What to do next

Disabling MFIB on a Distributed Platform

Multicast forwarding is automatically enabled when IPv6 multicast routing is enabled. However, you may want to disable multicast forwarding on a distributed platform.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 mfib-mode centralized-only**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ipv6 mfib-mode centralized-only** **Example:** `Device(config)# ipv6 mfib-mode centralized-only`	Disables distributed forwarding on a distributed platform.

Configuration Example for Distributed MFIB for IPv6 Multicast

This example shows how to disable multicast forwarding on a distributed platform:

```
Device(config)# ipv6 mfib-mode centralized-only
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference
IPv6 commands	*Cisco IOS IPv6 Command Reference*
IPv6 features	Cisco IOS IPv6 Feature Mapping
IPv6 addressing and connectivity	*IPv6 Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Distributed MFIB for IPv6 Multicast

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 20: Feature Information for Distributed MFIB for IPv6 Multicast

Feature Name	Releases	Feature Information
Distributed MFIB for IPv6 Multicast	12.0(26)S 12.2(25)S 12.2(28)SB 12.3(4)T 12.4 Cisco IOS XE Release 2.1	Distributed MFIB is used to switch multicast IPv6 packets on distributed platforms. The following command was introduced: **ipv6 mfib-mode centralized-only**.

CHAPTER 32

MLDP-Based MVPN

The MLDP-based MVPN feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network (MVPN) core network.

Information About MLDP-Based MVPN

Overview of MLDP-Based MVPN

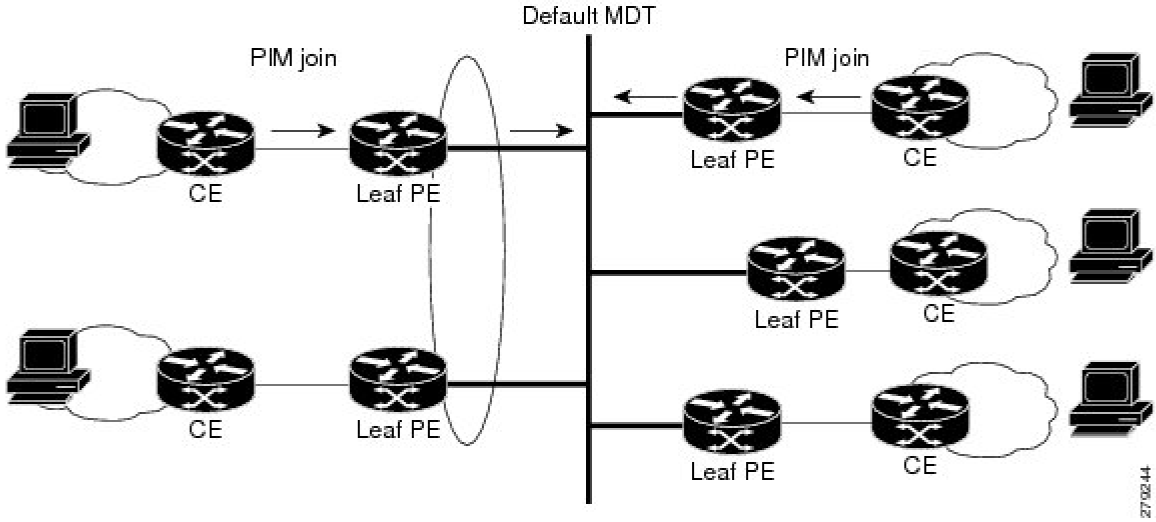
MVPN allows a service provider to configure and support multicast traffic in an MPLS VPN environment. This feature supports routing and forwarding of multicast packets for each individual VPN routing and forwarding (VRF) instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone.

A VPN is network connectivity across a shared infrastructure, such as an Internet service provider (ISP). Its function is to provide the same policies and performance as a private network, at a reduced cost of ownership, thus creating many opportunities for cost savings through operations and infrastructure.

An MVPN allows an enterprise to transparently interconnect its private network across the network backbone of a service provider. The use of an MVPN to interconnect an enterprise network in this way does not change the way that the enterprise network is administered, nor does it change general enterprise connectivity.

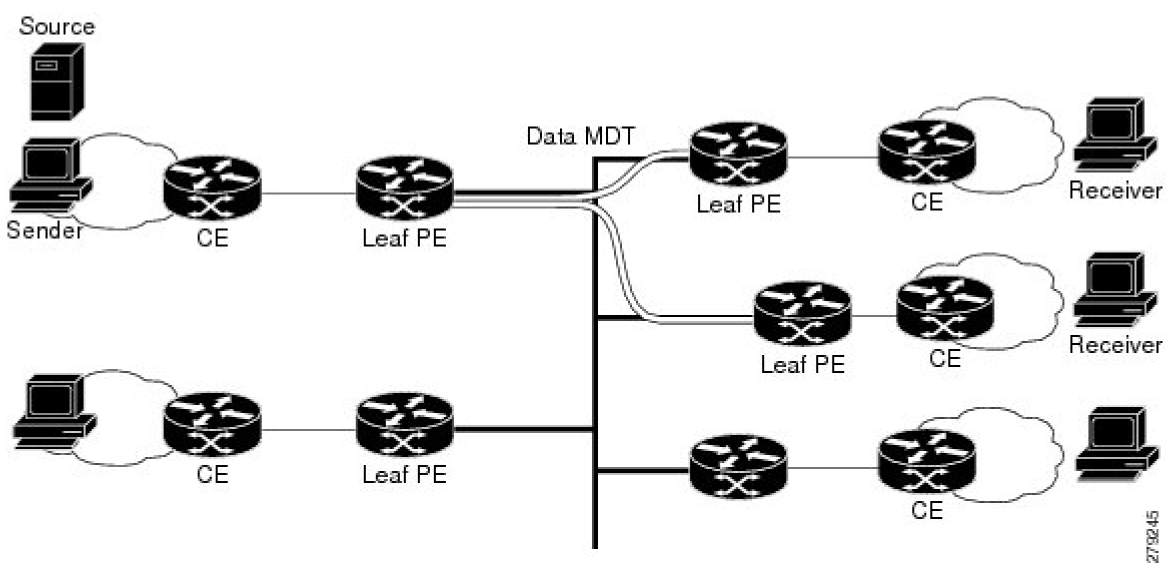
As shown in the figure, in an MLDP-based MVPN, a static default multicast distribution tree (MDT) is established for each multicast domain. The default MDT defines the path used by provider edge (PE) devices to send multicast data and control messages to every other PE device in the multicast domain. A default MDT is created in the core network using a single MP2MP LSP. The default MDT behaves like a virtual LAN.

Figure 46: MLDP with the Default MDT Scenario



As shown in the figure, an MLDP-based MVPN also supports the dynamic creation of data MDTs for high-bandwidth transmission. For high-rate data sources, a data MDT is created using P2MP LSPs to off-load traffic from the default MDT to avoid unnecessary waste of bandwidth to PEs that did not join the stream. The creation of the data MDT is signaled dynamically using MDT Join TLV messages. Data MDTs are a feature unique to Cisco IOS software. Data MDTs are intended for high-bandwidth sources such as full-motion video inside the VPN to ensure optimal traffic forwarding in the MPLS VPN core. The threshold at which the data MDT is created can be configured on a per-device or a per-VRF basis. When the multicast transmission exceeds the defined threshold, the sending PE device creates the data MDT and sends a User Datagram Protocol (UDP) message, which contains information about the data MDT to all devices on the default MDT.

Figure 47: MLDP with the Data MDT Scenario



Data MDTs are created only for (S, G) multicast route entries within the VRF multicast routing table. They are not created for (*, G) entries regardless of the value of the individual source data rate.

The only transport mechanism previously available was Protocol Independent Multicast (PIM) with Multipoint Generic Routing Encapsulation (mGRE) over an IP core network. The introduction of Multicast Label Distribution Protocol (MLDP) provides transport by using MLDP with label encapsulation over an MPLS core network.

MLDP creates the MDTs as follows:

- The default MDT uses MP2MP LSPs.
  - Supports low bandwidth and control traffic between VRFs.
- The data MDT uses P2MP LSPs.
  - Supports a single high-bandwidth source stream from a VRF.

All other operations of MVPN remain the same regardless of the tunneling mechanism:

- PIM neighbors in a VRF are seen across a Label Switched Path virtual interface (LSP-VIF).
- The VPN multicast state is signaled by PIM.

The only other difference when using MLDP is that the MDT group address used in the mGRE solution is replaced with a VPN ID.

Benefits of MLDP-Based MVPN

- Enables the use of a single MPLS forwarding plane for both unicast and multicast traffic.
- Enables existing MPLS protection (for example, MPLS Traffic Engineering/Resource Reservation Protocol (TE/RSVP link protection) and MPLS Operations Administration and Maintenance (OAM) mechanisms to be used for multicast traffic.
- Reduces operational complexity due to the elimination of the need for PIM in the MPLS core network.

Initial Deployment of an MLDP-Based MVPN

Initial deployment of an MLDP-based MVPN involves the configuration of a default MDT and one or more data MDTs.

A static default MDT is established for each multicast domain. The default MDT defines the path used by PE devices to send multicast data and control messages to every other PE device in the multicast domain. A default MDT is created in the core network using a single MP2MP LSP.

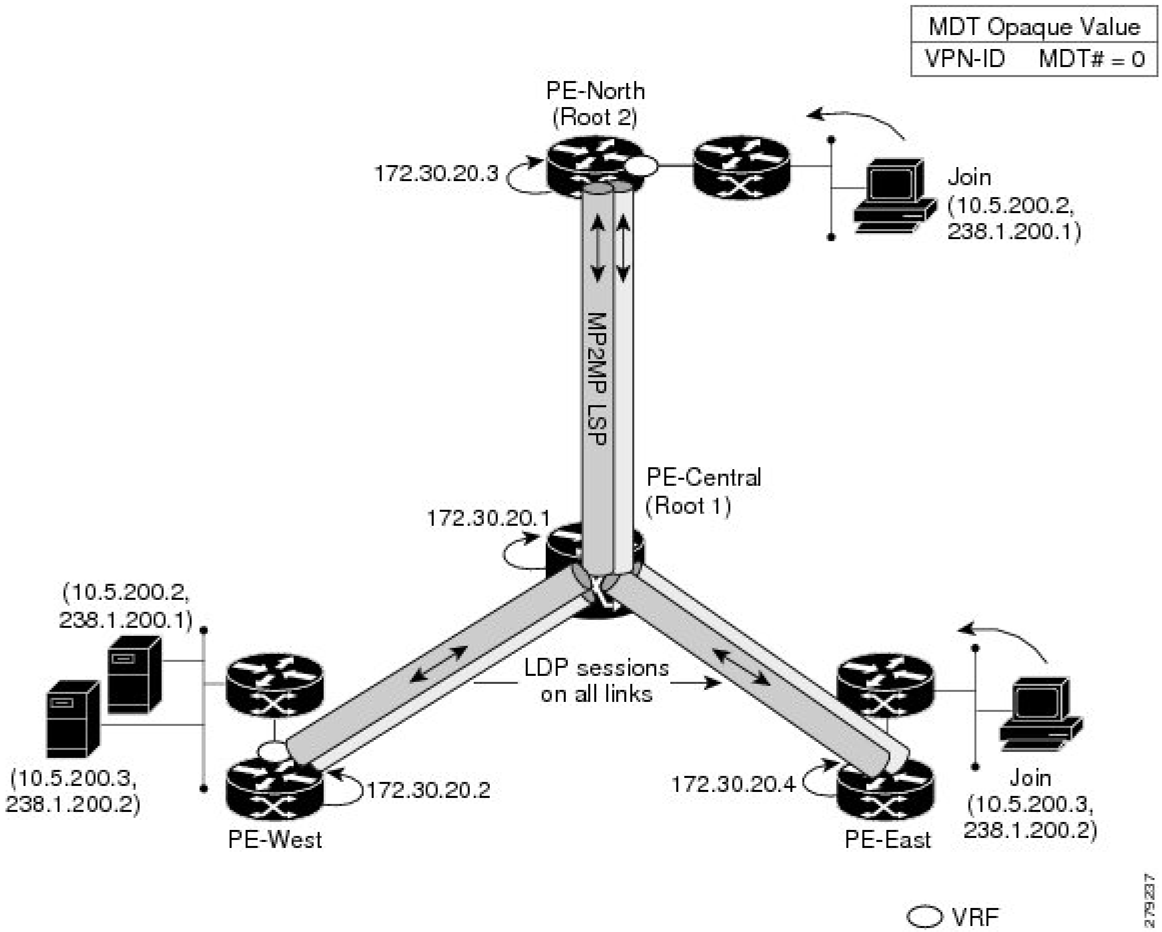
An MLDP-based MVPN also supports the dynamic creation of data MDTs for high-bandwidth transmission.

Default MDT Creation

The figure shows the default MDT scenario. The Opaque value used to signal a default MDT consists of two parameters: the VPN ID and the MDT number for the VPN in the format (vpn-id, 0) where vpn-id is a manually configured 7-byte number that uniquely identifies this VPN. The default MDT is set to zero.

In this scenario, each of the three PE devices belong to the VRF called VRF and they have the same VPN ID. Each PE device with the same VPN ID will join the same MP2MP tree. The PE devices have created a primary MP2MP tree rooted at P-Central (Root 1) and a backup MP2MP tree rooted at PE-North (Root 2). There are two sources at PE-West and interested receivers at both PE-North and PE-East. PE-West will choose one of the MP2MP trees to transmit the customer VPN traffic, but all PE devices can receive traffic on either of the MP2MP trees.

Figure 48: Default MDT Scenario



LSP Downstream Default MDT Creation

The figures show the downstream tree creation for each of the roots. Each PE device configured with VPN ID 100:2 creates the same Forwarding Equivalence Class (FEC) Type Length Value (TLV), but with a different root and downstream labels per MP2MP tree. The FEC type will be MP2MP Down, which prompts the receiving Label Switched Route (LSR) to respond with an upstream label mapping message to create the upstream path.

Figure 49: Default MDT Downstream--Root 1

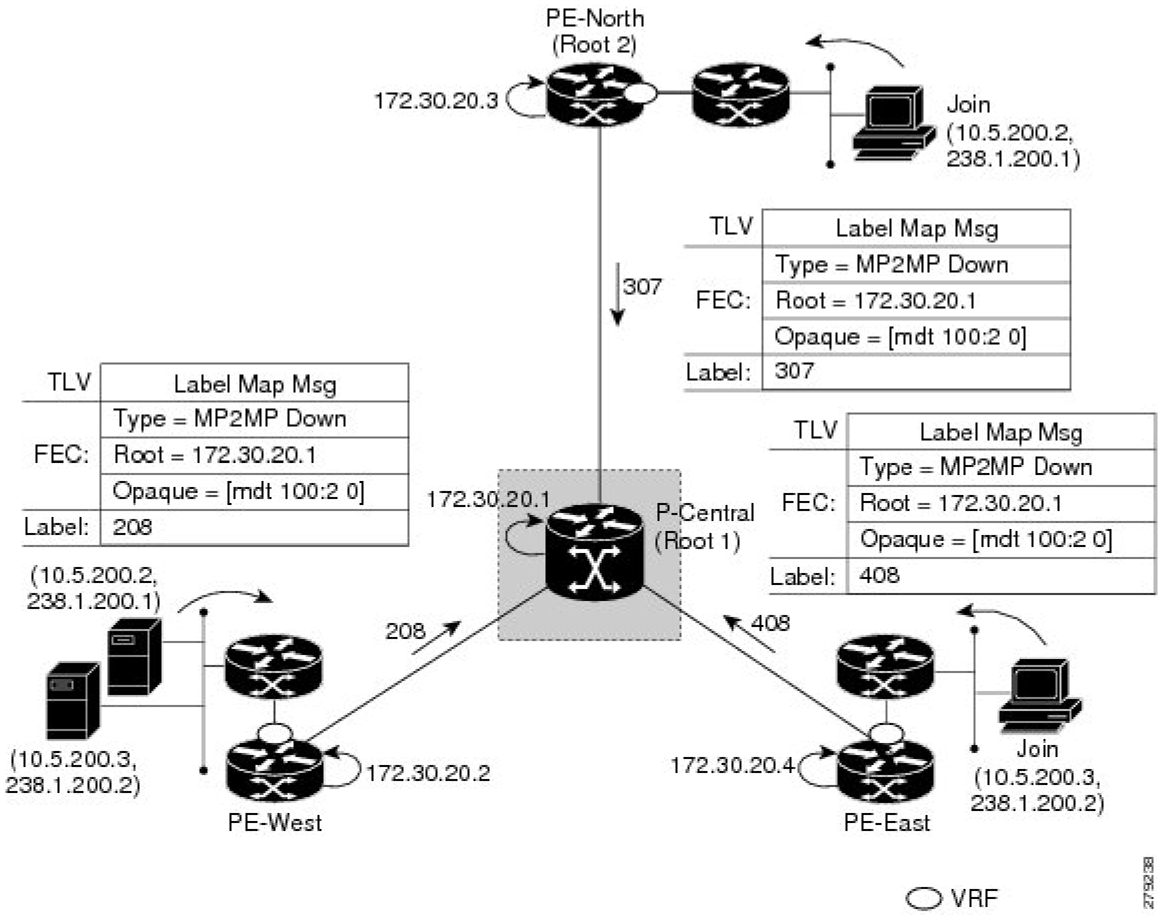
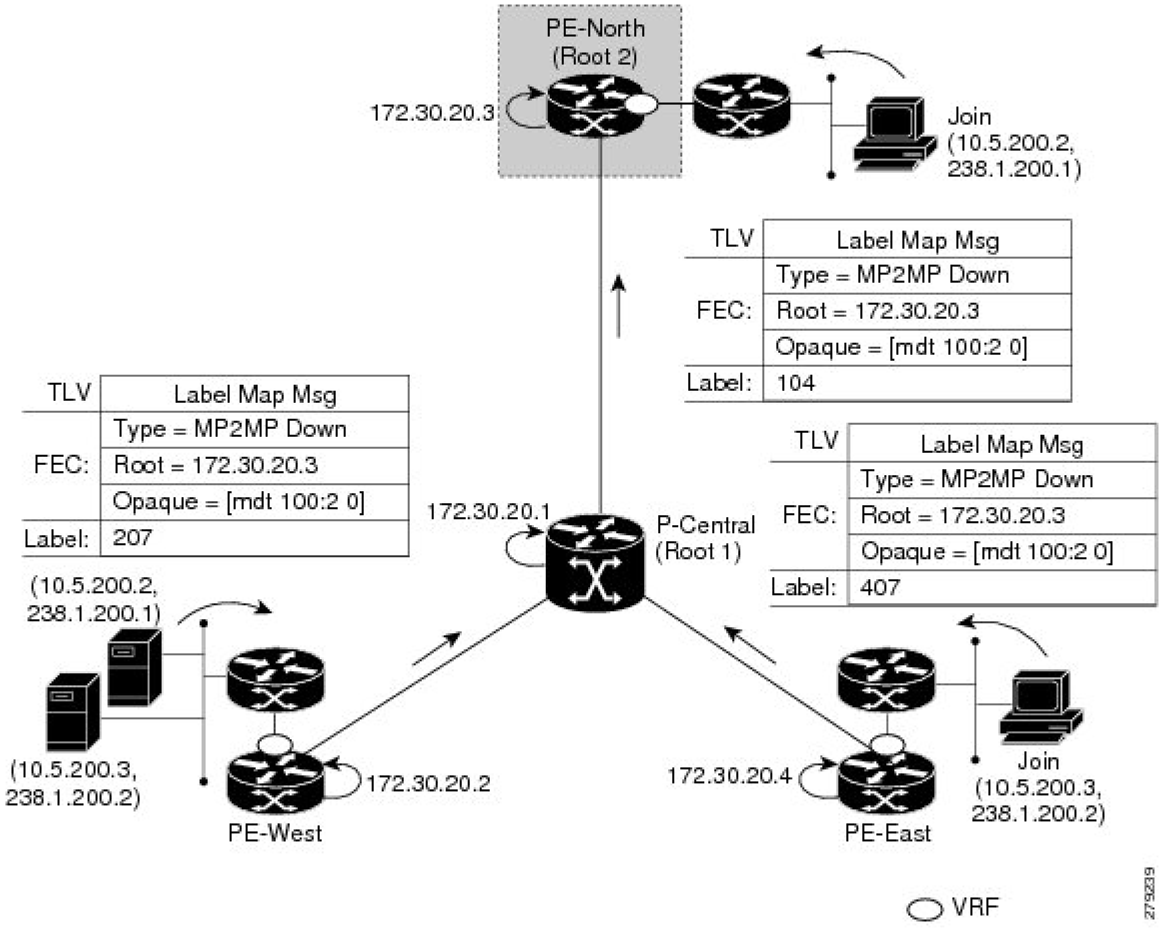


Figure 50: Default MDT Downstream--Root 2



LSP Upstream Default MDT Creation

The figures show the upstream LSP creation for the default MDTs. For each downstream label received, a corresponding upstream label is sent. In the first figure, P-Central sends out three upstream labels (111, 109, and 105) to each downstream directly connected neighbor (downstream is away from the root). The process for PE-North is the same except that it only sends a single upstream label (313) as there is only one directly connected downstream neighbor, as shown in the second figure.

Figure 51: Default MDT Upstream--Root 1

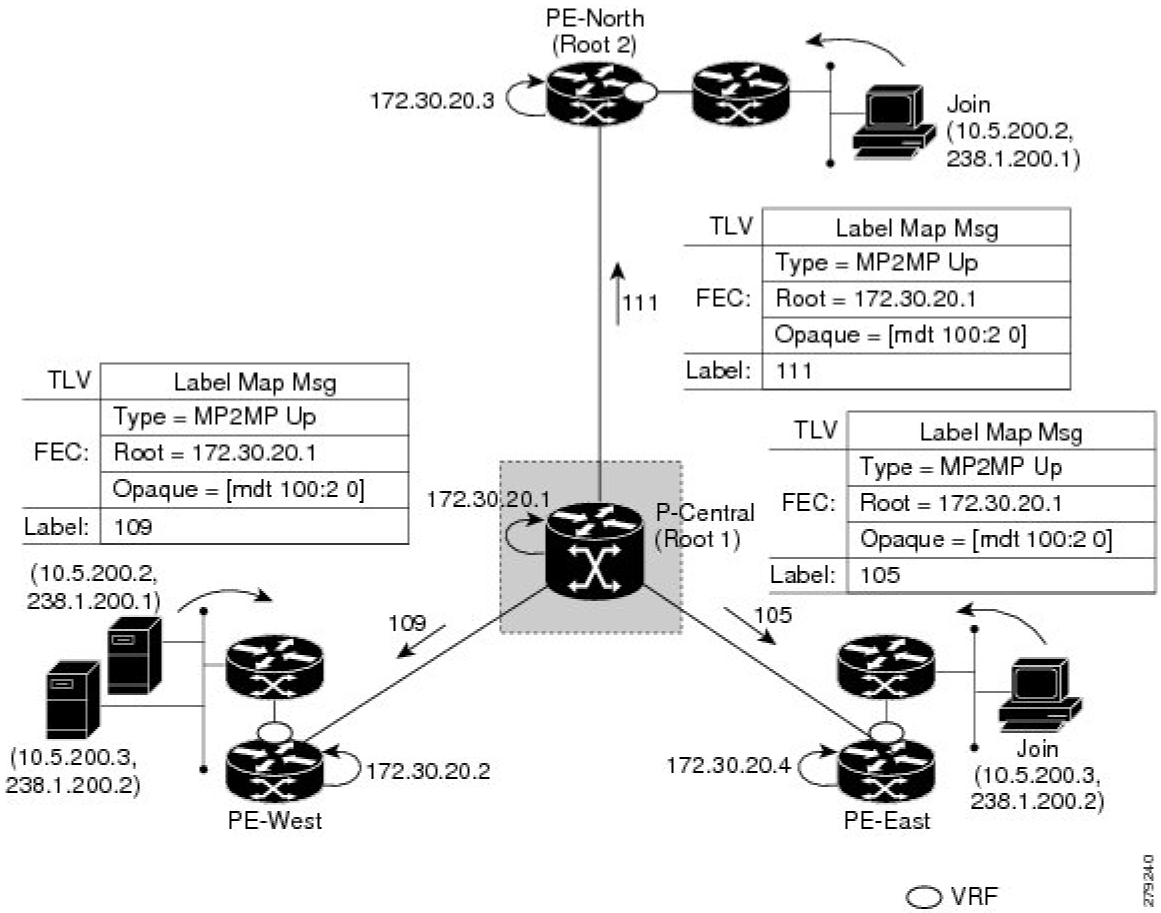
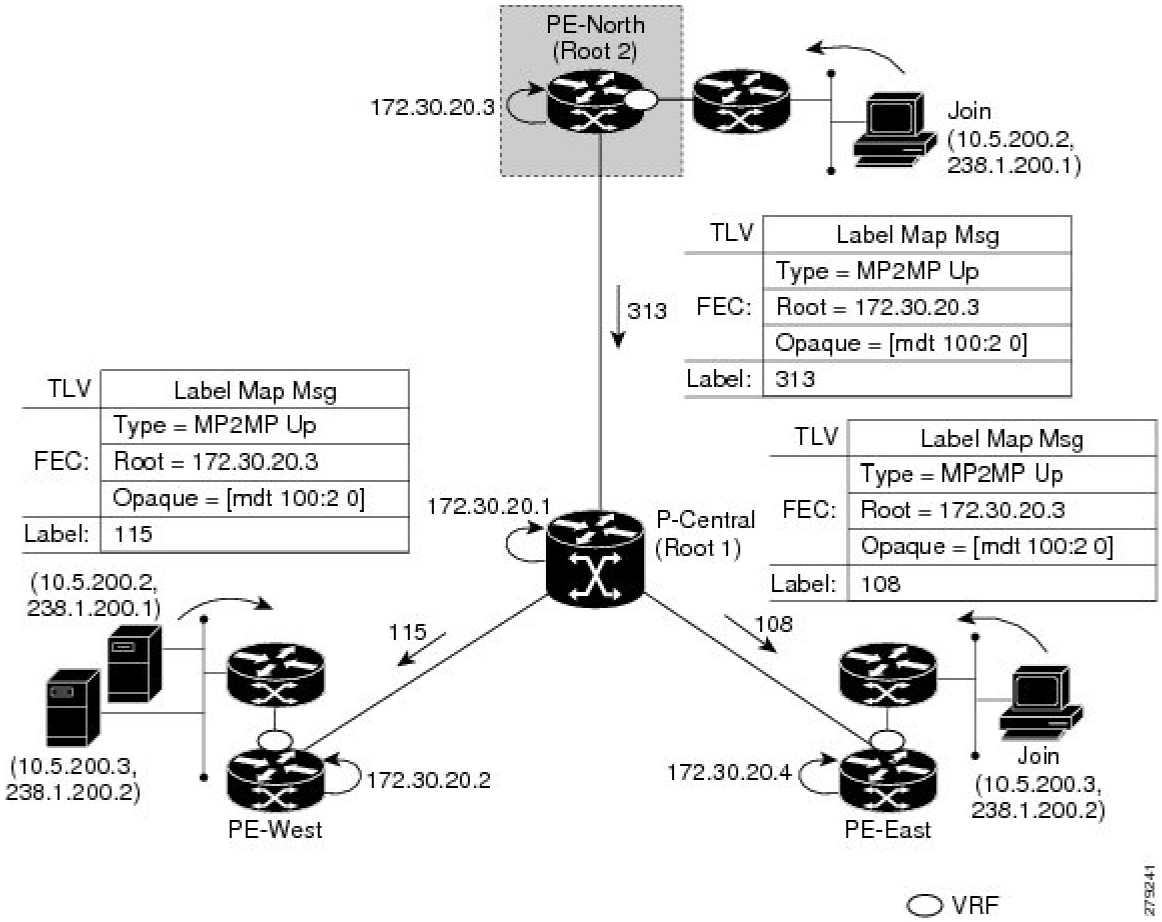


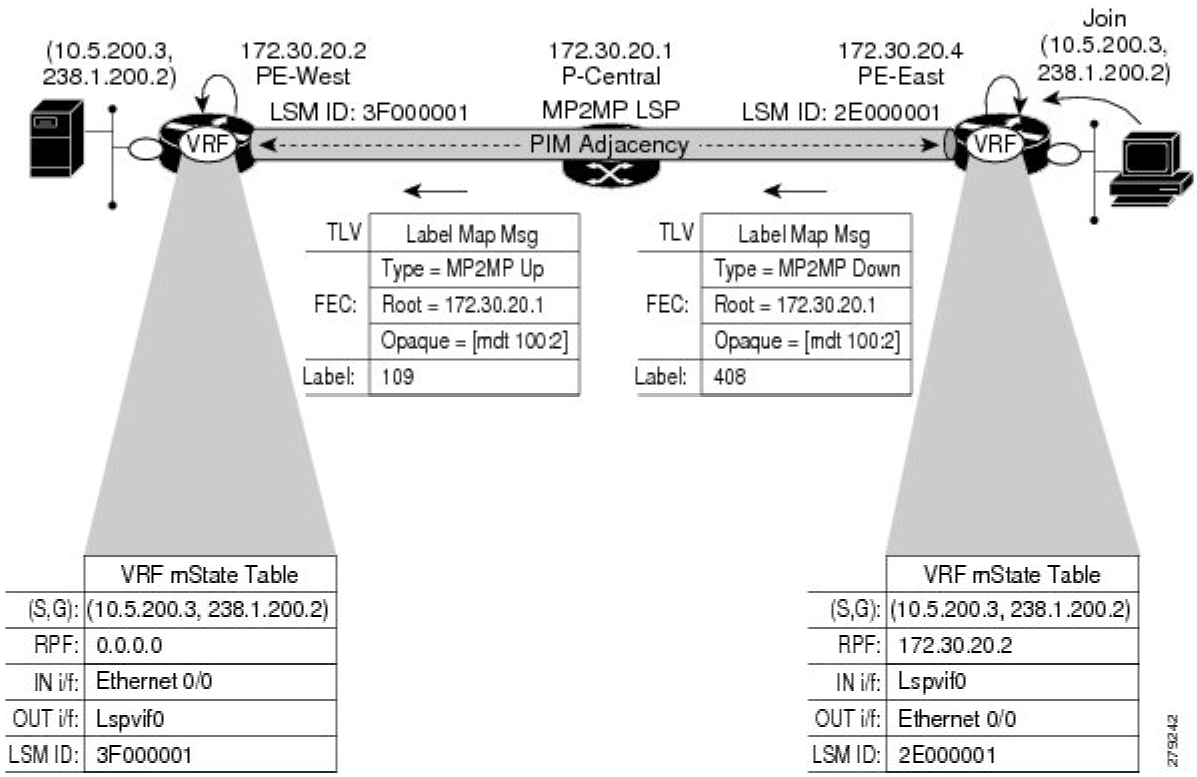
Figure 52: Default MDT Upstream--Root 2



PIM Overlay Signaling of VPN Multicast State

The signaling of the multicast state within a VPN is via PIM. It is called overlay signaling because the PIM session runs over the multipoint LSP and maps the VPN multicast flow to the LSP. In an MVPN, the operation of PIM is independent of the underlying tunnel technology. In the MVPN solution, a PIM adjacency is created between PE devices, and the multicast states within a VRF are populated over the PIM sessions. When using MLDP, the PIM session runs over an LSP-VIF interface. The figure shows PIM signaling running over the default MDT MP2MP LSP. Access to the MP2MP LSP is via the LSP-VIF, which can see all the leaf PE devices at the end of branches, much like a LAN interface. In the figure, PE-East sends a downstream label mapping message to the root, P-Central, which in turn sends an upstream label mapping message to PE-West. These messages result in the creation of the LSP between the two leaf PE devices. A PIM session can then be activated over the top of the LSP allowing the (S, G) states and control messages to be signaled between PE-West and PE-East. In this case, PE-East receives a Join TLV message for (10.5.200.3, 238.1.200.2) within VRF, which it inserts into the mroute table. The Join TLV message is then sent via the PIM session to PE-West (BGP next-hop of 10.5.200.3), which populates its VRF mroute table. This procedure is identical to the procedure using an mGRE tunnel.

Figure 53: PIM Signaling over LSP



Data MDT Scenario

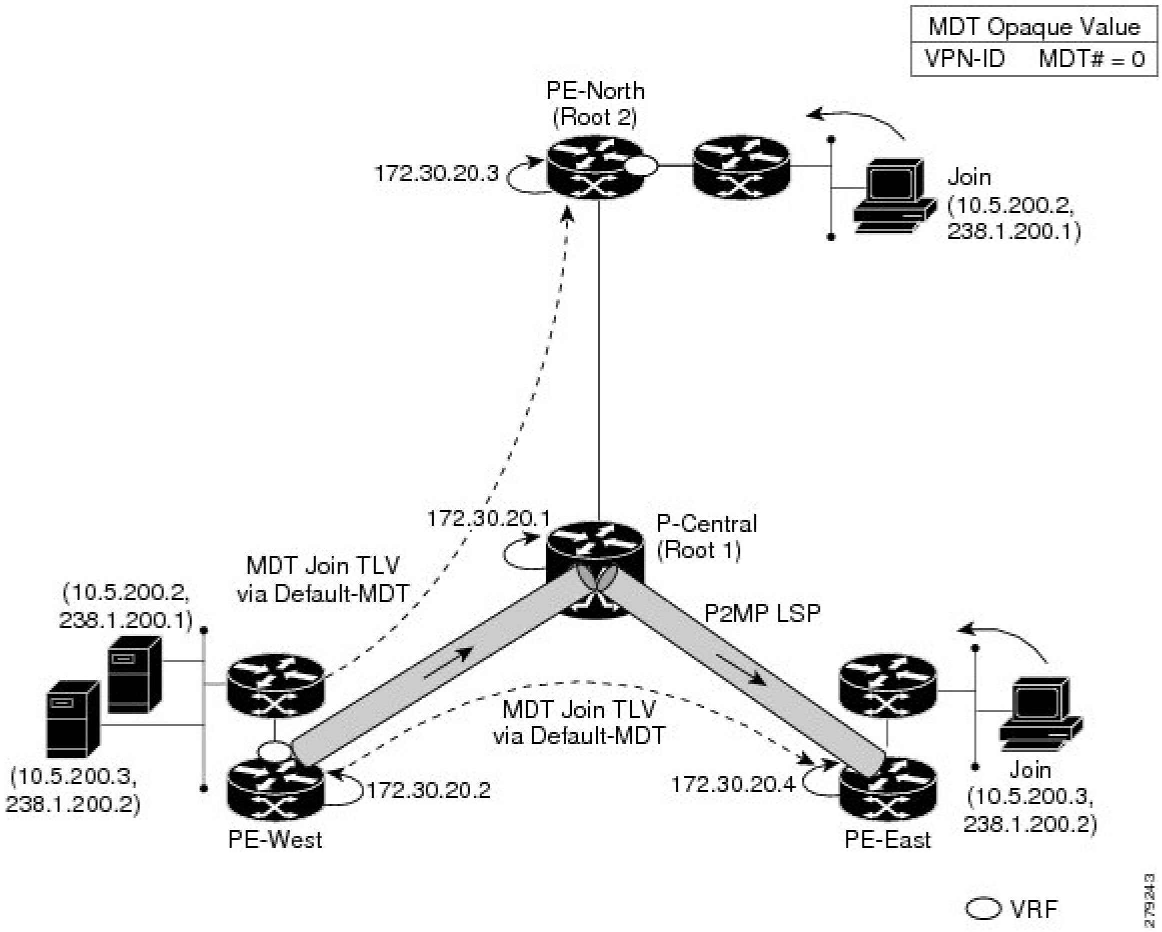
In an MVPN, traffic that exceeds a certain threshold can move off the default MDT onto a data MDT.

The figure shows the data MDT scenario. The Opaque value used to signal a data MDT consists of two parameters: the VPN ID and the MDT number in the format (vpn-id, MDT# > 0) where vpn-id is a manually configured 7-byte number that uniquely identifies this VPN. The second parameter is the unique data MDT number for this VPN, which is a number greater than zero.

In the scenario, two receivers at PE-North and PE-East are interested in two sources at PE-West. If the source 10.5.200.3 exceeds the threshold on the default MDT, PE-West will issue an MDT Join TLV message over the default MDT MP2MP LSP advising all PE devices that a new data MDT is being created.

Because PE-East has an interested receiver in VRF, it will build a multipoint LSP using P2MP back to PE-West, which will be the root of the tree. PE-North does not have a receiver for 10.5.200.3, therefore it will just cache the Join TLV message.

Figure 54: Data MDT Scenario



How to Configure MLDP-Based MVPN

Configuring Initial MLDP Settings

Perform this task to configure the initial MLDP settings.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls mldp logging notifications**
4. **mpls mldp forwarding recursive**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls mldp logging notifications** **Example:** `Device(config)# mpls mldp logging notifications`	Enables MLDP logging notifications.
Step 4	**mpls mldp forwarding recursive** **Example:** `Device(config)# mpls mldp forwarding recursive`	Enables MLDP recursive forwarding over a P2MP LSP.
Step 5	**end** **Example:** `Device(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.

Configuring an MLDP-Based MVPN

Perform this task to configure an MLDP-based MVPN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing**
4. **ip multicast-routing vrf** *vrf-name*
5. **ip vrf** *vrf-name*
6. **rd** *route-distinguisher*
7. **vpn id** *oui* **:** *vpn-index*
8. **route target export** *route-target-ext-community*
9. **route target import** *route-target-ext-community*
10. **mdt preference** *{* **mldp** | **pim** *}*
11. **mdt default mpls mldp** *group-address*
12. **mdt data mpls mldp** *number-of-data-mdt*
13. **mdt data threshold** *kb/s* **list** *access-list*

14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** **Example:** `Device(config)# ip multicast-routing`	Enables IP multicast routing.
Step 4	**ip multicast-routing vrf** *vrf-name* **Example:** `Device(config)# ip multicast-routing vrf VRF`	Enables IP multicast routing for the MVPN VRF specified for the *vrf-name* argument.
Step 5	**ip vrf** *vrf-name* **Example:** `Device(config-vrf)# ip vrf VRF`	Defines a VRF instance and enters VRF configuration mode.
Step 6	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 50:11`	Creates a route distinguisher (RD) (in order to make the VRF functional). Creates the routing and forwarding tables, associates the RD with the VRF instance, and specifies the default RD for a VPN.
Step 7	**vpn id** *oui* **:** *vpn-index* **Example:** `Device(config-vrf)# vpn id 50:10`	Sets or updates the VPN ID on a VRF instance.
Step 8	**route target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route target export 100:100`	Creates an export route target extended community for the specified VRF.
Step 9	**route target import** *route-target-ext-community* **Example:**	Creates an import route target extended community for the specified VRF.

	Command or Action	Purpose
	`Device(config-vrf)# route target import 100:100`	
Step 10	**mdt preference** { **mldp** | **pim** } **Example:** `Device(config-vrf)# mdt preference mldp`	Specifies a preference for a particular MDT type (MLDP or PIM).
Step 11	**mdt default mpls mldp** *group-address* **Example:** `Device(config-vrf)# mdt default mpls mldp 172.30.20.1`	Configures a default MDT group for a VPN VRF instance.
Step 12	**mdt data** **mpls mldp** *number-of-data-mdt* **Example:** `Device(config-vrf)# mdt data mpls mldp 255`	Specifies a range of addresses to be used in the data MDT pool.
Step 13	**mdt data threshold** *kb/s* **list** *access-list* **Example:** `Device(config-vrf)# mdt data threshold 40 list 1`	Defines the bandwidth threshold value in kilobits per second.
Step 14	**end** **Example:** `Device(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.

Verifying the Configuration of an MLDP-Based MVPN

Perform this task in privileged EXEC mode to verify the configuration of an MLDP-based MVPN.

SUMMARY STEPS

1. show mpls mldp database
2. **show ip pim neighbor** [**vrf** *vrf-name*] **neighbor** [*interface-type interface-number*]
3. **show ip mroute** [**vrf** *vrf-name*] [[**active** [*kbps*] [**interface** *type number*] | **bidirectional** | **count** [**terse**] | **dense** | **interface** *type number* | **proxy** | **pruned** | **sparse** | **ssm** | **static** | **summary**] | [*group-address* [*source-address*]] [**count** [**terse**] | **interface** *type number* | **proxy** | **pruned** | **summary**] | [*source-address group-address*] [**count** [**terse**] | **interface** *type number* | **proxy** | **pruned** | **summary**] | [*group-address*] **active** [*kbps*] [**interface** *type number* | **verbose**]]
4. **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [**-** *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
5. **show adjacency** [*ip-address*] [*interface-type interface-number* | **null** *number* | **port-channel** *number* | **sysclock** *number* | **vlan** *number* | **fcpa** *number* | **serial** *number*] [**connectionid** *number*] [**link** {**ipv4** | **mpls**}] [**detail** | **encapsulation**]

DETAILED STEPS

Step 1 show mpls mldp database

Enter the **show mpls mldp database**command to display information in the MLDP database. It shows the FEC, the Opaque value of the FEC decoded, and the replication clients associated with it:

Example:

```
Device# show mpls mldp database
* Indicates MLDP recursive forwarding is enabled
LSM ID : D3000001 (RNR LSM ID: 8A000002)   Type: MP2MP   Uptime : 00:04:54
  FEC Root           : 172.30.20.1
  Opaque decoded     : [mdt 100:2 0]
  Opaque length      : 11 bytes
  Opaque value       : 07 000B 0000010000000100000000
  RNR active LSP     : (this entry)
  Upstream client(s) :
    172.30.20.1:0    [Active]
      Expires        : Never         Path Set ID  : 99000001
      Out Label (U)  : 32            Interface    : Ethernet1/0*
      Local Label (D): 30            Next Hop     : 10.0.1.7
  Replication client(s):
    MDT  (VRF VRF)
      Uptime         : 00:04:54      Path Set ID  : 5000002
      Interface      : Lspvif0
```

Step 2 **show ip pim neighbor** [**vrf** *vrf-name*] **neighbor** [*interface-type interface-number*]

Enter the **show ip pim neighbor**command to display PIM adjacencies information:

Example:

```
Device# show ip pim vrf VRF neighbor
192.168.10.18     Serial6/0           04:53:19/00:01:18 v2 1 / G
172.30.20.3       Lspvif0             04:52:32/00:01:28 v2 1 / B S P G
172.30.20.2       Lspvif0             04:52:32/00:01:17 v2 1 / B S P G
```

Step 3 **show ip mroute** [**vrf** *vrf-name*] [[**active** [*kbps*] [**interface** *type number*] | **bidirectional** | **count** [**terse**] | **dense** | **interface** *type number* | **proxy** | **pruned** | **sparse** | **ssm** | **static** | **summary**] | [*group-address* [*source-address*]] [**count** [**terse**] | **interface** *type number* | **proxy** | **pruned** | **summary**] | [*source-address group-address*] [**count** [**terse**] | **interface** *type number* | **proxy** | **pruned** | **summary**] | [*group-address*] **active** [*kbps*] [**interface** *type number* | **verbose**]]

Enter the **show ip mroute**command to display the contents of the multicast routing (mroute) table:

Example:

```
Device# show ip mroute vrf VRF 238.1.200.2 10.5.200.3
(10.5.200.3, 238.1.200.2), 04:54:18/00:02:40, flags: sT
Incoming interface: Lspvif0, RPF nbr 172.30.20.2
Outgoing interface list:
Serial6/0, Forward/Sparse-Dense, 04:54:18/00:02:40
```

Step 4 **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [**-** *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]

Enter the **show mpls forwarding-table**command to display the contents of the MPLS Label Forwarding Information Base (LFIB):

Example:

```
Device# show mpls forwarding-table | inc 1F000001
105    307         mLDP:1F000001     38468        Se5/0      point2point
       208         mLDP:1F000001     38468        Se4/0      point2point
109    307         mLDP:1F000001     34738        Se5/0      point2point
       408         mLDP:1F000001     34738        Se6/0      point2point
111    408         mLDP:1F000001     282          Se6/0      point2point
       208         mLDP:1F000001     282          Se4/0      point2point
```

Step 5 **show adjacency** [*ip-address*] [*interface-type interface-number* | **null** *number* | **port-channel** *number* | **sysclock** *number* | **vlan** *number* | **fcpa** *number* | **serial** *number*] [**connectionid** *number*] [**link** {**ipv4** | **mpls**}] [**detail** | **encapsulation**]

Enter the **show adjacency**command to display adjacency information for the specified LSP-VIF interface:

Example:

```
Device# show adjacency lspvif0
105    307         mLDP:1F000001     38468        Se5/0      point2point
       208         mLDP:1F000001     38468        Se4/0      point2point
109    307         mLDP:1F000001     34738        Se5/0      point2point
       408         mLDP:1F000001     34738        Se6/0      point2point
111    408         mLDP:1F000001     282          Se6/0      point2point
       208         mLDP:1F000001     282          Se4/0      point2point
```

Configuration Examples for MLDP-Based MVPN

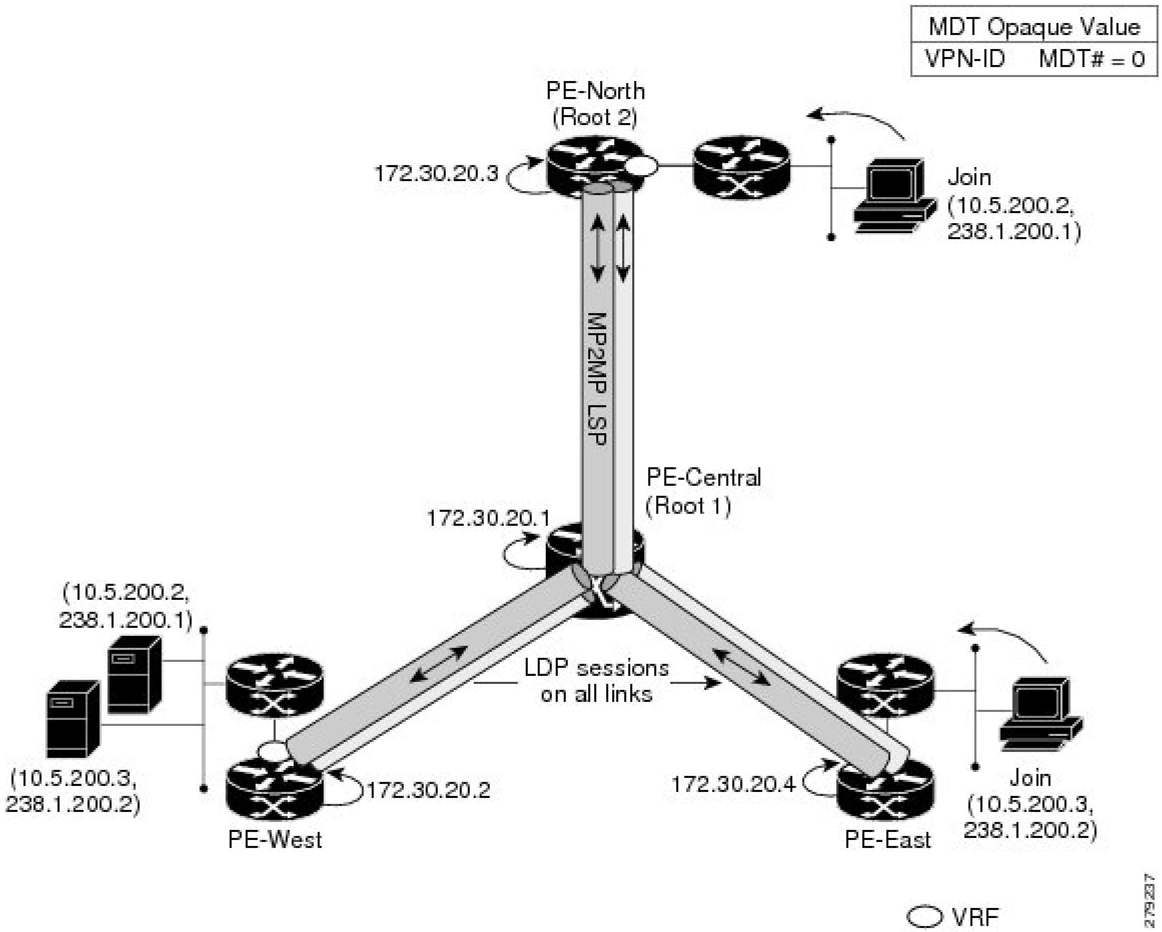
Example Initial Deployment of an MLDP-Based MVPN

Initial deployment of an MLDP-based MVPN involves the configuration of a default MDT and one or more data MDTs.

Default MDT Configuration

The following example shows how to configure the default MDT for an MLDP-based MVPN. This configuration is based on the sample topology illustrated in the figure.

Figure 55: Default MDT Example



This configuration is consistent for every PE device participating in the same VPN ID. The **vpn id 100:2** command replaces the MDT group address used with the mGRE transport method. To provide redundancy, two default MDT trees are statically configured, rooted at P-Central and PE-North. The selection as to which MP2MP tree the default MDT will use at a particular PE device is determined by Interior Gateway Protocol (IGP) metrics. An MP2MP LSP is implicit for the default MDT.

```
ip pim mpls source Loopback0
ip multicast-routing
ip multicast-routing vrf VRF
!
ip vrf VRF
 rd 100:2
 vpn id 100:2
 route-target export 200:2
 route-target import 200:2
 mdt default mpls mldp 172.30.20.1 (P-Central)
 mdt default mpls mldp 172.30.20.3 (PE-North)
```

PIM Adjacencies

PIM operates over the LSP-VIF as if it were a regular tunnel interface. That means PIM hellos are exchanged over the LSP-VIF to establish PIM adjacencies over the default MDT. The sample output in this section displays the three PIM adjacencies in VRF of PE-East. The first is the adjacency to the receiver network over serial interface 6/0, and the next two are the adjacencies to PE-West and PE-North over the MP2MP LSP via LSP-VIF interface 0.

```
PE-East# show ip pim vrf VRF neighbor
192.168.10.18     Serial6/0           04:53:19/00:01:18 v2 1 / G
172.30.20.3       Lspvif0             04:52:32/00:01:28 v2 1 / B S P G
172.30.20.2       Lspvif0             04:52:32/00:01:17 v2 1 / B S P G
```

The output from the **show ip mroute** command also shows the (S, G) entry for VRF. The stream 238.1.200.2 has the Reverse Path Forwarding (RPF) interface of LSP-VIF interface 0 and the neighbor 172.30.20.2, which is PE-West.

```
PE-East# show ip mroute vrf VRF 238.1.200.2 10.5.200.3
(10.5.200.3, 238.1.200.2), 04:54:18/00:02:40, flags: sT
Incoming interface: Lspvif0, RPF nbr 172.30.20.2
Outgoing interface list:
Serial6/0, Forward/Sparse-Dense, 04:54:18/00:02:40
```

MLDP Database Entry--PE-East

The sample output in this section displays the database entries for the MP2MP trees supporting the default MDT at PE-East. The database is searched by Opaque value MDT 100:2, which results in information for two MP2MP trees (one for each root) being returned. Both trees have different system IDs (2E000001, F2000005) and use the same Opaque value ([mdt 100:2 0]), but with different roots. The last 0 in the Opaque value indicates this tree is a default MDT. Entry 79000004 shows it is the primary MP2MP tree, therefore PE-East will transmit all source multicast traffic on this LSP, and B2000006 will be the backup root. Note that interface LSP-VIF interface 0 represents both MP2MP LSPs. The Local Label (D) is the downstream label allocated by PE-East for this tree. In other words, traffic from the root will be received with either label 408 (Primary Tree) or 407 (Backup Tree). The Out Label (U) is the label that PE-East will use to send traffic into the tree; upstream towards the root, either 105 for the Primary Tree or 108 for the Backup Tree. Both these labels were received from P-Central.

```
PE-East# show mpls mldp database opaque_type mdt 100:2
* Indicates MLDP recursive forwarding is enabled
LSM ID : 79000004 (RNR LSM ID: 8A000002)   Type: MP2MP   Uptime : 00:04:54
  FEC Root           : 172.30.20.1
  Opaque decoded     : [mdt 100:2 0]
  Opaque length      : 11 bytes
  Opaque value       : 07 000B 0000010000000100000000
  RNR active LSP     : (this entry)
  Upstream client(s) :
    172.30.20.1:0    [Active]
      Expires        : Never         Path Set ID  : 99000001
      Out Label (U)  : 32            Interface    : Ethernet1/0*
      Local Label (D): 30            Next Hop     : 10.0.1.7
  Replication client(s):
    MDT  (VRF VRF)
      Uptime         : 00:04:54      Path Set ID  : 5000002
      Interface      : Lspvif0
LSM ID : 79000005 (RNR LSM ID: 8A000003)   Type: MP2MP   Uptime : 00:04:54
  FEC Root           : 172.30.20.3
  Opaque decoded     : [mdt 100:2 0]
  Opaque length      : 11 bytes
  Opaque value       : 07 000B 0000010000000100000001
  RNR active LSP     : (this entry)
  Upstream client(s) :
    172.30.20.1:0    [Active]
```

```
      Expires        : Never         Path Set ID  : 99000002
      Out Label (U)  : 32            Interface    : Ethernet1/0*
      Local Label (D): 30            Next Hop     : 10.0.1.7
  Replication client(s):
    MDT  (VRF VRF)
      Uptime         : 00:04:54      Path Set ID  : 5000003
      Interface      : Lspvif0
```

Label Forwarding Entry--P-Central (Root 1)

The sample output shown in this section displays the VRF (MDT 100:2) MLDP database entry 1F000001 for the primary MP2MP LSP, which is P-Central. Because the local device P-Central is the root, there is no upstream peer ID, therefore no labels are allocated locally. However there are three replication clients, representing each of the three PE devices: PE-North, PE-West, and PE-East. These replication clients are the downstream nodes of the MP2MP LSP. These clients receive multipoint replicated traffic.

In the replication entry looking from the perspective of the root, there are two types of labels:

- Out label (D)--These are labels received from remote peers that are downstream to the root (remember traffic flows downstream away from the root).
- Local label (U)--These are labels provided by P-Central to its neighbors to be used as upstream labels (sending traffic to the root). It is easy to identify these labels as they all start in the 100 range, which we have configured for P-Central to use. P-Central sends these labels out when it receives a FEC with the type as MP2MP Down.

From the labels received and sent in the replication entries, the Label Forwarding Information Base (LFIB) is created. The LFIB has one entry per upstream path and one entry per downstream path. In this case because P-Central is the root, there are only upstream entries in the LFIB that have been merged with the corresponding downstream labels. For example, label 105 is the label P-Central sent to PE-East to send source traffic upstream. Traffic received from PE-East will then be replicated using the downstream labels 307 to PE-West and 208 to PE-North.

```
P-Central# show mpls mldp database opaque_type mdt 100:2
LSM ID : 79000006 (RNR LSM ID: 1F000001)   Type: MP2MP   Uptime : 00:04:54
  FEC Root           : 172.30.20.1
  Opaque decoded     : [mdt 100:2 0]
  Opaque length      : 11 bytes
  Opaque value       : 07 000B 0000010000000100000000
  RNR active LSP     : (this entry)
  Upstream client(s) : None
  Replication client(s):
    172.3.20.2:0
      Uptime         : 01:46:43      Path Set ID  : AC000008
      Out label (D)  : 208            Interface    : Serial4/0
      Local label (U): 109           Next Hop     : 172.30.10.2
    172.3.20.3:0
      Uptime         : 01:42:43      Path Set ID  : E00000C
      Out label (D)  : 307            Interface    : Serial5/0
      Local label (U): 111           Next Hop     : 172.30.10.6
    172.3.20.4:0
      Uptime         : 01:40:43      Path Set ID  : 3D000010
      Out label (D)  : 408            Interface    : Serial6/0
      Local label (U): 105           Next Hop     : 172.30.10.10
P-Central# show mpls forwarding-table | inc 1F000001
105    307          mLDP:1F000001     38468         Se5/0      point2point
       208          mLDP:1F000001     38468         Se4/0      point2point
109    307          mLDP:1F000001     34738         Se5/0      point2point
       408          mLDP:1F000001     34738         Se6/0      point2point
```

```
111    408          mLDP:1F000001     282           Se6/0      point2point
       208          mLDP:1F000001     282           Se4/0      point2point
```

The sample output shown in this section displays the entry on P-Central for the MP2MP LSP rooted at PE-North (backup root). In this tree P-Central is a branch of the tree, not a root, therefore there are some minor differences to note:

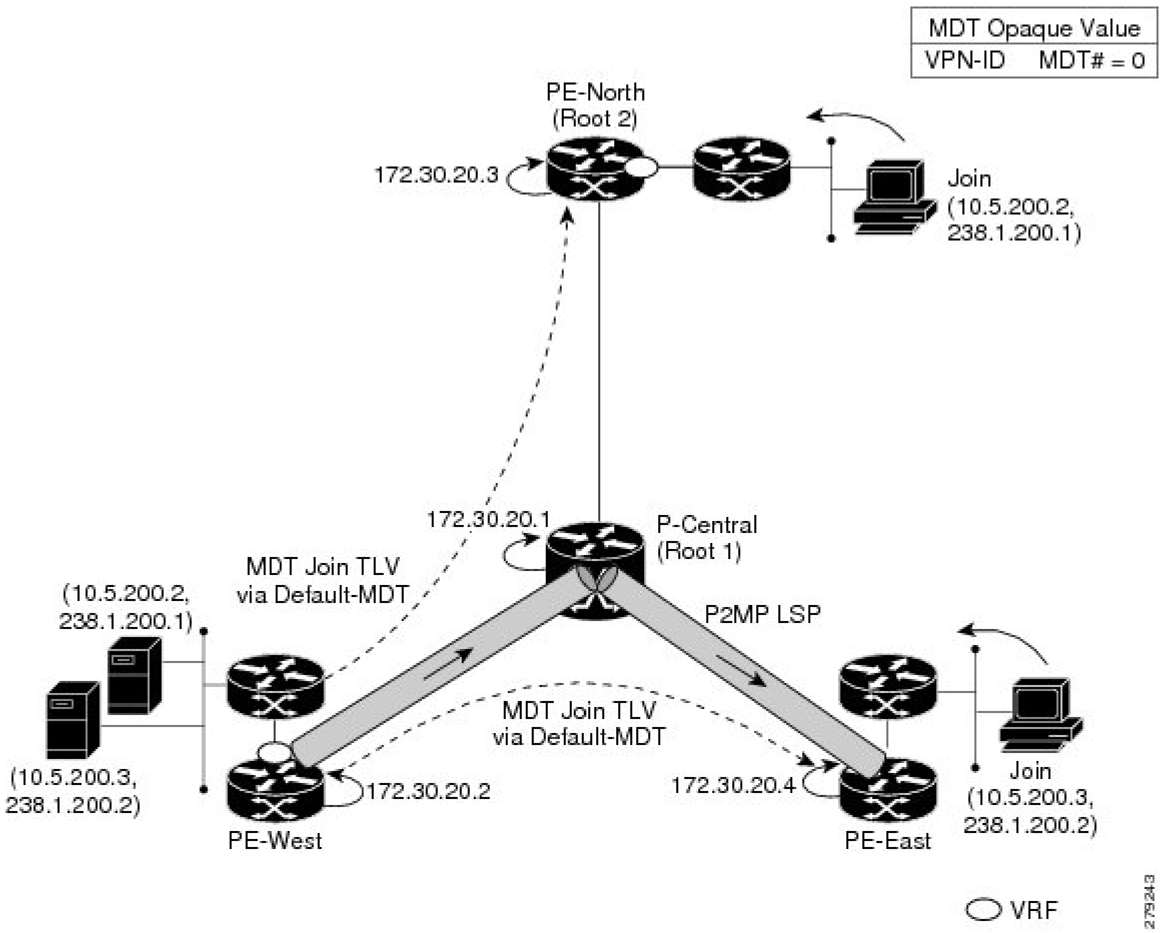
- The upstream peer ID is PE-North, therefore P-Central has allocated label 104 in the downstream direction towards PE-North and subsequently PE-North has responded with an upstream label of 313.
- Two replication entries representing PE-East and PE-West are displayed.
- The merged LFIB shows three entries:
  - One downstream entry label 104 receiving traffic from Root 2 (PE-North), which is then directed further downstream using labels 207 PE-West and 407 PE-East.
  - Two upstream entries 108 and 115 receiving traffic from the leaves and directing it either downstream 207, 407 or upstream using label 313.

```
Central_P# show mpls mldp database opaque_type mdt 100:2
LSM ID             : E6000004
 Uptime             : 00:42:03
 Tree type          : MP2MP
 FEC Root           : 172.30.20.3
 Opaque length      : 14 bytes
 Opaque value       : 07000B00 01000000 00020000 00009C
 Opaque decoded     : [mdt 100:2 0]
 Upstream peer ID   : 172.30.20.3:0, Label local (D): 104  remote (U): 313 active
 Path Set ID        : 48000003
 Replication client(s):
 172.30.20.2:0   uptime: 00:42:03    Path Set ID: CF000004
                 remote label (D): 207  local label (U): 115
                 nhop: 172.30.10.2 intrf: Serial4/0
 172.30.20.4:0   uptime: 00:41:44    Path Set ID: 5800000E
                 remote label (D): 407  local label (U): 108
                 nhop: 172.30.10.10 intrf: Serial6/0
Central_P# show mpls forwarding-table | inc E6000004
104    207          mLDP:E6000004     251228        Se4/0      point2point
       407          mLDP:E6000004     251334        Se6/0      point2point
108    207          mLDP:E6000004     0             Se4/0      point2point
       313          mLDP:E6000004     0             Se5/0      point2point
115    313          mLDP:E6000004     0             Se5/0      point2point
       407          mLDP:E6000004     0             Se6/0      point2point
```

Data MDT Configuration

The following example shows how to configure the data MDT for an MLDP-based MVPN. This configuration is based on the sample topology illustrated in the figure.

Figure 56: Data MDT Example



The sample output in this section displays the data MDT configuration for all the PE devices. The **mdt data** commands are the only additional commands necessary. The first **mdt data**command allows a maximum of 60 data MDTs to be created, and the second **mdt data**command sets the threshold. If the number of data MDTs exceeds 60, then the data MDTs will be reused in the same way as they are for the mGRE tunnel method (the one with the lowest reference count).

```
ip pim vrf VRF mpls source Loopback0
!
ip vrf VRF
 rd 100:2
 vpn id 100:2
 route-target export 200:2
 route-target import 200:2
 mdt default mpls mldp 172.30.20.1 (P-Central)
 mdt default mpls mldp 172.30.20.3 (PE-North)
 mdt data mpls mldp 60
 mdt data threshold 1
```

VRF mroute Table--PE-West

The sample output in this section displays the VRF mroute table on PE-West before the high-bandwidth source exceeds the threshold. At this point there are two streams, representing each of the two VPN sources at

PE-West, on a single MP2MP LSP (System ID D8000000). The LSP represents the default MDT accessed via LSP-VIF interface 0.

```
PE-West# show ip mroute vrf VRF verbose
.
.
.
(10.5.200.2, 238.1.200.1), 00:00:25/00:03:29, flags: sT
  Incoming interface: Serial6/0, RPF nbr 192.168.10.6
  Outgoing interface list:
    Lspvif0, LSM MDT: D8000000 (default),Forward/Sparse-Dense,
.
.
.
(10.5.200.3, 238.1.200.2), 00:11:14/00:02:48, flags: sT
  Incoming interface: Serial6/0, RPF nbr 192.168.10.6
  Outgoing interface list:
    Lspvif0, LSM MDT: D8000000 (default),Forward/Sparse-Dense,
.
.
.
```

The sample output in this section displays the output after the source transmission exceeds the threshold. PE-West sends an MDT Join TLV message to signal the creation of a data MDT. In this case, the data MDT number is 1, therefore PE-East will send a label mapping message back to PE-West with a FEC TLV containing root=PE-West, Opaque value=(mdt vpn-id 1). The System ID is now changed to 4E000003 signaling a different LSP; however, the LSP-VIF is still LSP-VIF interface 0. The (S, G) entry also has the “y” flag set indicating this stream has switched to a data MDT.

```
PE-West# show ip mroute vrf VRF 10.5.200.3 238.1.200.2 verbose
.
.
.
(10.5.200.3, 238.1.200.2), 00:00:08/00:03:27, flags: sTy
  Incoming interface: Serial6/0, RPF nbr 192.168.10.6
    MDT TX nr: 1 LSM-ID 4E000003
  Outgoing interface list:
    Lspvif0, LSM MDT: 4E000003 (data) Forward/Sparse-Dense,
```

LSP-VIF Adjacencies--PE-West

For the interface LSP-VIF, each virtual circuit represents a unique multipoint LSP forwarding instance. The correct adjacency is selected when sending the multicast packet. The sample output in this section displays the application of that concept on PE-West. There is a single LSP-VIF interface 0 interface, but it has three adjacencies as follows:

- 4E000003 is the single data MDT created for (10.5.200.3, 238.1.200.2)
- 58000000 is the default MDT (backup root)
- D8000000 is the default MDT (primary root)

```
PE-West# show adjacency lspvif 0

Protocol Interface                 Address
IP       Lspvif0                   4E000003(5)
IP       Lspvif0                   58000000(4)
IP       Lspvif0                   D8000000(3)
```

MLDP Database Entries

The sample output in this section displays the MLDP entry for the data MDT (4E000003) on the ingress device PE-West. The following points about this entry should be noted:

- The tree type is P2MP with PE-West (172.30.20.2) as the root.
- The Opaque value is [mdt 100:2 1] denoting the first data MDT.
- There are no labels allocated as it is the root.
- There are two replication client entries on this tree.
- Label 112 will be used to send the traffic downstream towards PE-East (via P-Central).
- The MDT entry is an internal construct.

```
PE-West# show mpls mldp database id 4E000003

LSM ID : 4E000003 (RNR LSM ID: 8A000002)   Type: P2MP   Uptime : 00:04:54
  FEC Root           : 172.30.20.2
  Opaque decoded     : [mdt 100:2 1]
  Opaque length      : 11 bytes
  Opaque value       : 07 000B 0000010000000100000000
  RNR active LSP     : (this entry)
  Upstream client(s) : None
  Replication client(s):
    MDT  (VRF VRF)
      Uptime         : 00:04:54      Path Set ID  : 5000002
      Interface      : Lspvif0
  172.30.20.1:0
      Uptime         : 01:41:43      Path Set ID  : D9000007
      Out label (D)  : 27            Interface    : Serial4/0
      Local label (U): 112           Next Hop     : 172.30.10.1
```

The sample output in this section displays the database entry for the data MDT on PE-East, the egress device. Also shown is the MDT Join TLV message that was sent from PE-West over the default MDT. The MDT Join TLV message contains all the necessary information to allow PE-East to create a label mapping message P2MP LSP back to the root of PE-West. Label 414 will be used by P-Central to send traffic to PE-East.

*Feb 19 04:43:24.039: PIM(1): MDT join TLV received for (10.5.200.3,238.1.200.2)

*Feb 19 04:43:24.039: MLDP: LDP root 172.30.20.2 added

*Feb 19 04:43:24.039: MLDP: [mdt 100:2 1] label mapping msg sent to 172.30.20.1:0

```
PE-East# show mpls mldp database opaque_type mdt 100:2 1
LSM ID : 4E000003 (RNR LSM ID: 8A000002)   Type: P2MP   Uptime : 00:04:54
  FEC Root           : 172.30.20.2
  Opaque decoded     : [mdt 100:2 1]
  Opaque length      : 11 bytes
  Opaque value       : 07 000B 0000010000000100000000
  RNR active LSP     : (this entry)
  Upstream client(s) : None
  Replication client(s):
    MDT  (VRF VRF)
      Uptime         : 00:04:54      Path Set ID  : 5000002
      Interface      : Lspvif0
```

LFIB Entry for the Data MDT

The sample output in this section displays the LFIB entry for the data MDT as it passes through P-Central and PE-East. The Tunnel ID used for the LSP is the Opaque value [mdt 100:2 1].

```
P-Central# show mpls for label 112
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
111        414        [mdt 100:2 1]    2993584       Se6/0      point2point
PE-East# show mpls for label 400

Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
414   [T]  No Label   [mdt 100:2 1][V] 3297312       aggregate/green
```

Example Migration from a PIM with mGRE-Based MVPN to an MLDP-Based MPVN

The following example shows an MLDP-based MVPN configuration that has been migrated from a PIM with mGRE based MVPN. The differences in the CLI from the PIM with mGRE-based MVPN are highlighted via comments below. In this example, MLDP derives the FEC from the import route target configured in the VRF.

```
ip vrf VRF
 rd 50:1111
 vpn id 50:10 ! MLDP-based MVPN configuration
 route-target export 100:100
 route-target import 100:100
 mdt preference mldp pim
 mdt default mpls mldp 1.1.1.1 ! MLDP-based MVPN configuration
 mdt default mpls mldp 2.2.2.2 ! MLDP-based MVPN configuration
 mdt data mpls mldp 255 ! MLDP-based MVPN configuration
 mdt data threshold 40 list 1 ! MLDP-based MVPN configuration
!
ip multicast-routing
ip multicast-routing vrf VRF
!
interface Loopback0
 ip address 205.1.0.1 255.255.255.0
 ip router isis
 ip pim sparse-dense-mode
!
interface Ethernet1/0
 ip vrf forwarding green
 ip address 220.0.2.1 255.255.255.0
 ip pim sparse-dense-mode
!
interface Ethernet2/0
 ip address 200.0.0.1 255.255.255.0
 ip pim sparse-dense-mode
 ip router isis
 mpls ip ! MLDP-based MVPN configuration
!
router isis
 net 49.0000.0000.0000.00
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS Multicast Command Reference
Overview of the IP multicast technology area	“ IP Multicast Technology Overview ” module
Concepts, tasks, and examples for configuring an IP multicast network using PIM	“ Configuring a Basic IP Multicast Network ” module

MIBs

MIB	MIBs Link
No new or modified MIBs are supported.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MLDP-Based MVPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 21: Feature Information for MLDP-Based MVPN

Feature Name	Releases	Feature Information
MLDP-Based MVPN	15.0(1)S 15.1(1)SY 15.4(1)T	The MLDP-based MVPN feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network (MVPN) core network. The following commands were introduced or modified: **debug mpls mldp all, debug mpls mldp filter opaque type, debug mpls mldp generic, debug mpls mldp gr, debug mpls mldp mfi, debug mpls mldp mrib, debug mpls mldp neighbor, debug mpls mldp packet, mdt data, mdt default, mdt preference, mpls mldp forwarding recursive, mpls logging notifications, mpls mldp path, show ip multicast mpls mrib-client, show ip multicast mpls vif, show mpls ldp discovery detailed, show mpls ldp bindings, show mpls mldp count, show mpls mldp database, show mpls mldp label release, show mpls mldp neighbors, show mpls mldp root.**

CHAPTER 33

IPv6 Multicast Listener Discovery Protocol

Information About IPv6 Multicast Listener Discovery Protocol

IPv6 Multicast Overview

An IPv6 multicast group is an arbitrary group of receivers that want to receive a particular data stream. This group has no physical or geographical boundaries--receivers can be located anywhere on the Internet or in any private network. Receivers that are interested in receiving data flowing to a particular group must join the group by signaling their local device. This signaling is achieved with the MLD protocol.

Devices use the MLD protocol to learn whether members of a group are present on their directly attached subnets. Hosts join multicast groups by sending MLD report messages. The network then delivers data to a potentially unlimited number of receivers, using only one copy of the multicast data on each subnet. IPv6 hosts that wish to receive the traffic are known as group members.

Packets delivered to group members are identified by a single multicast group address. Multicast packets are delivered to a group using best-effort reliability, just like IPv6 unicast packets.

The multicast environment consists of senders and receivers. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message.

A multicast address is chosen for the receivers in a multicast group. Senders use that address as the destination address of a datagram to reach all members of the group.

Membership in a multicast group is dynamic; hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group. A host can be a member of more than one multicast group at a time.

How active a multicast group is, its duration, and its membership can vary from group to group and from time to time. A group that has members may have no activity.

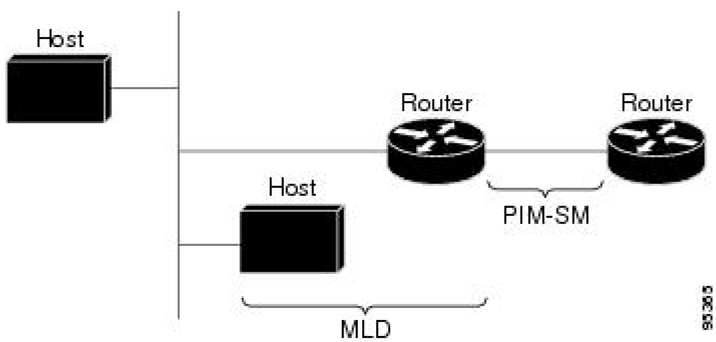
IPv6 Multicast Routing Implementation

Cisco software supports the following protocols to implement IPv6 multicast routing:

- MLD is used by IPv6 devices to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. There are two versions of MLD:
  - MLD version 1 is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4.
  - MLD version 2 is based on version 3 of the IGMP for IPv4.
- IPv6 multicast for Cisco software uses both MLD version 2 and MLD version 1. MLD version 2 is fully backward-compatible with MLD version 1 (described in RFC 2710). Hosts that support only MLD version 1 will interoperate with a device running MLD version 2. Mixed LANs with both MLD version 1 and MLD version 2 hosts are likewise supported.
- PIM-SM is used between devices so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- PIM in Source Specific Multicast (PIM-SSM) is similar to PIM-SM with the additional ability to report interest in receiving packets from specific source addresses (or from all but the specific source addresses) to an IP multicast address.

The figure below shows where MLD and PIM-SM operate within the IPv6 multicast environment.

Figure 57: IPv6 Multicast Routing Protocols Supported for IPv6



Multicast Listener Discovery Protocol for IPv6

To start implementing multicasting in the campus network, users must first define who receives the multicast. The MLD protocol is used by IPv6 devices to discover the presence of multicast listeners (for example, nodes that want to receive multicast packets) on their directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. It is used for discovering local group and source-specific group membership. The MLD protocol provides a means to automatically control and limit the flow of multicast traffic throughout your network with the use of special multicast queriers and hosts.

The difference between multicast queriers and hosts is as follows:

- A querier is a network device, such as a device, that sends query messages to discover which network devices are members of a given multicast group.
- A host is a receiver, including devices, that send report messages to inform the querier of a host membership.

A set of queriers and hosts that receive multicast data streams from the same source is called a multicast group. Queriers and hosts use MLD reports to join and leave multicast groups and to begin receiving group traffic.

MLD uses the Internet Control Message Protocol (ICMP) to carry its messages. All MLD messages are link-local with a hop limit of 1, and they all have the alert option set. The alert option implies an implementation of the hop-by-hop option header.

MLD has three types of messages:

- Query--General, group-specific, and multicast-address-specific. In a query message, the multicast address field is set to 0 when MLD sends a general query. The general query learns which multicast addresses have listeners on an attached link.

Group-specific and multicast-address-specific queries are the same. A group address is a multicast address.

- Report--In a report message, the multicast address field is that of the specific IPv6 multicast address to which the sender is listening.
- Done--In a done message, the multicast address field is that of the specific IPv6 multicast address to which the source of the MLD message is no longer listening.

An MLD report must be sent with a valid IPv6 link-local source address, or the unspecified address (::), if the sending interface has not yet acquired a valid link-local address. Sending reports with the unspecified address is allowed to support the use of IPv6 multicast in the Neighbor Discovery Protocol.

For stateless autoconfiguration, a node is required to join several IPv6 multicast groups in order to perform duplicate address detection (DAD). Prior to DAD, the only address the reporting node has for the sending interface is a tentative one, which cannot be used for communication. Therefore, the unspecified address must be used.

MLD states that result from MLD version 2 or MLD version 1 membership reports can be limited globally or by interface. The MLD group limits feature provides protection against denial of service (DoS) attacks caused by MLD packets. Membership reports in excess of the configured limits will not be entered in the MLD cache, and traffic for those excess membership reports will not be forwarded.

MLD provides support for source filtering. Source filtering allows a node to report interest in listening to packets only from specific source addresses (as required to support SSM), or from all addresses except specific source addresses sent to a particular multicast address.

When a host using MLD version 1 sends a leave message, the device needs to send query messages to reconfirm that this host was the last MLD version 1 host joined to the group before it can stop forwarding traffic. This function takes about 2 seconds. This "leave latency" is also present in IGMP version 2 for IPv4 multicast.

MLD Access Group

MLD access groups provide receiver access control in Cisco IPv6 multicast devices. This feature limits the list of groups a receiver can join, and it allows or denies sources used to join SSM channels.

How to Configure IPv6 Multicast Listener Discovery Protocol

Enabling IPv6 Multicast Routing

IPv6 multicast uses MLD version 2. This version of MLD is fully backward-compatible with MLD version 1 (described in *RFC 2710*). Hosts that support only MLD version 1 will interoperate with a device running MLD version 2. Mixed LANs with both MLD version 1 and MLD version 2 hosts are likewise supported.

Before you begin

You must first enable IPv6 unicast routing on all interfaces of the device on which you want to enable IPv6 multicast routing .

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast-routing** [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 multicast-routing** [**vrf** *vrf-name*] **Example:** `Device(config)# ipv6 multicast-routing`	Enables multicast routing on all IPv6-enabled interfaces and enables multicast forwarding for PIM and MLD on all enabled interfaces of the device. • IPv6 multicast routing is disabled by default when IPv6 unicast routing is enabled. IPv6 multicast-routing needs to be enabled for IPv6 multicast routing to function.

Customizing and Verifying MLD on an Interface

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface** *type number*
4. **ipv6 mld join-group** [*group-address*] [[**include** | **exclude**] {*source-address* | **source-list** [*acl*]}
5. **ipv6 mld access-group** *access-list-name*
6. **ipv6 mld static-group** [*group-address*] [[**include**| **exclude**] {*source-address* | **source-list** [*acl*]}
7. **ipv6 mld query-max-response-time** *seconds*
8. **ipv6 mld query-timeout** *seconds*
9. **ipv6 mld query-interval** *seconds*
10. **end**
11. **show ipv6 mld groups** [**link-local**] [*group-name* | *group-address*] [*interface-type interface-number*] [**detail** | **explicit**
12. **show ipv6 mfib summary**
13. **show ipv6 mld interface** [*type number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 mld join-group** [*group-address*] [[**include** | **exclude**] {*source-address* | **source-list** [*acl*]} **Example:** `Device(config-if)# ipv6 mld join-group FF04::12 exclude 2001:DB8::10::11`	Configures MLD reporting for a specified group and source.
Step 5	**ipv6 mld access-group** *access-list-name* **Example:** `Device(config-if)# ipv6 access-list acc-grp-1`	Allows the user to perform IPv6 multicast receiver access control.
Step 6	**ipv6 mld static-group** [*group-address*] [[**include**| **exclude**] {*source-address* | **source-list** [*acl*]} **Example:**	Statically forwards traffic for the multicast group onto a specified interface and cause the interface to behave as if a MLD joiner were present on the interface.

	Command or Action	Purpose
	`Device(config-if)# ipv6 mld static-group ff04::10 include 100::1`	
Step 7	**ipv6 mld query-max-response-time** *seconds* **Example:** `Device(config-if)# ipv6 mld query-max-response-time 20`	Configures the maximum response time advertised in MLD queries.
Step 8	**ipv6 mld query-timeout** *seconds* **Example:** `Device(config-if)# ipv6 mld query-timeout 130`	Configures the timeout value before the device takes over as the querier for the interface.
Step 9	**ipv6 mld query-interval** *seconds* **Example:** `Device(config-if)# ipv6 mld query-interval 60`	Configures the frequency at which the Cisco IOS XE software sends MLD host-query messages. **Caution** Changing this value may severely impact multicast forwarding.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.
Step 11	**show ipv6 mld groups** [**link-local**] [*group-name* | *group-address*] [*interface-type interface-number*] [**detail** | **explicit** **Example:** `Device# show ipv6 mld groups GigabitEthernet 2/1/0`	Displays the multicast groups that are directly connected to the device and that were learned through MLD.
Step 12	**show ipv6 mfib summary** **Example:** `Device# show ipv6 mfib summary`	Displays summary information about the number of IPv6 Multicast Forwarding Information Base (MFIB) entries (including link-local groups) and interfaces.
Step 13	**show ipv6 mld interface** [*type number* **Example:** `Device# show ipv6 mld interface GigabitEthernet 2/1/0`	Displays multicast-related information about an interface.

Disabling MLD Device-Side Processing

A user might only want specified interfaces to perform IPv6 multicast and will therefore want to turn off MLD device-side processing on a specified interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ipv6 mld router**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**no ipv6 mld router** **Example:** `Device(config-if)# no ipv6 mld router`	Disables MLD device-side processing on a specified interface.

Resetting the MLD Traffic Counters

SUMMARY STEPS

1. **enable**
2. **clear ipv6 mld** [**vrf** *vrf-name*] traffic
3. **show ipv6 mld** [**vrf** *vrf-name*] traffic

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**clear ipv6 mld** [**vrf** *vrf-name*] traffic **Example:** `Device# clear ipv6 mld traffic`	Resets all MLD traffic counters.
Step 3	**show ipv6 mld** [**vrf** *vrf-name*] traffic **Example:** `Device# show ipv6 mld traffic`	Displays the MLD traffic counters.

Clearing the MLD Interface Counters

SUMMARY STEPS

1. **enable**
2. **clear ipv6 mld** [**vrf** *vrf-name*] **counters** *interface-type*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 mld** [**vrf** *vrf-name*] **counters** *interface-type* **Example:** `Device# clear ipv6 mld counters GigabitEthernet1/0/0`	Clears the MLD interface counters.

Configuration Examples for IPv6 Multicast Listener Discovery Protocol

Example: Enabling IPv6 Multicast Routing

The following example enables multicast routing on all interfaces and also enables multicast forwarding for PIM and MLD on all enabled interfaces of the device.

```
Device> enable
Device# configure terminal
Device(config)# ipv6 multicast-routing
```

Example: Configuring the MLD Protocol

The following example shows how to configure the query maximum response time, the query timeout, and the query interval on GigabitEthernet interface 1/0/0:

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 1/0/0

Device(config-if)# ipv6 mld query-max-response-time  20

Device(config-if)# ipv6 mld query-timeout 130

Device(config-if)# ipv6 mld query-interval 60
```

The following example shows how to configure MLD reporting for a specified group and source, allows the user to perform IPv6 multicast receiver access control, and statically forwards traffic for the multicast group onto GigabitEthernet interface 1/0/0:

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 1/0/0
Device(config)# ipv6 mld join-group FF04::10
Device(config)# ipv6 mld static-group FF04::10 100::1
Device(config)# ipv6 mld access-group acc-grp-1
```

The following example shows information from the **show ipv6 mld interface** command for GigabitEthernet interface 2/1/0:

```
Device# show ipv6 mld interface GigabitEthernet 2/1/1

GigabitEthernet2/1/1 is up, line protocol is up
Internet address is FE80::205:5FFF:FEAF:2C39/10
MLD is enabled in interface
Current MLD version is 2
MLD query interval is 125 seconds
MLD querier timeout is 255 seconds
MLD max query response time is 10 seconds
Last member query response interval is 1 seconds
MLD activity: 25 joins, 17 leaves
MLD querying router is FE80::205:5FFF:FEAF:2C39 (this system)
```

The following example displays the MLD protocol messages received and sent:

```
Device# show ipv6 mld traffic

MLD Traffic Counters
Elapsed time since counters cleared:00:00:21

                              Received     Sent
Valid MLD Packets               3           1
Queries                         1           0
Reports                         2           1
Leaves                          0           0
Mtrace packets                  0           0

Errors:
Malformed Packets                           0
```

```
Bad Checksums                          0
Martian source                         0
Packets Received on MLD-disabled Interface  0
```

Example: Disabling MLD Router-Side Processing

The following example turns off MLD device-side processing on GigabitEthernet interface 1/0/0:

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 1/0/0

Device(config-if)# no ipv6 mld router
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

IPv6 Multicast Listener Discovery Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 22: Feature Information for IPv6 Multicast Listener Discovery Protocol

Feature Name	Releases	Feature Information
IPv6 Multicast: Multicast Listener Discovery (MLD) Protocol, Versions 1 and 2	12.0(26)S 12.2(18)S 12.2(25)SG 12.2(33)SRA 12.3(2)T 15.0(1)S Cisco IOS XE Release 2.1	MLD is used by IPv6 routers to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. The following commands were introduced or modified: **debug ipv6 mld**, **ipv6 mld join-group**, **ipv6 mld static-group**, **ipv6 mld query-interval**, **ipv6 mld query-max-response-time**, **ipv6 mld query-timeout**, **ipv6 mld router**, **show ipv6 mld groups**, **show ipv6 mld groups summary**, **show ipv6 mld interface**.
IPv6 Multicast: MLD Access Group	12.2(33)SRE 12.2(50)SY 12.4(2)T 15.0(1)S Cisco IOS XE Release 2.1	The MLD access group provides receiver access control in Cisco IPv6 multicast routers. The following command was introduced: **ipv6 mld access-group**.

CHAPTER 34

MLD Group Limits

The IPv6 Multicast Listener Discovery (MLD) group limits feature provides global and per-interface MLD join limits.

Information About MLD Group Limits

Multicast Listener Discovery Protocol for IPv6

To start implementing multicasting in the campus network, users must first define who receives the multicast. The MLD protocol is used by IPv6 devices to discover the presence of multicast listeners (for example, nodes that want to receive multicast packets) on their directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. It is used for discovering local group and source-specific group membership. The MLD protocol provides a means to automatically control and limit the flow of multicast traffic throughout your network with the use of special multicast queriers and hosts.

The difference between multicast queriers and hosts is as follows:

- A querier is a network device, such as a device, that sends query messages to discover which network devices are members of a given multicast group.
- A host is a receiver, including devices, that send report messages to inform the querier of a host membership.

A set of queriers and hosts that receive multicast data streams from the same source is called a multicast group. Queriers and hosts use MLD reports to join and leave multicast groups and to begin receiving group traffic.

MLD uses the Internet Control Message Protocol (ICMP) to carry its messages. All MLD messages are link-local with a hop limit of 1, and they all have the alert option set. The alert option implies an implementation of the hop-by-hop option header.

MLD has three types of messages:

- Query--General, group-specific, and multicast-address-specific. In a query message, the multicast address field is set to 0 when MLD sends a general query. The general query learns which multicast addresses have listeners on an attached link.

Group-specific and multicast-address-specific queries are the same. A group address is a multicast address.

- Report--In a report message, the multicast address field is that of the specific IPv6 multicast address to which the sender is listening.
- Done--In a done message, the multicast address field is that of the specific IPv6 multicast address to which the source of the MLD message is no longer listening.

An MLD report must be sent with a valid IPv6 link-local source address, or the unspecified address (::), if the sending interface has not yet acquired a valid link-local address. Sending reports with the unspecified address is allowed to support the use of IPv6 multicast in the Neighbor Discovery Protocol.

For stateless autoconfiguration, a node is required to join several IPv6 multicast groups in order to perform duplicate address detection (DAD). Prior to DAD, the only address the reporting node has for the sending interface is a tentative one, which cannot be used for communication. Therefore, the unspecified address must be used.

MLD states that result from MLD version 2 or MLD version 1 membership reports can be limited globally or by interface. The MLD group limits feature provides protection against denial of service (DoS) attacks caused by MLD packets. Membership reports in excess of the configured limits will not be entered in the MLD cache, and traffic for those excess membership reports will not be forwarded.

MLD provides support for source filtering. Source filtering allows a node to report interest in listening to packets only from specific source addresses (as required to support SSM), or from all addresses except specific source addresses sent to a particular multicast address.

When a host using MLD version 1 sends a leave message, the device needs to send query messages to reconfirm that this host was the last MLD version 1 host joined to the group before it can stop forwarding traffic. This function takes about 2 seconds. This "leave latency" is also present in IGMP version 2 for IPv4 multicast.

How to Implement MLD Group Limits

Implementing MLD Group Limits Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 mld** [**vrf** *vrf-name*] **state-limit** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 mld** [**vrf** *vrf-name*] **state-limit** *number* **Example:** `Device(config)# ipv6 mld state-limit 300`	Limits the number of MLD states globally.

Implementing MLD Group Limits per Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 mld limit** *number* [**except** *access-list*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**ipv6 mld limit** *number* [**except** *access-list* **Example:** `device(config-if)# ipv6 mld limit 100`	Limits the number of MLD states on a per-interface basis.

Configuration Examples for MLD Group Limits

Example: Implementing MLD Group Limits

This example shows the groups and channels that are being accounted when the MLD group limit function is active:

```
Device# show ipv6 mld groups FF03::1 detail

Interface: FastEthernet5/1
Group:  FF03::1
Uptime:  00:00:05
Router mode: EXCLUDE (Expires: 00:04:14)
Host mode: INCLUDE
Last reporter: FE80::20A:8BFF:FE4D:6039
State accounted
Source list is empty

Interface: FastEthernet5/1
Group:  FF33::1
Uptime:  00:00:03
Router mode: INCLUDE
Host mode: INCLUDE
Last reporter: FE80::20A:8BFF:FE4D:6039
Group source list:
Source Address                           Uptime    Expires   Fwd  Flags
2001:DB8:0::1                                    00:00:03  00:04:16  Yes  Remote Ac 4
```

The following example shows all of the groups joined by Fast Ethernet interface 2/1, including link-local groups used by network protocols.

```
Device# show ipv6 mld groups FastEthernet 2/1

MLD Connected Group Membership
Group Address          Interface           Uptime        Expires
FF02::2                FastEthernet2/1     3d18h         never
FF02::D                FastEthernet2/1     3d18h         never
FF02::16               FastEthernet2/1     3d18h         never
FF02::1:FF00:1         FastEthernet2/1     3d18h         00:00:27
FF02::1:FF00:79        FastEthernet2/1     3d18h         never
FF02::1:FF23:83C2      FastEthernet2/1     3d18h         00:00:22
FF02::1:FFAF:2C39      FastEthernet2/1     3d18h         never
FF06:7777::1           FastEthernet2/1     3d18h         00:00:26
```

The following is sample output from the **show ipv6 mld groups summary** command:

```
Device# show ipv6 mld groups summary


MLD Route Summary
  No. of (*,G) routes = 5
  No. of (S,G) routes = 0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MLD Group Limits

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 23: Feature Information for MLD Group Limits

Feature Name	Releases	Feature Information
MLD Group Limits	12.2(33)SRE 12.2(50)SY 12.4(2)T 15.0(1)S 15.0(1)SY 15.1(1)SY Cisco IOS XE Release 2.6	The IPv6 MLD group limits feature provides global and per-interface MLD join limits. The following commands were introduced or modified: **ipv6 mld limit**, **ipv6 mld state-limit**.

CHAPTER 35

MLDP In-Band Signaling/Transit Mode

This module contains information for configuring Multicast Label Distribution Protocol (MLDP) in-band signaling to enable the MLDP core to create (S,G) or (*,G) state without using out-of-band signaling such as Border Gateway protocol (BGP) or Protocol Independent Multicast (PIM).

Restrictions for MLDP In-Band Signaling

- MLDP in-band signaling supports SOURCE-SPECIFIC MULTICAST (SSM) multicast traffic only.
- MLDP in-band signaling is not supported in the same VRF for which Rosen Model MLDP-based MVPN or GRE-based MVPN is configured.

Information About MLDP In-Band Signaling/Transit Mode

MLDP In-Band Signaling/Transit Mode

Multicast Label Distribution Protocol (MLDP)-supported multicast VPN (MVPN) allows VPN multicast streams to be aggregated over a VPN-specific tree. No customer state is created in the MLDP core;, there is only state for default and data multicast distribution trees (MDTs). In certain scenarios, the state created for VPN streams is limited and does not appear to be a risk or limiting factor. In these scenarios, MLDP can build in-band MDTs that are transit Label Switched Paths (LSPs).

Trees used in a VPN space are MDTs. Trees used in the global table are transit point-to-multipoint (P2MP) or multipoint-to-multipoint (MP2MP) LSPs. In both cases, a single multicast stream (VPN or not) is associated with a single LSP in the MPLS core. The stream information is encoded in the Forwarding Equivalence Class (FEC) of the LSP. This is in-band signaling.

MLDP in-band signaling uses access control lists (ACLs) with the range of the multicast (S, G) to be transported by the MLDP LSP. Each multicast channel (S, G) maps, one-to-one, to each tree in the in-band tree. The (S,G) join is registered in the Multicast Routing Information Base (MRIB), which is a client of MLDP. Each MLDP LSP is identified by the FEC of [(S,G) + RD], where RD is the Route Distinquisher (RD) obtained from BGP. This differs from MLDP-based MVPN, where the identity is in a FEC of [MDT #, VPN ID, Tree #]).

The ingress Provider Edge (PE) device uses the FEC to decode the stream information and associate the multicast stream with the LSP (in the FEC). This service model is only applicable for transporting Protocol Independent Multicast (PIM) source-specific multicast (SSM) traffic. There is no need to run PIM over the LSP because the stream signaling is done in-band.

The MLDP In-Band Signaling/Transit Mode feature is supported on IPv4 networks. MLDP in-band signaling and MLDP-based MVPN cannot be supported in the same VRF.

How to Configure MLDP In-Band Signaling/Transit Mode

Enabling In-Band Signaling on a PE Device

Before you begin

- VRF instances for in-band signaling must be configured.
- Access control lists (ACLs) for controlling streams must be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Use one of the following commands:
   - **ip multicast** [**vrf** *vrf*] **mpls mldp** [**range** *acl*]
   - **ipv6 multicast** [**vrf** *vrf*] **mpls mldp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose

Step 3	Use one of the following commands:
	• **ip multicast** [**vrf** *vrf*] **mpls mldp** [**range** *acl*]
	• **ipv6 multicast** [**vrf** *vrf*] **mpls mldp**
	Example:
	`Device (config)# ip multicast vrf vrf1 mpls mldp`
	`Device (config)# ipv6 multicast vrf vrf1 mpls mldp`
		Brings up the MLDP MRIB process and registers MLDP with the MRIB.
		• To enable in-band signaling globally, use this command without the **vrf** *vrf* keyword and argument combination.
		• IPv4 only: To identify streams for in-band signaling, use this command with the **range** keyword on the egress PE.

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Configuration Examples for MLDP In-Band Signaling/Transit Mode

Example: In-Band Signaling on PE1

```
PE1# show running-config
Building configuration...

Current configuration : 8247 bytes
!
! Last configuration change at 12:44:13 IST Thu Nov 15 2012
!

hostname PE1
!
mls ipv6 vrf
!
vrf definition vrf1
 rd 1:1
 vpn id 1:1
 route-target export 1:1
 route-target import 1:1
 !
 address-family ipv4
  route-target export 1:1
  route-target import 1:1
 exit-address-family
 !
 address-family ipv6
  route-target export 1:1
  route-target import 1:1
 exit-address-family
!

ip multicast-routing
ip multicast-routing vrf vrf1
ip multicast hardware-switching replication-mode egress
ip multicast mpls mldp
ip multicast vrf vrf1 mpls mldp
!
!
!
ipv6 unicast-routing
```

```
ipv6 multicast-routing
ipv6 multicast-routing vrf vrf1
ipv6 multicast rpf use-bgp
ipv6 multicast mpls source Loopback0
ipv6 multicast mpls mldp
ipv6 multicast vrf vrf1 rpf use-bgp
ipv6 multicast vrf vrf1 mpls source Loopback0
ipv6 multicast vrf vrf1 mpls mldp
!
!
vtp domain cisco
vtp mode off
mpls label protocol ldp
mpls ldp graceful-restart
mls flow ip interface-full
no mls flow ipv6
mls rate-limit multicast ipv4 igmp 100 10
mls cef error action reset
mls mpls tunnel-recir
multilink bundle-name authenticated
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
no diagnostic bootup level
!
redundancy
 main-cpu
  auto-sync running-config
 mode sso
!

vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
 ip pim sparse-mode
 ip ospf 100 area 0
 ipv6 address 1::1:1:1/64
 ipv6 enable
!
.
.
.
!
interface GigabitEthernet2/0/0.1
 encapsulation dot1Q 2
 vrf forwarding vrf1
 ip address 192.0.2.1 255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
 ipv6 address FE80::10:1:1 link-local
 ipv6 address 2001:DB8::/64
 ipv6 enable
!
interface GigabitEthernet2/0/0.2000
 encapsulation dot1Q 2000
 ip address 192.0.2.2 255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
 ipv6 address 2001:DB8:0:1/64
```

```
 ipv6 enable
!
.
.
.
interface GigabitEthernet2/0/12
 ip address 192.0.2.3 255.255.255.0
 ip pim sparse-mode
 ip ospf 100 area 0
 ipv6 address 2001:DB8::/64
 ipv6 enable
 mpls ip
 mpls label protocol ldp
 no mls qos trust
!
!
!
router ospf 100
 router-id 1.1.1.1
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 4.4.4.4 remote-as 100
 neighbor 4.4.4.4 update-source Loopback0
 !
 address-family ipv4
  redistribute static
  redistribute connected
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community both
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community both
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 send-community both
 exit-address-family
 !
 address-family vpnv4
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community extended
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 send-community extended
 exit-address-family
 !
 address-family ipv4 mdt
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community extended
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 send-community extended
 exit-address-family
 !
 address-family ipv6
  redistribute connected
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community extended
  neighbor 2.2.2.2 send-label
  neighbor 3.3.3.3 activate
```

```
  neighbor 3.3.3.3 send-community extended
  neighbor 3.3.3.3 send-label
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 send-community extended
  neighbor 4.4.4.4 send-label
 exit-address-family
 !
 address-family vpnv6
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community extended
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vrf1
  redistribute connected
 exit-address-family
 !
 address-family ipv6 vrf vrf1
  redistribute connected
 exit-address-family
!
no ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim ssm default
ip pim mpls source Loopback0
ip pim vrf vrf1 ssm default
ip pim vrf vrf1 mpls source Loopback0
ip route 192.0.2.25 255.255.255.255 7.37.0.1
!
!
mpls ldp router-id Loopback0 force
!
!
!
end
```

Example: In-Band Signaling on PE2

```
PE2# show running-config
Building configuration...

Current configuration : 7609 bytes
!
! Last configuration change at 13:18:45 IST Thu Nov 15 2012
!
hostname PE2
!
mls ipv6 vrf
!
vrf definition vrf1
 rd 1:1
 vpn id 1:1
 route-target export 1:1
 route-target import 1:1
 !
 address-family ipv4
```

```
  route-target export 1:1
  route-target import 1:1
 exit-address-family
 !
 address-family ipv6
  route-target export 1:1
  route-target import 1:1
 exit-address-family
!
.
.
.
!
ip multicast-routing
ip multicast-routing vrf vrf1
ip multicast hardware-switching replication-mode egress
ip multicast mpls mldp
ip multicast vrf vrf1 mpls mldp
!
!
!
ipv6 unicast-routing
ipv6 multicast-routing
ipv6 multicast-routing vrf vrf1
ipv6 multicast rpf use-bgp
ipv6 multicast mpls source Loopback0
ipv6 multicast mpls mldp
ipv6 multicast vrf vrf1 rpf use-bgp
ipv6 multicast vrf vrf1 mpls source Loopback0
ipv6 multicast vrf vrf1 mpls mldp
!
!
vtp domain isbu-devtest
vtp mode off
mpls label protocol ldp
mpls ldp graceful-restart
mls flow ip interface-full
no mls flow ipv6
mls cef error action reset
multilink bundle-name authenticated
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
diagnostic bootup level minimal
!
redundancy
 main-cpu
  auto-sync running-config
 mode sso
!
!
!
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
 ip pim sparse-mode
 ip ospf 100 area 0
 ipv6 enable
!
.
.
.
```

```
!
interface GigabitEthernet3/0/3.1
 encapsulation dot1Q 2
 vrf forwarding vrf1
 ip address 192.0.2.1 255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
 ipv6 address FE80::30:1:1 link-local
 ipv6 address 2001:DB8::/64
 ipv6 enable
!
interface GigabitEthernet3/0/3.2000
 encapsulation dot1Q 2000
 ip address 192.0.2.2 255.255.255.0
 ip pim sparse-mode
 ip igmp static-group 232.1.1.1 source 50.0.0.2
 ip igmp version 3
 ipv6 address 2001:DB8:0:1/64
 ipv6 enable
!
.
.
.
!
interface GigabitEthernet4/15
 ip address 192.0.2.3 255.255.255.0
 ip pim sparse-mode
 ip ospf 100 area 0
 ipv6 address 2001:DB8::/64
 ipv6 enable
 mpls ip
 mpls label protocol ldp
!
.
.
.
!
interface Vlan1
 no ip address
 shutdown
!
router ospf 100
 router-id 4.4.4.4
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 100
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback0
 !
 address-family ipv4
  redistribute static
  redistribute connected
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community both
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community both
 exit-address-family
 !
```

```
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community extended
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community extended
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
 exit-address-family
 !
 address-family ipv4 mdt
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community extended
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community extended
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
 exit-address-family
 !
 address-family ipv6
  redistribute connected
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community extended
  neighbor 1.1.1.1 send-label
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community extended
  neighbor 2.2.2.2 send-label
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
  neighbor 3.3.3.3 send-label
 exit-address-family
 !
 address-family vpnv6
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community extended
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community extended
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vrf1
  redistribute connected
 exit-address-family
 !
 address-family ipv6 vrf vrf1
  redistribute connected
 exit-address-family
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim ssm default
ip pim mpls source Loopback0
ip pim vrf vrf1 ssm default
ip pim vrf vrf1 mpls source Loopback0
ip route 192.0.2.25 255.255.255.255 7.37.0.1
!
!
mpls ldp router-id Loopback0 force
!
!
.
.
```

```
.
!
!
end
```

Feature Information for MLDP In-Band Signaling/Transit Mode

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 24: Feature Information for MLDP In-Band Signaling/Transit Mode

Feature Name	Releases	Feature Information
MLDP In-Band Signaling/Transit Mode	15.3(1)S Cisco IOS XE 3.14S	Multicast Label Distribution Protocol (MLDP) in-band signaling supports point-to-multipoint (P2P) and multipoint-to-multipoint (MP2MP) Label Switched Paths (LSPs) and enables the MLDP core to create (S,G) or (*,G) state without using out-of-band signaling such as Border Gateway Protocol (BGP) or Protocol Independent Multicast (PIM). This feature is supported for IPv4 and IPv6 multicast groups. The following commands were introduced or modified: **ip multicast mpls mldp**, **ipv6 multicast mpls mldp**.

CHAPTER 36

HA Support for MLDP

The HA Support for MLDP feature enables Cisco Multicast Label Distribution Protocol (MLDP) to checkpoint sufficient signaling and forwarding information for repopulating the necessary database on a dual Route Processor (RP) platform on which Stateful Switchover/Nonstop Forwarding (SSO/NSF) and Label Distribution Protocol (LDP) Graceful Restart are configured, after a switchover.

Prerequisites for HA Support for MLDP

- Stateful Switchover/Nonstop Forwarding (SSO/NSF) and LDP Graceful Restart must be configured on the dual Route Processor (RP) platform.
- LDP Graceful Restart must be configured on the NSF router peers.
- The Cisco IOS release software installed on the active and standby RPs must support MLDP-based MVPN and HA Support for MLDP.

Restrictions for HA Support for MLDP

- If Label Distribution Protocol (LDP) Graceful Restart is not enabled on the dual Route Processor (RP) platform, Nonstop Forwarding (NSF) peers will remove existing forwarding and label information from their Multicast Label Distribution Protocol (MLDP) database entries immediately following a switchover.

Information About HA Support for MLDP

The HA Support for MLDP feature enables MLDP to checkpoint label forwarding or path set information. To support NSF, MLDP uses existing PIM HA architecture to checkpoint the information to the standby RP.

MDT data group creation is a dynamic event triggered by traffic exceeding a specified threshold. When the threshold is exceeded (requiring an MDT data group to be created) or when traffic falls below the threshold (requiring the MDT data group to be deleted), the router detecting the event creates, deletes, or updates an MDT data "send" entry, creates the corresponding (S,G) state, if necessary, and sends a message to PE peers to create, delete, or update a corresponding MDT data "receive" entry and the corresponding (S,G) state.

The active RP will checkpoint the current state of the MLDP peer, paths to peers, root, paths to root, and the database and replication/branch entry onto the standby RP and use this state to recreate the MLDP state after a switchover.

How to Montior HA Support for MLDP

Displaying Check Pointed Information

SUMMARY STEPS

1. **enable**
2. **show mpls mldp ha database**
3. **show mpls mldp ha database summary**
4. **show mpls mldp ha neighbor**
5. **show mpls mldp ha root**
6. **show mpls mldp ha count**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `PE2> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls mldp ha database** **Example:** `PE2# show mpls mldp ha database`	Displays checkpoint data information.
Step 3	**show mpls mldp ha database summary** **Example:** `PE2# show mpls mldp ha database summary`	Displays synched database information only.
Step 4	**show mpls mldp ha neighbor** **Example:** `PE2# show mpls mldp ha neighbor`	Displays information about synched peers.

	Command or Action	Purpose
Step 5	**show mpls mldp ha root** **Example:** `PE2# show mpls mldp ha root`	Displays synched root information.
Step 6	**show mpls mldp ha count** **Example:** `PE2# show mpls mldp ha count`	Displays number of trees.

Displaying Data MDT Mappings for MLDP on Standby Device

SUMMARY STEPS

1. **enable**
2. **show ip pim vrf** *vrf* **mdt send**
3. **show ip pim vrf** *vrf* **mdt recv**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `PE1-standby> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip pim vrf** *vrf* **mdt send** **Example:** `PE1-standby# show ip pim vrf blue mdt send`	Displays data MDT mappings for MLDP.
Step 3	**show ip pim vrf** *vrf* **mdt recv** **Example:** `PE1-standby# show ip pim vrf blue mdt recv`	Displays data MDT mappings for MLDP.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP Multicast commands	Cisco IOS IP Multicast Command Reference
Cisco HA	*High Availability Configuration Guide*

Standards and RFCs

Standard/RFC	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

MIBs

MIB	MIBs Link
--	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for HA Support for MLDP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 25: Feature Information for HA Support for MLDP

Feature Name	Releases	Feature Information
HA Support for MLDP	15.1(3)S 15.1(1)SY Cisco IOS XE Release 3.8S	The HA Support for MLDP feature enables Cisco Multicast Label Distribution Protocol (MLDP) to checkpoint sufficient signaling and forwarding information for repopulating the necessary database on a dual Route Processor (RP) platform on which Stateful Switchover/Nonstop Forwarding (SSO/NSF) and Label Distribution Protocol (LDP) Graceful Restart are configured, after a switchover. The following commands were introduced or modified: **show ip pim mdt recv**, **show ip pim mdt send**, **show mpls mldp ha database** , **show mpls mldp ha neighbor**, **show mpls mldp ha root**.

PART III

IGMP

CHAPTER 37

Customizing IGMP

Internet Group Management Protocol (IGMP) is used to dynamically register individual hosts in a multicast group on a particular LAN segment. Enabling Protocol Independent Multicast (PIM) on an interface also enables IGMP operation on that interface.

This module describes ways to customize IGMP, including how to:

- Configure the router to forward multicast traffic in the absence of directly connected IGMP hosts.
- Enable an IGMP Version 3 (IGMPv3) host stack so that the router can function as a multicast network endpoint or host.
- Enable routers to track each individual host that is joined to a particular group or channel in an IGMPv3 environment.
- Control access to an SSM network using IGMP extended access lists.
- Configure an IGMP proxy that enables hosts that are not directly connected to a downstream router to join a multicast group sourced from an upstream network.

Prerequisites for IGMP

- Before performing the tasks in this module, you should be familiar with the concepts explained in the "IP Multicast Routing Technology Overview" module.
- The tasks in this module assume that IP multicast has been enabled and that the Protocol Independent Multicast (PIM) interfaces have been configured using the tasks described in the "Configuring IP Multicast Routing” module.

Restrictions for Customizing IGMP

Traffic Filtering with Multicast Groups That Are Not Configured in SSM Mode

IGMPv3 membership reports are not utilized by the software to filter or restrict traffic for multicast groups that are not configured in Source Specific Multicast (SSM) mode. Effectively, Cisco IOS software interprets all IGMPv3 membership reports for groups configured in dense, sparse, or bidirectional mode to be group membership reports and forwards traffic from all active sources onto the network.

Interoperability with IGMP Snooping

You must be careful when using IGMPv3 with switches that support and are enabled for IGMP snooping, because IGMPv3 messages are different from the messages used in IGMP Version 1 (IGMPv1) and Version 2 (IGMPv2). If a switch does not recognize IGMPv3 messages, then hosts will not correctly receive traffic if IGMPv3 is being used. In this case, either IGMP snooping may be disabled on the switch or the router may be configured for IGMPv2 on the interface, which would remove the ability to use SSM for host applications that cannot resort to URL Rendezvous Directory (URD) or IGMP v3lite.

Interoperability with CGMP

Networks using Cisco Group Management Protocol (CGMP) will have better group leave behavior if they are configured with IGMPv2 than IGMPv3. If CGMP is used with IGMPv2 and the switch is enabled for the CGMP leave functionality, then traffic to a port joined to a multicast group will be removed from the port shortly after the last member on that port has dropped membership to that group. This fast-leave mechanism is part of IGMPv2 and is specifically supported by the CGMP fast-leave enabled switch.

With IGMPv3, there is currently no CGMP switch support of fast leave. If IGMPv3 is used in a network, CGMP will continue to work, but CGMP fast-leave support is ineffective and the following conditions apply:

- Each time a host on a new port of the CGMP switch joins a multicast group, that port is added to the list of ports to which the traffic of this group is sent.
- If all hosts on a particular port leave the multicast group, but there are still hosts on other ports (in the same virtual LAN) joined to the group, then nothing happens. In other words, the port continues to receive traffic from that multicast group.
- Only when the last host in a virtual LAN (VLAN) has left the multicast group does forwarding of the traffic of this group into the VLAN revert to no ports on the forwarding switch.

This join behavior only applies to multicast groups that actually operate in IGMPv3 mode. If legacy hosts only supporting IGMPv2 are present in the network, then groups will revert to IGMPv2 and fast leave will work for these groups.

If fast leave is needed with CGMP-enabled switches, we recommend that you not enable IGMPv3 but configure IGMPv2 on that interface.

If you want to use SSM, you need IGMPv3 and you have two configuration alternatives, as follows:

- Configure only the interface for IGMPv2 and use IGMP v3lite and URD.
- Enable IGMPv3 and accept the higher leave latencies through the CGMP switch.

Information About Customizing IGMP

Role of the Internet Group Management Protocol

IGMP is used to dynamically register individual hosts in a multicast group on a particular LAN. Enabling PIM on an interface also enables IGMP. IGMP provides a means to automatically control and limit the flow of multicast traffic throughout your network with the use of special multicast queriers and hosts.

- A querier is a network device, such as a router, that sends query messages to discover which network devices are members of a given multicast group.
- A host is a receiver, including routers, that sends report messages (in response to query messages) to inform the querier of a host membership. Hosts use IGMP messages to join and leave multicast groups.

Hosts identify group memberships by sending IGMP messages to their local multicast device. Under IGMP, devices listen to IGMP messages and periodically send out queries to discover which groups are active or inactive on a particular subnet.

IGMP Versions Differences

There are three versions of IGMP, as defined by Request for Comments (RFC) documents of the Internet Engineering Task Force (IETF). IGMPv2 improves over IGMPv1 by adding the ability for a host to signal desire to leave a multicast group and IGMPv3 improves over IGMPv2 mainly by adding the ability to listen to multicast originating from a set of source IP addresses only.

Table 26: IGMP Versions

IGMP Version	Description
IGMPv1	Provides the basic query-response mechanism that allows the multicast device to determine which multicast groups are active and other processes that enable hosts to join and leave a multicast group. RFC 1112 defines the IGMPv1 host extensions for IP multicasting.
IGMPv2	Extends IGMP, allowing such capabilities as the IGMP leave process, group-specific queries, and an explicit maximum response time field. IGMPv2 also adds the capability for devices to elect the IGMP querier without dependence on the multicast protocol to perform this task. RFC 2236 defines IGMPv2.

Note By default, enabling a PIM on an interface enables IGMPv2 on that device. IGMPv2 was designed to be as backward compatible with IGMPv1 as possible. To accomplish this backward compatibility, RFC 2236 defined special interoperability rules. If your network contains legacy IGMPv1 hosts, you should be familiar with these operability rules. For more information about IGMPv1 and IGMPv2 interoperability, see RFC 2236, Internet Group Management Protocol, Version 2 .

Devices That Run IGMPv1

IGMPv1 devices send IGMP queries to the “all-hosts” multicast address of 224.0.0.1 to solicit multicast groups with active multicast receivers. The multicast receivers also can send IGMP reports to the device to notify it that they are interested in receiving a particular multicast stream. Hosts can send the report asynchronously or in response to the IGMP queries sent by the device. If more than one multicast receiver exists for the same multicast group, only one of these hosts sends an IGMP report message; the other hosts suppress their report messages.

In IGMPv1, there is no election of an IGMP querier. If more than one device on the segment exists, all the devices send periodic IGMP queries. IGMPv1 has no special mechanism by which the hosts can leave the group. If the hosts are no longer interested in receiving multicast packets for a particular group, they simply do not reply to the IGMP query packets sent from the device. The device continues sending query packets. If the device does not hear a response in three IGMP queries, the group times out and the device stops sending multicast packets on the segment for the group. If the host later wants to receive multicast packets after the timeout period, the host simply sends a new IGMP join to the device, and the device begins to forward the multicast packet again.

If there are multiple devices on a LAN, a designated router (DR) must be elected to avoid duplicating multicast traffic for connected hosts. PIM devices follow an election process to select a DR. The PIM device with the highest IP address becomes the DR.

The DR is responsible for the following tasks:

- Sending PIM register and PIM Join and Prune messages toward the rendezvous point (RP) to inform it about host group membership.
- Sending IGMP host-query messages.
- Sending host-query messages by default every 60 seconds in order to keep the IGMP overhead on hosts and networks very low.

Devices That Run IGMPv2

IGMPv2 improves the query messaging capabilities of IGMPv1.

The query and membership report messages in IGMPv2 are identical to the IGMPv1 messages with two exceptions:

- IGMPv2 query messages are broken into two categories: general queries (identical to IGMPv1 queries) and group-specific queries.
- IGMPv1 membership reports and IGMPv2 membership reports have different IGMP type codes.

IGMPv2 also enhances IGMP by providing support for the following capabilities:

- Querier election process--Provides the capability for IGMPv2 devices to elect the IGMP querier without having to rely on the multicast routing protocol to perform the process.
- Maximum Response Time field--A new field in query messages permits the IGMP querier to specify the maximum query-response time. This field permits the tuning of the query-response process to control response burstiness and to fine-tune leave latencies.
- Group-Specific Query messages--Permits the IGMP querier to perform the query operation on a specific group instead of all groups.

- Leave-Group messages--Provides hosts with a method of notifying devices on the network that they wish to leave the group.

Unlike IGMPv1, in which the DR and the IGMP querier are typically the same device, in IGMPv2 the two functions are decoupled. The DR and the IGMP querier are selected based on different criteria and may be different devices on the same subnet. The DR is the device with the highest IP address on the subnet, whereas the IGMP querier is the device with the lowest IP address.

Query messages are used to elect the IGMP querier as follows:

1. When IGMPv2 devices start, they each multicast a general query message to the all-systems group address of 224.0.0.1 with their interface address in the source IP address field of the message.
2. When an IGMPv2 device receives a general query message, the device compares the source IP address in the message with its own interface address. The device with the lowest IP address on the subnet is elected the IGMP querier.
3. All devices (excluding the querier) start the query timer, which is reset whenever a general query message is received from the IGMP querier. If the query timer expires, it is assumed that the IGMP querier has gone down, and the election process is performed again to elect a new IGMP querier.

By default, the timer is two times the query interval.

IGMP Join Process

When a host wants to join a multicast group, the host sends one or more unsolicited membership reports for the multicast group it wants to join. The IGMP join process is the same for IGMPv1 and IGMPv2 hosts.

In IGMPv3, the join process for hosts proceeds as follows:

- When a hosts wants to join a group, it sends an IGMPv3 membership report to 224.0.0.22 with an empty EXCLUDE list.
- When a host wants to join a specific channel, it sends an IGMPv3 membership report to 224.0.0.22 with the address of the specific source included in the INCLUDE list.
- When a host wants to join a group excluding particular sources, it sends an IGMPv3 membership report to 224.0.0.22 excluding those sources in the EXCLUDE list.

Note If some IGMPv3 hosts on a LAN wish to exclude a source and others wish to include the source, then the device will send traffic for the source on the LAN (that is, inclusion trumps exclusion in this situation).

IGMP Leave Process

The method that hosts use to leave a group varies depending on the version of IGMP in operation.

IGMPv1 Leave Process

There is no leave-group message in IGMPv1 to notify the devices on the subnet that a host no longer wants to receive the multicast traffic from a specific group. The host simply stops processing traffic for the multicast group and ceases responding to IGMP queries with IGMP membership reports for the group. As a result, the only way IGMPv1 devices know that there are no longer any active receivers for a particular multicast group on a subnet is when the devices stop receiving membership reports. To facilitate this process, IGMPv1 devices associate a countdown timer with an IGMP group on a subnet. When a membership report is received for the group on the subnet, the timer is reset. For IGMPv1 devices, this timeout interval is typically three times the query interval (3 minutes). This timeout interval means that the device may continue to forward multicast traffic onto the subnet for up to 3 minutes after all hosts have left the multicast group.

IGMPv2 Leave Process

IGMPv2 incorporates a leave-group message that provides the means for a host to indicate that it wishes to stop receiving multicast traffic for a specific group. When an IGMPv2 host leaves a multicast group, if it was the last host to respond to a query with a membership report for that group, it sends a leave-group message to the all-devices multicast group (224.0.0.2).

IGMPv3 Leave Process

IGMPv3 enhances the leave process by introducing the capability for a host to stop receiving traffic from a particular group, source, or channel in IGMP by including or excluding sources, groups, or channels in IGMPv3 membership reports.

IGMP Multicast Addresses

IP multicast traffic uses group addresses, which are Class D IP addresses. The high-order four bits of a Class D address are 1110. Therefore, host group addresses can be in the range 224.0.0.0 to 239.255.255.255.

Multicast addresses in the range 224.0.0.0 to 224.0.0.255 are reserved for use by routing protocols and other network control traffic. The address 224.0.0.0 is guaranteed not to be assigned to any group.

IGMP packets are transmitted using IP multicast group addresses as follows:

- IGMP general queries are destined to the address 224.0.0.1 (all systems on a subnet).
- IGMP group-specific queries are destined to the group IP address for which the device is querying.
- IGMP group membership reports are destined to the group IP address for which the device is reporting.
- IGMPv2 leave-group messages are destined to the address 224.0.0.2 (all devices on a subnet).
- IGMPv3 membership reports are destined to the address 224.0.0.22; all IGMPv3-capable multicast devices must listen to this address.

Extended ACL Support for IGMP to Support SSM in IPv4

The Extended ACL Support for IGMP to Support SSM in IPv4 feature enables IGMPv3 to accommodate extended access lists. IGMPv3 support of extended access lists allows you to leverage an important advantage of SSM in IPv4, that of filtering IGMPv3 reports based on source address, group address, or both.

Benefits of Extended Access List Support for IGMP to Support SSM in IPv4

IGMPv3 accommodates extended access lists, which allow you to leverage an important advantage of SSM in IPv4, that of basing access on source IP address. Prior to this feature, an IGMP access list accepted only a standard access list, allowing membership reports to be filtered based only on multicast group addresses.

IGMPv3 allows multicast receivers not only to join to groups, but to groups including or excluding sources. For appropriate access control, it is therefore necessary to allow filtering of IGMPv3 messages not only by group addresses reported, but by group and source addresses. IGMP extended access lists introduce this functionality. Using SSM with an IGMP extended access list (ACL) allows you to permit or deny source S and group G (S, G) in IGMPv3 reports, thereby filtering IGMPv3 reports based on source address, group address, or source and group address.

Source Addresses in IGMPv3 Reports for ASM Groups

IGMP extended access lists also can be used to permit or filter (deny) traffic based on (0.0.0.0, G), that is, (*, G) in IGMP reports that are non-SSM, such as Any Source Multicast (ASM).

Note The permit and deny statements equivalent to (*, G) are **permit host 0.0.0.0 host** *group-address* and **deny host 0.0.0.0 host group** *group-address*, respectively.

Filtering applies to IGMPv3 reports for both ASM and SSM groups, but it is most important for SSM groups because IP multicast routing ignores source addresses in IGMPv3 reports for ASM groups. Source addresses in IGMPv3 membership reports for ASM groups are stored in the IGMP cache (as displayed with the **show ip igmp membership** command), but PIM-based IP multicast routing considers only the ASM groups reported. Therefore, adding filtering for source addresses for ASM groups impacts only the IGMP cache for ASM groups.

How IGMP Checks an Extended Access List

When an IGMP extended access list is referenced in the **ip igmp access-group** command on an interface, the (S, G) pairs in the **permit** and **deny** statements of the extended access list are matched against the (S, G) pair of the IGMP reports received on the interface. For example, if an IGMP report with (S1, S2...Sn, G) is received, first the group (0.0.0.0, G) is checked against the access list statements. The convention (0.0.0.0, G) means (*, G), which is a wildcard source with a multicast group number. If the group is denied, the entire IGMP report is denied. If the group is permitted, each individual (S, G) pair is checked against the access list. Denied sources are taken out of the IGMP report, thereby denying the sources access to the multicast traffic.

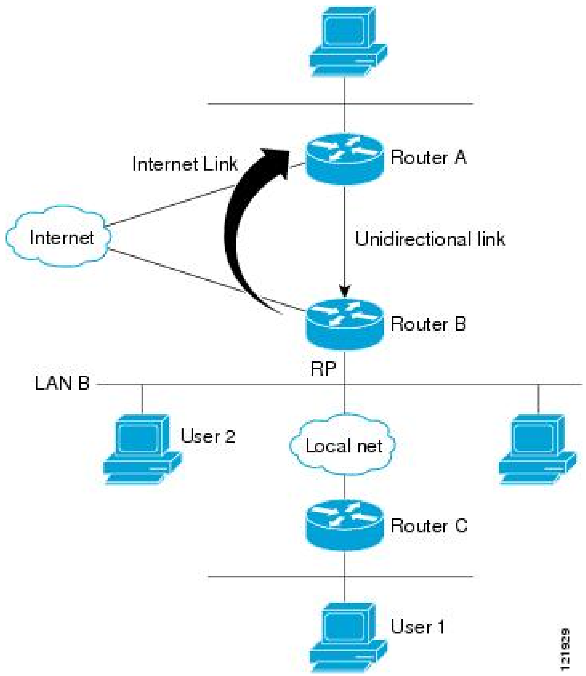
IGMP Proxy

An IGMP proxy enables hosts in a unidirectional link routing (UDLR) environment that are not directly connected to a downstream router to join a multicast group sourced from an upstream network.

The figure below illustrates a sample topology that shows two UDLR scenarios:

- Traditional UDL routing scenario--A UDL device with directly connected receivers.
- IGMP proxy scenario--UDL device without directly connected receivers.

Note IGMP UDLs are needed on the upstream and downstream devices.



Scenario 1--Traditional UDLR Scenario (UDL Device with Directly Connected Receivers)

For scenario 1, no IGMP proxy mechanism is needed. In this scenario, the following sequence of events occurs:

1. User 2 sends an IGMP membership report requesting interest in group G.
2. Router B receives the IGMP membership report, adds a forwarding entry for group G on LAN B, and proxies the IGMP report to Router A, which is the UDLR upstream device.
3. The IGMP report is then proxied across the Internet link.
4. Router A receives the IGMP proxy and maintains a forwarding entry on the unidirectional link.

Scenario 2--IGMP Proxy Scenario (UDL Device without Directly Connected Receivers)

For scenario 2, the IGMP proxy mechanism is needed to enable hosts that are not directly connected to a downstream device to join a multicast group sourced from an upstream network. In this scenario, the following sequence of events occurs:

1. User 1 sends an IGMP membership report requesting interest in group G.
2. Router C sends a PIM Join message hop-by-hop to the RP (Router B).
3. Router B receives the PIM Join message and adds a forwarding entry for group G on LAN B.
4. Router B periodically checks its mroute table and proxies the IGMP membership report to its upstream UDL device across the Internet link.
5. Router A creates and maintains a forwarding entry on the unidirectional link (UDL).

In an enterprise network, it is desirable to be able to receive IP multicast traffic via satellite and forward the traffic throughout the network. With unidirectional link routing (UDLR) alone, scenario 2 would not be possible because receiving hosts must be directly connected to the downstream device, Router B. The IGMP proxy mechanism overcomes this limitation by creating an IGMP report for (*, G) entries in the multicast forwarding table. To make this scenario functional, therefore, you must enable IGMP report forwarding of proxied (*, G) multicast static route (mroute) entries (using the **ip igmp mroute-proxy** command) and enable the mroute proxy service (using the **ip igmp proxy-service** command) on interfaces leading to PIM-enabled networks with potential members.

Note Because PIM messages are not forwarded upstream, each downstream network and the upstream network have a separate domain.

How to Customize IGMP

Configuring the Device to Forward Multicast Traffic in the Absence of Directly Connected IGMP Hosts

Perform this optional task to configure the device to forward multicast traffic in the absence of directly connected IGMP hosts.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
    - **ip igmp join-group** *group-address*
    - **ip igmp static-group** {***** | *group-address* [**source** *source-address*]}
5. **end**
6. **show ip igmp interface** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number*	Enters interface configuration mode.

	Command or Action	Purpose
	Example: `device(config)# interface gigabitethernet 1`	• For the *type* and *number* arguments, specify an interface that is connected to hosts.

Step 4	Do one of the following:
	• **ip igmp join-group** *group-address*
	• **ip igmp static-group** {***** | *group-address* [**source** *source-address*]}
	Example:
	
	`device(config-if)# ip igmp join-group 225.2.2.2`
	Example:
	
	`device(config-if)# ip igmp static-group 225.2.2.2`
		The first sample shows how to configure an interface on the device to join the specified group.
		With this method, the device accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the device from fast switching.
		The second example shows how to configure static group membership entries on an interface. With this method, the device does not accept the packets itself, but only forwards them. Hence, this method allows fast switching. The outgoing interface appears in the IGMP cache, but the device itself is not a member, as evidenced by lack of an "L" (local) flag in the multicast route entry

Step 5	**end** **Example:** `device#(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ip igmp interface** [*interface-type interface-number*] **Example:** `device# show ip igmp interface`	(Optional) Displays multicast-related information about an interface.

Controlling Access to an SSM Network Using IGMP Extended Access Lists

Perform this optional task to control access to an SSM network by using an IGMP extended access list that filters SSM traffic based on source address, group address, or both.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. **ip pim ssm** {**default** | **range** *access-list*}
5. **ip access-list extended** *access-list* -name
6. **deny igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
7. **permit igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
8. **exit**
9. interface type number
10. **ip igmp access-group** *access-list*

11. **ip pim sparse-mode**
12. Repeat Steps 1 through 11 on all interfaces that require access control of SSM channel membership.
13. **ip igmp version 3**
14. Repeat Step 13 on all host-facing interfaces.
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Device(config)# ip multicast-routing distributed`	Enables IP multicast routing. • The **distributed** keyword is required for IPv4 multicast..
Step 4	**ip pim ssm** {**default** | **range** *access-list*} **Example:** `Device(config)# ip pim ssm default`	Configures SSM service. • The **default** keyword defines the SSM range access list as 232/8. • The **range** keyword specifies the standard IP access list number or name that defines the SSM range.
Step 5	**ip access-list extended** *access-list* -name **Example:** `Device(config)# ip access-list extended mygroup`	Specifies an extended named IP access list.
Step 6	**deny igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# deny igmp host 10.1.2.3 any`	(Optional) Filters the specified source address or group address from the IGMP report, thereby restricting hosts on a subnet from membership to the (S, G) channel. • Repeat this step to restrict hosts on a subnet membership to other (S, G) channels. (These sources should be more specific than a subsequent **permit** statement because any sources or groups not specifically permitted are denied.) • Remember that the access list ends in an implicit **deny** statement.

	Command or Action	Purpose
		• This example shows how to create a deny statement that filters all groups for source 10.1.2.3, which effectively denies the source.
Step 7	**permit igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit igmp any any`	Allows a source address or group address in an IGMP report to pass the IP access list. • You must have at least one **permit** statement in an access list. • Repeat this step to allow other sources to pass the IP access list. • This example shows how to allow group membership to sources and groups not denied by prior **deny** statements.
Step 8	**exit** **Example:** `Device(config-ext-nacl)# exit`	Exits the current configuration session and returns to global configuration mode.
Step 9	interface type number **Example:** `Device(config)# interface ethernet 0`	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.
Step 10	**ip igmp access-group** *access-list* **Example:** `Device(config-if)# ip igmp access-group mygroup`	Applies the specified access list to IGMP reports.
Step 11	**ip pim sparse-mode** **Example:** `Device(config-if)# ip pim sparse-mode`	Enables PIM-SM on the interface. **Note** You must use sparse mode.
Step 12	Repeat Steps 1 through 11 on all interfaces that require access control of SSM channel membership.	--
Step 13	**ip igmp version 3** **Example:** `Device(config-if)# ip igmp version 3`	Enables IGMPv3 on this interface. The default version of IGMP is IGMP version 2. Version 3 is required by SSM.
Step 14	Repeat Step 13 on all host-facing interfaces.	--
Step 15	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	

Configuring an IGMP Proxy

Perform this optional task to configure unidirectional link (UDL) routers to use the IGMP proxy mechanism. An IGMP proxy enables hosts in a unidirectional link routing (UDLR) environment that are not directly connected to a downstream router to join a multicast group sourced from an upstream network.

To configure an IGMP proxy, you will need to perform the following tasks:

Prerequisites for IGMP Proxy

Before configuring an IGMP proxy, ensure that the following conditions exist:

- All routers on the IGMP UDL have the same subnet address. If all routers on the UDL cannot have the same subnet address, the upstream router must be configured with secondary addresses to match all the subnets that the downstream routers are attached to.
- This task assumes that IP multicast has been enabled and that the PIM interfaces have been configured.

When enabling PIM on the interfaces for the IGMP proxy scenario, keep in mind the following guidelines:

-
  - Use PIM sparse mode (PIM-SM) when the interface is operating in a sparse-mode region and you are running static RP, bootstrap (BSR), or Auto-RP with the Auto-RP listener capability.
  - Use PIM sparse-dense mode when the interface is running in a sparse-dense mode region and you are running Auto-RP without the Auto-RP listener capability.
  - Use PIM dense mode (PIM-DM) for this step when the interface is operating in dense mode and is, thus, participating in a dense-mode region.
  - Use PIM-DM with the proxy-register capability when the interface is receiving source traffic from a dense-mode region that needs to reach receivers that are in a sparse-mode region.

Configuring the Upstream UDL Device for IGMP UDLR

Perform this task to configure the upstream UDL device for IGMP UDLR.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp unidirectional-link**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Enters interface configuration mode. • For the *type* and *number* arguments, specify the interface to be used as the UDL on the upstream device.
Step 4	**ip igmp unidirectional-link** **Example:** `Device(config-if)# ip igmp unidirectional-link`	Configures IGMP on the interface to be unidirectional for IGMP UDLR.
Step 5	**end** **Example:** `Device(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.

Configuring the Downstream UDL Device for IGMP UDLR with IGMP Proxy Support

Perform this task to configure the downstream UDL device for IGMP UDLR with IGMP proxy support.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp unidirectional-link**
5. **exit**
6. **interface** *type number*
7. **ip igmp mroute-proxy** *type number*
8. **exit**
9. **interface** *type number*
10. **ip igmp helper-address udl** *interface-type interface-number*
11. **ip igmp proxy-service**
12. **end**
13. **show ip igmp interface**
14. **show ip igmp udlr**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type* *number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Enters interface configuration mode. • For the *type* and *number* arguments, specify the interface to be used as the UDL on the downstream device for IGMP UDLR.
Step 4	**ip igmp unidirectional-link** **Example:** `Device(config-if)# ip igmp unidirectional-link`	Configures IGMP on the interface to be unidirectional for IGMP UDLR.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Enters interface configuration mode. • For the *type* and *number* arguments, select an interface that is facing the nondirectly connected hosts.
Step 7	**ip igmp mroute-proxy** *type number* **Example:** `Device(config-if)# ip igmp mroute-proxy loopback 0`	Enables IGMP report forwarding of proxied (*, G) multicast static route (mroute) entries. • This step is performed to enable the forwarding of IGMP reports to a proxy service interface for all (*, G) forwarding entries in the multicast forwarding table. • In this example, the **ip igmp mroute-proxy** command is configured on Gigabit Ethernet interface 1/0/0 to request that IGMP reports be sent to loopback interface 0 for all groups in the mroute table that are forwarded to Gigabit Ethernet interface 1/0/0.
Step 8	**exit** **Example:**	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-if)# exit`	
Step 9	**interface** *type number* **Example:** `Device(config)# interface loopback 0`	Enters interface configuration mode for the specified interface. • In this example, loopback interface 0 is specified.
Step 10	**ip igmp helper-address udl** *interface-type interface-number* **Example:** `Device(config-if)# ip igmp helper-address udl gigabitethernet 0/0/0`	Configures IGMP helpering for UDLR. • This step allows the downstream device to helper IGMP reports received from hosts to an upstream device connected to a UDL associated with the interface specified for the *interface-type* and *interface-number* arguments. • In the example topology, IGMP helpering is configured over loopback interface 0 on the downstream device. Loopback interface 0, thus, is configured to helper IGMP reports from hosts to an upstream device connected to Gigabit Ethernet interface 0/0/0.
Step 11	**ip igmp proxy-service** **Example:** `Device(config-if)# ip igmp proxy-service`	Enables the mroute proxy service. • When the mroute proxy service is enabled, the device periodically checks the static mroute table for (*, G) forwarding entries that match interfaces configured with the **ip igmp mroute-proxy** command (see Step 7) based on the IGMP query interval. Where there is a match, one IGMP report is created and received on this interface. **Note** The **ip igmp proxy-service** command is intended to be used with the **ip igmp helper-address** (UDL) command. • In this example, the **ip igmp proxy-service** command is configured on loopback interface 0 to enable the forwarding of IGMP reports out the interface for all groups on interfaces registered through the **ip igmp mroute-proxy** command (see Step 7).
Step 12	**end** **Example:** `Device(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 13	**show ip igmp interface** **Example:**	(Optional) Displays multicast-related information about an interface.

	Command or Action	Purpose
	`Device# show ip igmp interface`	
Step 14	**show ip igmp udlr** **Example:** `Device# show ip igmp udlr`	(Optional) Displays UDLR information for directly connected multicast groups on interfaces that have a UDL helper address configured.

Configuration Examples for Customizing IGMP

Example: Configuring the Device to Forward Multicast Traffic in the Absence of Directly Connected IGMP Hosts

The following example shows how to configure a device to forward multicast traffic in the absence of directly connected IGMP hosts using the **ip igmp join-group** command. With this method, the device accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the device from fast switching.

In this example, Fast Ethernet interface 0/0/0 on the device is configured to join the group 225.2.2.2:

```
interface FastEthernet0/0/0
 ip igmp join-group 225.2.2.2
```

The following example shows how to configure a device to forward multicast traffic in the absence of directly connected IGMP hosts using the **ip igmp static-group** command. With this method, the device does not accept the packets itself, but only forwards them. Hence, this method allows fast switching. The outgoing interface appears in the IGMP cache, but the device itself is not a member, as evidenced by lack of an "L" (local) flag in the multicast route entry.

In this example, static group membership entries for group 225.2.2.2 are configured on Fast Ethernet interface 0/1/0:

```
interface FastEthernet0/1/0
 ip igmp static-group 225.2.2.2
```

Controlling Access to an SSM Network Using IGMP Extended Access Lists

This section contains the following configuration examples for controlling access to an SSM network using IGMP extended access lists:

Note Keep in mind that access lists are very flexible: there are many combinations of permit and deny statements one could use in an access list to filter multicast traffic. The examples in this section simply provide a few examples of how it can be done.

Example: Denying All States for a Group G

The following example shows how to deny all states for a group G. In this example, Fast Ethernet interface 0/0/0 is configured to filter all sources for SSM group 232.2.2.2 in IGMPv3 reports, which effectively denies this group.

```
ip access-list extended test1
 deny igmp any host 232.2.2.2
 permit igmp any any
!
interface FastEthernet0/0/0
 ip igmp access-group test1
```

Example: Denying All States for a Source S

The following example shows how to deny all states for a source S. In this example, Gigabit Ethernet interface 1/1/0 is configured to filter all groups for source 10.2.1.32 in IGMPv3 reports, which effectively denies this source.

```
ip access-list extended test2
 deny igmp host 10.2.1.32 any
 permit igmp any any
!
interface GigabitEthernet1/1/0
 ip igmp access-group test2
```

Example: Permitting All States for a Group G

The following example shows how to permit all states for a group G. In this example, Gigabit Ethernet interface 1/2/0 is configured to accept all sources for SSM group 232.1.1.10 in IGMPv3 reports, which effectively accepts this group altogether.

```
ip access-list extended test3
 permit igmp any host 232.1.1.10
!
interface GigabitEthernet1/2/0
 ip igmp access-group test3
```

Example: Permitting All States for a Source S

The following example shows how to permit all states for a source S. In this example, Gigabit Ethernet interface 1/2 is configured to accept all groups for source 10.6.23.32 in IGMPv3 reports, which effectively accepts this source altogether.

```
ip access-list extended test4
 permit igmp host 10.6.23.32 any
!
interface GigabitEthernet1/2/0
 ip igmp access-group test4
```

Example: Filtering a Source S for a Group G

The following example shows how to filter a particular source S for a group G. In this example, Gigabit Ethernet interface 0/3/0 is configured to filter source 232.2.2.2 for SSM group 232.2.30.30 in IGMPv3 reports.

```
ip access-list extended test5
 deny igmp host 10.4.4.4 host 232.2.30.30
 permit igmp any any
!
interface GigabitEthernet0/3/0
 ip igmp access-group test5
```

Example: IGMP Proxy Configuration

The following example shows how to configure the upstream UDL device for IGMP UDLR and the downstream UDL device for IGMP UDLR with IGMP proxy support.

Upstream Device Configuration

```
interface gigabitethernet 0/0/0
ip address 10.1.1.1 255.255.255.0
ip pim sparse-mode
!
interface gigabitethernet 1/0/0
ip address 10.2.1.1 255.255.255.0
ip pim sparse-mode
ip igmp unidirectional-link
!
interface gigabitethernet 2/0/0
ip address 10.3.1.1 255.255.255.0
```

Downstream Device Configuration

```
ip pim rp-address 10.5.1.1 5
access-list 5 permit 239.0.0.0 0.255.255.255
!
interface loopback 0
ip address 10.7.1.1 255.255.255.0
ip pim sparse-mode
ip igmp helper-address udl ethernet 0
ip igmp proxy-service
!
interface gigabitethernet 0/0/0
ip address 10.2.1.2 255.255.255.0
ip pim sparse-mode
ip igmp unidirectional-link
!
interface gigabitethernet 1/0/0
ip address 10.5.1.1 255.255.255.0
ip pim sparse-mode
ip igmp mroute-proxy loopback 0
!
interface gigabitethernet 2/0/0
ip address 10.6.1.1 255.255.255.0
```

Additional References

The following sections provide references related to customizing IGMP.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP Multicast Command Reference
Overview of the IP multicast technology area	" IP Multicast Technology Overview " module
Basic IP multicast concepts, configuration tasks, and examples	" Configuring Basic IP Multicast " or "Configuring IP Multicast in IPv6 Networks" module

Standards and RFCs

Standard/RFC	Title
RFC 1112	*Host extensions for IP multicasting*
RFC 2236	*Internet Group Management Protocol, Version 2*
RFC 3376	*Internet Group Management Protocol, Version 3*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by these features, and support for existing MIBs has not been modified by these features.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Customizing IGMP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 27: Feature Information for Customizing IGMP

Feature Name	Releases	Feature Information
IGMP Version 3 - Explicit Tracking of Hosts, Groups, and Channels	Cisco IOS XE Release 3.8S	IGMPv3 provides for source filtering, which enables a multicast receiver host to signal to a device which groups it wants to receive multicast traffic from, and from which sources this traffic is expected. In addition, IGMPv3 supports the link local address 224.0.0.22, which is the destination IP address for IGMPv3 membership reports; all IGMPv3-capable multicast devices must listen to this address. RFC 3376 defines IGMPv3.
UDLR Tunnel ARP and IGMP Proxy	Cisco IOS XE Release 3.8S	This feature enables ARP over a unidirectional link, and overcomes the existing limitation of requiring downstream multicast receivers to be directly connected to the unidirectional link downstream router.

CHAPTER 38

IGMPv3 Host Stack

This module describes how to configure Internet Group Management Protocol (IGMP) Version 3 (v3) Host Stack feature for enabling devices to function as multicast network endpoints or hosts.

Prerequisites for IGMPv3 Host Stack

- IP multicast is enabled and all Protocol Independent Multicast (PIM) interfaces have been configured using the tasks described in the "Configuring Basic IP Multicast" module of the *IP Multicast: PIM Configuration Guide*.
- IGMP version 3 must be configured on the interface.
- The device must be configured for SSM. IGMPv3 membership reports are sent for SSM channels only.

Information About IGMPv3 Host Stack

IGMPv3

Internet Group Management Protocol (IGMP) is the protocol used by IPv4 devices to report their IP multicast group memberships to neighboring multicast devices. Version 3 (v3) of IGMP adds support for source filtering. Source filtering enables a multicast receiver host to signal from which groups it wants to receive multicast traffic, and from which sources this traffic is expected. 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.

In addition, IGMPv3 supports the link local address 224.0.0.22, which is the destination IP address for IGMPv3 membership reports; all IGMPv3-capable multicast devices must listen to this address. RFC 3376 defines IGMPv3.

IGMPv3 Host Stack

The IGMPv3 Host Stack feature enables devices to function as multicast network endpoints or hosts. The feature adds INCLUDE mode capability to the IGMPv3 host stack for Source Specific Multicast (SSM) groups. Enabling the IGMPv3 host stack ensures that hosts on a LAN can leverage SSM by enabling the device to initiate IGMPv3 joins, such as in environments where fast channel change is required in a SSM deployments.

To support of the IGMPv3 Host Stack feature, you must configure the INCLUDE mode capability on the IGMPv3 host stack for SSM groups. When the IGMPv3 Host Stack feature is configured, an IGMPv3 membership report is sent when one of the following events occurs:

- When an interface is configured to join a group and source and there is no existing state for this (S, G) channel, an IGMPv3 report of group record type ALLOW_NEW_SOURCES for the source specified is sent on that interface.
- When membership for a group and source is cancelled and there is state for this (S, G) channel, an IGMPv3 report of group record type BLOCK_OLD_SOURCES for the source specified is sent on that interface.
- When a query is received, an IGMPv3 report is sent as defined in RFC 3376.

Note For more information about IGMPv3 group record types and membership reports, see *RFC 3376, Internet Group Management Protocol, Version 3* .

How to Configure IGMPv3 Host Stack

Enabling the IGMPv3 Host Stack

Note If the **ip igmp join-group** command is configured for a group and source and IGMPv3 is not configured on the interface, (S, G) state will be created but no IGMPv3 membership reports will be sent.

Perform this task to add INCLUDE mode capability to the IGMPv3 host stack for SSM groups

.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp version 3**
5. **ip igmp join-group** *group - address* **source** *source - address*
6. **end**
7. **show ip igmp groups detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type* *number* **Example:** `Device(config)# interface FastEthernet 1`	Enters interface configuration mode. The specified interface must be connected to hosts.
Step 4	**ip igmp version 3** **Example:** `Device(config-if)# ip igmp version 3`	Enables IGMPv3 on the interface.
Step 5	**ip igmp join-group** *group* **-** *address* **source** *source* - *address* **Example:** `Device(config-if)# ip igmp join-group 232.2.2.2 source 10.1.1.1`	Configures the interface to join the specified (S, G) channel and enables the device to provide INCLUDE mode capability for the (S, G) channel . **Note** Repeat this step for each channel to be configured with the INCLUDE mode capability.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show ip igmp groups detail** **Example:** `Device# show ip igmp groups detail`	Displays directly-connected multicast groups that were learned through IGMP.

Configuration Examples for IGMPv3 Host Stack

Example: Enabling the IGMPv3 Host Stack

The following example shows how to add INCLUDE mode capability to the IGMPv3 host stack for SSM groups:

```
interface FastEthernet0/0/0
 ip igmp join-group 232.2.2.2 source 10.1.1.1
 ip igmp join-group 232.2.2.2 source 10.5.5.5
 ip igmp join-group 232.2.2.2 source 10.5.5.6
 ip igmp join-group 232.2.2.4 source 10.5.5.5
 ip igmp join-group 232.2.2.4 source 10.5.5.6
 ip igmp version 3
```

Based on the configuration presented in the preceding example, the following is sample output from the **debug igmp** command. The messages confirm that IGMPv3 membership reports are being sent after IGMPv3 and SSM are enabled:

```
Device# debug igmp

*May 4 23:48:34.251: IGMP(0): Group 232.2.2.2 is now in the SSM range, changing
*May 4 23:48:34.251: IGMP(0): Building v3 Report on GigabitEthernet0/0/0
*May 4 23:48:34.251: IGMP(0): Add Group Record for 232.2.2.2, type 5
*May 4 23:48:34.251: IGMP(0): Add Source Record 10.1.1.1
*May 4 23:48:34.251: IGMP(0): Add Source Record 10.5.5.5
*May 4 23:48:34.251: IGMP(0): Add Source Record 10.5.5.6
*May 4 23:48:34.251: IGMP(0): Add Group Record for 232.2.2.2, type 6
*May 4 23:48:34.251: IGMP(0): No sources to add, group record removed from report
*May 4 23:48:34.251: IGMP(0): Send unsolicited v3 Report with 1 group records on
FastEthernet0/0/0
*May 4 23:48:34.251: IGMP(0): Group 232.2.2.4 is now in the SSM range, changing
*May 4 23:48:34.251: IGMP(0): Building v3 Report on GigabitEthernet0/0/0
*May 4 23:48:34.251: IGMP(0): Add Group Record for 232.2.2.4, type 5
*May 4 23:48:34.251: IGMP(0): Add Source Record 10.5.5.5
*May 4 23:48:34.251: IGMP(0): Add Source Record 10.5.5.6
*May 4 23:48:34.251: IGMP(0): Add Group Record for 232.2.2.4, type 6
*May 4 23:48:34.251: IGMP(0): No sources to add, group record removed from report
*May 4 23:48:34.251: IGMP(0): Send unsolicited v3 Report with 1 group records on
FastEthernet0/0/0
*May 4 23:48:35.231: IGMP(0): Building v3 Report on GigabitEthernet0/0/0
*May 4 23:48:35.231: IGMP(0): Add Group Record for 232.2.2.2, type 5
*May 4 23:48:35.231: IGMP(0): Add Source Record 10.1.1.1
*May 4 23:48:35.231: IGMP(0): Add Source Record 10.5.5.5
*May 4 23:48:35.231: IGMP(0): Add Source Record 10.5.5.6
*May 4 23:48:35.231: IGMP(0): Add Group Record for 232.2.2.2, type 6
*May 4 23:48:35.231: IGMP(0): No sources to add, group record removed from report
*May 4 23:48:35.231: IGMP(0): Send unsolicited v3 Report with 1 group records on
FastEthernet0/0/0
*May 4 23:48:35.231: IGMP(0): Building v3 Report on GigabitEthernet0/0/0
*May 4 23:48:35.231: IGMP(0): Add Group Record for 232.2.2.4, type 5
*May 4 23:48:35.231: IGMP(0): Add Source Record 10.5.5.5
*May 4 23:48:35.231: IGMP(0): Add Source Record 10.5.5.6
*May 4 23:48:35.231: IGMP(0): Add Group Record for 232.2.2.4, type 6
*May 4 23:48:35.231: IGMP(0): No sources to add, group record removed from report
*May 4 23:48:35.231: IGMP(0): Send unsolicited v3 Report with 1 group records on
FastEthernet0/0/0
```

The following is sample output from the **show ip igmp groups detail** command for this configuration example scenario. This command can be used to verify that the device has received membership reports for (S, G) channels that are configured to join a group. When the device is correctly receiving IGMP membership reports for a channel, the "Flags:" output field will display the L and SSM flags.

```
Device# show ip igmp groups detail

Flags: L - Local, U - User, SG - Static Group, VG - Virtual Group,
       SS - Static Source, VS - Virtual Source
Interface:      FastEthernet0/0/0
```

```
Group:          232.2.2.2
Flags:          L SSM
Uptime:         00:04:12
Group mode:     INCLUDE
Last reporter:  10.4.4.7
Group source list: © - Cisco Src Report, U - URD, R - Remote, S - Static,
                    V - Virtual, Ac - Accounted towards access control limit,
                    M - SSM Mapping, L - Local)
  Source Address   Uptime    v3 Exp   CSR Exp   Fwd  Flags
  10.1.1.1         00:04:10  stopped  stopped    Yes  L
  10.5.5.5         00:04:12  stopped  stopped    Yes  L
  10.5.5.6         00:04:12  stopped  stopped    Yes  L
Interface:      FastEthernet0/0/0
Group:          232.2.2.3
Flags:          L SSM
Uptime:         00:04:12
Group mode:     INCLUDE
Last reporter:  10.4.4.7
Group source list: © - Cisco Src Report, U - URD, R - Remote, S - Static,
                    V - Virtual, Ac - Accounted towards access control limit,
                    M - SSM Mapping, L - Local)
  Source Address   Uptime    v3 Exp   CSR Exp   Fwd  Flags
  10.5.5.5         00:04:14  stopped  stopped   Yes  L
  10.5.5.6         00:04:14  stopped  stopped   Yes  L
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 3376	*Internet Group Management Protocol, Version 3*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by these features, and support for existing MIBs has not been modified by these features.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IGMPv3 Host Stack

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 28: Feature Information for IGMPv3 Host Stack

Feature Name	Releases	Feature Information
IGMPv3 Host Stack	12.3(14)T 12.2(33)SRE Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.3SG 15.1(1)SG 15.1(1)SY	The IGMPv3 Host Stack feature enables Cisco devices to function as multicast network endpoints or hosts. The feature adds the INCLUDE mode capability to the IGMPv3 host stack for SSM groups. Enabling the IGMPv3 host stack ensures that hosts on a LAN can leverage SSM by enabling the device to initiate IGMPv3 joins, such as in environments where fast channel change is required in a SSM deployments. The following commands were introduced or modified: **ip igmp join-group**.

CHAPTER 39

IGMP Static Group Range Support

This module describes how you can simplify the administration of networks with devices that require static group membership entries on many interfaces by configuring IGMP static group range support to specify group ranges in class maps and attach the class maps to an interface.

Information About IGMP Static Group Range Support

IGMP Static Group Range Support Overview

Prior to the introduction of the IGMP Static Group Range Support feature, there was no option to specify group ranges for static group membership. Administering devices that required static group membership entries on many interfaces was challenging in some network environments because each static group had to be configured individually. The result was configurations that were excessively long and difficult to manage.

The IGMP Static Group Range Support feature introduces the capability to configure group ranges in class maps and attach class maps to the interface.

Class Maps for IGMP Static Group Range Support

A class is a way of identifying a set of packets based on its contents. A class is designated through class maps. Typically, class maps are used to create traffic policies. Traffic policies are configured using the modular quality of service (QoS) command-line interface (CLI) (MQC). The normal procedure for creating traffic policies entails defining a traffic class, creating a traffic policy, and attaching the policy to an interface.

The IGMP Static Group Range Support feature introduces a type of class map that is used to define group ranges, group addresses, Source Specific Multicast (SSM) channels, and SSM channel ranges. Once created, the class map can be attached to interfaces.

Although IGMP Static Group Range Support feature uses the MQC to define class maps, the procedure for configuring IGMP static group class maps is different from the procedure used to create class maps for configuring QoS traffic policies. To configure the IGMP Static Group Range Support feature, you must perform the following:

1. Create an IGMP static group class map.
2. Define the group entries associated with the class map.
3. Attach the class map to an interface.

Unlike QoS class maps, which are defined by specifying numerous match criteria, IGMP static group class maps are defined by specifying multicast groups entries (group addresses, group ranges, SSM channels, and SSM channel ranges). Also, IGMP static group range class maps are not configured in traffic policies. Rather, the **ip ignmp static-group** command has been extended to support IGMP static group ranges.

Once a class map is attached to an interface, all group entries defined in the class map become statically connected members on the interface and are added to the IGMP cache and IP multicast route (mroute) table.

General Procedure for Configuring IGMP Group Range Support

To configure the IGMP Static Group Range Support feature, you would complete the following procedure:

1. Create an IGMP static group class map (using the **class-map type multicast-flows** command).
2. Define the group entries associated with the class map (using the **group** command).
3. Attach the class map to an interface (using the **ip igmp static-group** command).

The **class-map type multicast-flows** command is used to enter multicast-flows class map configuration mode to create or modify an IGMP static group class map.

Unlike QoS class maps, which are defined by specifying numerous match criteria, IGMP static group class maps are defined by specifying multicast groups entries (group addresses, group ranges, SSM channels, and SSM channel ranges). The following forms of the group command are entered from multicast-flows class map configuration mode to define group entries to associate with the class map:

- **group** *group-address*

Defines a group address to be associated with an IGMP static group class map.

- **group** *group-address* **to** *group-address*

Defines a range of group addresses to be associated with an IGMP static group class map.

- **group** *group-address* **source** *source-address*

Defines an SSM channel to be associated with an IGMP static group class map.

- **group** *group-address* **to** *group-address* **source** *source-address*

Defines a range of SSM channels to be associated with an IGMP static group class map.

Unlike QoS class maps, IGMP static group range class maps are not configured in traffic policies. Rather, the **ip igmp static-group** command has been extended to support IGMP static group ranges. After creating an IGMP static group class map, you can attach the class map to interfaces using the **ip igmp static-group** command with the **class-map**keyword and *class-map-name* argument. Once a class map is attached to an interface, all group entries defined in the class map become statically connected members on the interface and are added to the IGMP cache and IP multicast route (mroute) table.

Additional Guidelines for Configuring IGMP Static Group Range Support

- Only one IGMP static group class map can be attached to an interface.
- If an IGMP static group class map is modified (that is, if group entries are added to or removed from the class map using the **group** command), the group entries that are added to or removed from the IGMP static group class map are added to or deleted from the IGMP cache and the mroute table, respectively.
- If an IGMP static group class map is replaced on an interface by another class map using the **ip igmp static-group** command, the group entries associated with old class map are removed, and the group entries defined in the new class map are added to the IGMP cache and mroute table.
- The **ip igmp static-group** command accepts an IGMP static group class map for the *class-map-name* argument, regardless of whether the class map configuration exists. If a class map attached to an interface does not exist, the class map remains inactive. Once the class map is configured, all group entries associated with the class map are added to the IGMP cache and mroute table.
- If a class map is removed from an interface using the **no** form of the **ip igmp static-group** command, all group entries defined in the class map are removed from the IGMP cache and mroute tables.

Benefits of IGMP Static Group Range Support

The IGMP Static Group Range Support feature provides the following benefits:

- Simplifies the administration of devices that require many interfaces to be configured with many different **ip igmp static-group** command configurations by introducing the capability to configure group ranges in class maps and attach class maps to the **ip igmp static-group** command.
- Reduces the number of commands required to administer devices that require many **ip igmp static-group** command configurations.

How to Configure IGMP Static Group Range Support

Configuring IGMP Static Group Range Support

Perform this task to create and define an IGMP static group class and attach the class to an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type multicast-flows** *class-map-name*
4. **group** *group-address* [**to** *group-address*] [**source** *source-address*]
5. **exit**
6. Repeat Steps 3 to 5 to create additional class maps.
7. **interface** *type number*
8. **ip igmp static-group class-map** *class-map-name*
9. **ip igmp static-group ***
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type multicast-flows** *class-map-name* **Example:** `Device(config)# class-map type multicast-flows static1`	Enters multicast-flows class map configuration mode to create or modify an IGMP static group class map.
Step 4	**group** *group-address* [**to** *group-address*] [**source** *source-address*] **Example:** `Device(config-mcast-flows-cmap)# group 232.1.1.7 to 232.1.1.20`	Defines the group entries to be associated with the class map. • Repeat this step to associate additional group entires to the class map being configured.
Step 5	**exit** **Example:** `Device(config-mcast-flows-cmap)# exit`	Exits multicast-flows class-map configuration mode and returns to global configuration mode.
Step 6	Repeat Steps 3 to 5 to create additional class maps.	--
Step 7	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/1`	Enters interface configuration mode.
Step 8	**ip igmp static-group class-map** *class-map-name* **Example:** `Device(config-if)# ip igmp static-group class-map static1`	Attaches an IGMP static group class map to the interface.
Step 9	**ip igmp static-group *** **Example:** `Device(config-if)# ip igmp static-group *`	(Optional) Places the interface into all created multicast route (mroute) entries. • Depending on your Cisco software release, this step is required if the interface of a last hop device does not have any PIM neighbors and does not have a

	Command or Action	Purpose
		receiver. See the **ip igmp static-group** command in the *Cisco IOS IP Multicast Command Reference*.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode, and enters privileged EXEC mode.

Verifying IGMP Static Group Range Support

Perform this optional task to verify the contents of IGMP static group class maps configurations, and to confirm that all group entries defined in class maps were added to the IGMP cache and the mroute table after you attached class maps to interfaces.

SUMMARY STEPS

1. **show ip igmp static-group class-map** [**interface** [*type number*]]
2. **show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]
3. **show ip mroute**

DETAILED STEPS

Step 1 **show ip igmp static-group class-map** [**interface** [*type number*]]

Displays the contents of IGMP static group class maps and the interfaces using class maps.

The following is sample output from the **show ip igmp static-group class-map** command:

Example:

```
Device# show ip igmp static-group class-map

Class-map static1
  Group address range 228.8.8.7 to 228.8.8.9
  Group address 232.8.8.7, source address 10.1.1.10
  Interfaces using the classmap:
    Loopback0
Class-map static
  Group address range 232.7.7.7 to 232.7.7.9, source address 10.1.1.10
  Group address 227.7.7.7
  Group address range 227.7.7.7 to 227.7.7.9
  Group address 232.7.7.7, source address 10.1.1.10
  Interfaces using the classmap:
    FastEthernet3/1
```

The following is sample output from the **show ip igmp static-group class-map**command with the **interface** keyword:

Example:

```
Device# show ip igmp static-group class-map interface

Loopback0
```

```
  Class-map attached: static1
FastEthernet3/1
  Class-map attached: static
```

The following is sample output from the **show ip igmp static-group class-map**command with the **interface** keyword and *type number* arguments:

Example:

```
Device# show ip igmp static-group class-map interface FastEthernet 3/1

FastEthernet3/1
  Class-map attached: static
```

Step 2 **show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]

Displays the multicast groups with receivers that are directly connected to the device and that are learned through IGMP.

The following is sample output from the **show ip igmp groups** command:

Example:

```
device# show ip igmp groups

IGMP Connected Group Membership
Group Address    Interface                 Uptime    Expires   Last Reporter
232.7.7.7        FastEthernet3/1           00:00:09  stopped   0.0.0.0
232.7.7.9        FastEthernet3/1           00:00:09  stopped   0.0.0.0
232.7.7.8        FastEthernet3/1           00:00:09  stopped   0.0.0.0
227.7.7.7        FastEthernet3/1           00:00:09  stopped   0.0.0.0
227.7.7.9        FastEthernet3/1           00:00:09  stopped   0.0.0.0
227.7.7.8        FastEthernet3/1           00:00:09  stopped   0.0.0.0
224.0.1.40       FastEthernet3/2           01:44:50  00:02:09  10.2.2.5
224.0.1.40       Loopback0                 01:45:22  00:02:32  10.3.3.4
```

Step 3 **show ip mroute**

Displays the contents of the mroute table.

The following is sample output from the **show ip mroute** command:

Example:

```
Device# show ip mroute

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.1.1.10, 232.7.7.7), 00:00:17/00:02:42, flags: sTI
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.5
  Outgoing interface list:
    FastEthernet3/1, Forward/Sparse-Dense, 00:00:17/00:02:42
(10.1.1.10, 232.7.7.9), 00:00:17/00:02:42, flags: sTI
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.5
  Outgoing interface list:
    FastEthernet3/1, Forward/Sparse-Dense, 00:00:17/00:02:42
```

```
(10.1.1.10, 232.7.7.8), 00:00:18/00:02:41, flags: sTI
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.5
  Outgoing interface list:
    FastEthernet3/1, Forward/Sparse-Dense, 00:00:18/00:02:41
(*, 227.7.7.7), 00:00:18/00:02:41, RP 10.2.2.6, flags: SJC
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.6
  Outgoing interface list:
    FastEthernet3/1, Forward/Sparse-Dense, 00:00:18/00:02:41
(*, 227.7.7.9), 00:00:18/00:02:41, RP 10.2.2.6, flags: SJC
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.6
  Outgoing interface list:
    FastEthernet3/1, Forward/Sparse-Dense, 00:00:18/00:02:41
(*, 227.7.7.8), 00:00:18/00:02:41, RP 10.2.2.6, flags: SJC
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.6
  Outgoing interface list:
    FastEthernet3/1, Forward/Sparse-Dense, 00:00:18/00:02:41
(*, 224.0.1.40), 00:01:40/00:02:23, RP 10.2.2.6, flags: SJCL
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.6
  Outgoing interface list:
    Loopback0, Forward/Sparse-Dense, 00:01:40/00:02:23
```

Configuration Examples for IGMP Static Group Range Support

Example: Configuring IGMP Static Group Support

The following example shows how to configure a class map and attach the class map to an interface. In this example, a class map named static is configured and attached to FastEthernet interface 3/1.

```
class-map type multicast-flows static
 group 227.7.7.7
 group 232.7.7.7 to 232.7.7.9 source 10.1.1.10
 group 232.7.7.7 source 10.1.1.10
 group 227.7.7.7 to 227.7.7.9
.
.
.
!
interface FastEthernet3/1
 ip address 192.168.1. 2 255.255.255.0
 ip pim sparse-dense-mode
 ip igmp static-group class-map static
!
```

Example: Verifying IGMP Static Group Support

The following is sample output from the **show ip igmp static-group class-map** command. In this example, the output displays the contents of the IGMP static group class map named static (the class map configured in the Example: Configuring IGMP Static Group Support, on page 525 section).

```
Device# show ip igmp static-group class-map

Class-map static
  Group address range 227.7.7.7 to 227.7.7.9
```

```
  Group address 232.7.7.7, source address 10.1.1.10
  Group address range 232.7.7.7 to 232.7.7.9, source address 10.1.1.10
  Group address 227.7.7.7
  Interfaces using the classmap:
    FastEthernet3/1
```

The following is sample output from the **show ip igmp groups** command. In this example, the command is issued to confirm that the group entries defined in the class map named static (the class map configured in the Example: Configuring IGMP Static Group Support, on page 525 section) were added to the IGMP cache.

```
Device# show ip igmp groups
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
232.7.7.7        FastEthernet3/1          00:00:09  stopped   0.0.0.0
232.7.7.9        FastEthernet3/1          00:00:09  stopped   0.0.0.0
232.7.7.8        FastEthernet3/1          00:00:09  stopped   0.0.0.0
227.7.7.7        FastEthernet3/1          00:00:09  stopped   0.0.0.0
227.7.7.9        FastEthernet3/1          00:00:09  stopped   0.0.0.0
227.7.7.8        FastEthernet3/1          00:00:09  stopped   0.0.0.0
224.0.1.40       FastEthernet3/2          01:44:50  00:02:09  10.2.2.5
224.0.1.40       Loopback0                01:45:22  00:02:32  10.3.3.4
```

The following is sample output from the **show ip mroute**command. In this example, the command is issued to confirm that the group entries defined in the class map named static (the class map configured in the Example: Configuring IGMP Static Group Support, on page 525 section) were added to the mroute table.

```
Device# show ip mroute

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.1.1.10, 232.7.7.7), 00:00:17/00:02:42, flags: sTI
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.5
  Outgoing interface list:
    FastEthernet3/1, Forward/Sparse-Dense, 00:00:17/00:02:42
(10.1.1.10, 232.7.7.9), 00:00:17/00:02:42, flags: sTI
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.5
  Outgoing interface list:
    FastEthernet3/1, Forward/Sparse-Dense, 00:00:17/00:02:42
(10.1.1.10, 232.7.7.8), 00:00:18/00:02:41, flags: sTI
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.5
  Outgoing interface list:
    FastEthernet3/1, Forward/Sparse-Dense, 00:00:18/00:02:41
(*, 227.7.7.7), 00:00:18/00:02:41, RP 10.2.2.6, flags: SJC
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.6
  Outgoing interface list:
    FastEthernet3/1, Forward/Sparse-Dense, 00:00:18/00:02:41
(*, 227.7.7.9), 00:00:18/00:02:41, RP 10.2.2.6, flags: SJC
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.6
  Outgoing interface list:
    FastEthernet3/1, Forward/Sparse-Dense, 00:00:18/00:02:41
(*, 227.7.7.8), 00:00:18/00:02:41, RP 10.2.2.6, flags: SJC
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.6
  Outgoing interface list:
    FastEthernet3/1, Forward/Sparse-Dense, 00:00:18/00:02:41
```

```
(*, 224.0.1.40), 00:01:40/00:02:23, RP 10.2.2.6, flags: SJCL
  Incoming interface: FastEthernet3/2, RPF nbr 10.2.2.6
  Outgoing interface list:
    Loopback0, Forward/Sparse-Dense, 00:01:40/00:02:23
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2933	*Internet Group Management Protocol MIB*

MIBs

MIB	MIBs Link
IGMP-MIB	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IGMP Static Group Range Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 29: Feature Information for IGMP Static Group Range Support

Feature Name	Releases	Feature Information
IGMP Static Group Range Support	12.2(18)SXF5 Cisco IOS XE Release 2.6 15.0(1)M 12.2(33)SRE 15.1(1)SG Cisco IOS XE Release 3.3SG	The IGMP Static Group Range Support feature introduces the capability to configure group ranges in class maps and attach class maps to an interface. This feature is an enhancement that simplifies the administration of networks with devices that require static group membership entries on many interfaces. The following commands were introduced or modified by this feature: **class-map type multicast-flows**, **group** (multicast-flows), **ip igmp static-group** , **show ip igmp static-group class-map**.
IGMP MIB Support Enhancements for SNMP	12.2(11)T 12.2(33)SRE Cisco IOS XE Release 2.1 15.1(1)SG 12.2(50)SY 15.0(1)S	The Internet Group Management Protocol (IGMP) is used by IP hosts to report their multicast group memberships to neighboring multicast routers. The IGMP MIB describes objects that enable users to remotely monitor and configure IGMP using Simple Network Management Protocol (SNMP). It also allows users to remotely subscribe and unsubscribe from multicast groups. The IGMP MIB Support Enhancements for SNMP feature adds full support of RFC 2933 (Internet Group Management Protocol MIB) in Cisco IOS software. There are no new or modfied commands for this feature.

CHAPTER 40

SSM Mapping

The Source Specific Multicast (SSM) Mapping feature extends the Cisco suite of SSM transition tools, which also includes URL Rendezvous Directory (URD) and Internet Group Management Protocol Version 3 Lite (IGMP v3lite). SSM mapping supports SSM transition in cases where neither URD nor IGMP v3lite is available, or when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. SSM mapping enables you to leverage SSM for video delivery to legacy set-top boxes (STBs) that do not support IGMPv3 or for applications that do not take advantage of the IGMPv3 host stack.

Prerequisites for SSM Mapping

One option available for using SSM mapping is to install it together with a Domain Name System (DNS) server to simplify administration of the SSM Mapping feature in larger deployments.

Before you can configure and use SSM mapping with DNS lookups, you need to add records to a running DNS server. If you do not already have a DNS server running, you need to install one.

Restrictions for SSM Mapping

- The SSM Mapping feature does not share the benefit of full SSM. SSM mapping takes a group G join from a host and identifies this group with an application associated with one or more sources, therefore, it can only support one such application per group G. Nevertheless, full SSM applications may still share the same group also used in SSM mapping.
- Enable IGMPv3 with care on the last hop router when you rely solely on SSM mapping as a transition solution for full SSM.

Information About SSM Mapping

SSM Components

SSM is a datagram delivery model that best supports one-to-many applications, also known as broadcast applications. SSM is a core networking technology for the Cisco implementation of IP multicast solutions targeted for audio and video broadcast application environments and is described in RFC 3569. The following two components together support the implementation of SSM:

- Protocol Independent Multicast source-specific mode (PIM-SSM)
- Internet Group Management Protocol Version 3 (IGMPv3)

Protocol Independent Multicast (PIM) SSM, or PIM-SSM, is the routing protocol that supports the implementation of SSM and is derived from PIM sparse mode (PIM-SM). IGMP is the Internet Engineering Task Force (IETF) standards track protocol used for hosts to signal multicast group membership to routers. IGMP Version 3 supports source filtering, which is required for SSM. IGMP For SSM to run with IGMPv3, SSM must be supported in the router, the host where the application is running, and the application itself.

Benefits of Source Specific Multicast

IP Multicast Address Management Not Required

In the ISM service, applications must acquire a unique IP multicast group address because traffic distribution is based only on the IP multicast group address used. If two applications with different sources and receivers use the same IP multicast group address, then receivers of both applications will receive traffic from the senders of both applications. Even though the receivers, if programmed appropriately, can filter out the unwanted traffic, this situation would cause generally unacceptable levels of unwanted traffic.

Allocating a unique IP multicast group address for an application is still a problem. Most short-lived applications use mechanisms like Session Description Protocol (SDP) and Session Announcement Protocol (SAP) to get a random address, a solution that does not work well with a rising number of applications in the Internet. The best current solution for long-lived applications is described in RFC 2770, but this solution suffers from the restriction that each autonomous system is limited to only 255 usable IP multicast addresses.

In SSM, traffic from each source is forwarded between routers in the network independent of traffic from other sources. Thus different sources can reuse multicast group addresses in the SSM range.

Denial of Service Attacks from Unwanted Sources Inhibited

In SSM, multicast traffic from each individual source will be transported across the network only if it was requested (through IGMPv3, IGMP v3lite, or URD memberships) from a receiver. In contrast, ISM forwards traffic from any active source sending to a multicast group to all receivers requesting that multicast group. In Internet broadcast applications, this ISM behavior is highly undesirable because it allows unwanted sources to easily disturb the actual Internet broadcast source by simply sending traffic to the same multicast group. This situation depletes bandwidth at the receiver side with unwanted traffic and thus disrupts the undisturbed reception of the Internet broadcast. In SSM, this type of denial of service (DoS) attack cannot be made by simply sending traffic to a multicast group.

Easy to Install and Manage

SSM is easy to install and provision in a network because it does not require the network to maintain which active sources are sending to multicast groups. This requirement exists in ISM (with IGMPv1, IGMPv2, or IGMPv3).

The current standard solutions for ISM service are PIM-SM and MSDP. Rendezvous point (RP) management in PIM-SM (including the necessity for Auto-RP or BSR) and MSDP is required only for the network to learn about active sources. This management is not necessary in SSM, which makes SSM easier than ISM to install and manage, and therefore easier than ISM to operationally scale in deployment. Another factor that contributes to the ease of installation of SSM is the fact that it can leverage preexisting PIM-SM networks and requires only the upgrade of last hop routers to support IGMPv3, IGMP v3lite, or URD.

Ideal for Internet Broadcast Applications

The three benefits previously described make SSM ideal for Internet broadcast-style applications for the following reasons:

- The ability to provide Internet broadcast services through SSM without the need for unique IP multicast addresses allows content providers to easily offer their service (IP multicast address allocation has been a serious problem for content providers in the past).
- The prevention against DoS attacks is an important factor for Internet broadcast services because, with their exposure to a large number of receivers, they are the most common targets for such attacks.
- The ease of installation and operation of SSM makes it ideal for network operators, especially in those cases where content needs to be forwarded between multiple independent PIM domains (because there is no need to manage MSDP for SSM between PIM domains).

SSM Transition Solutions

The Cisco IOS suite of SSM transition solutions consists of the following transition solutions that enable the immediate development and deployment of SSM services, without the need to wait for the availability of full IGMPv3 support in host operating systems and SSM receiver applications:

- Internet Group Management Protocol Version 3 lite (IGMP v3lite)
- URL Rendezvous Directory (URD)
- SSM mapping

IGMP v3lite is a solution for application developers that allows immediate development of SSM receiver applications switching to IGMPv3 as soon as it becomes available.

For more information about IGMP v3lite, see the “ Configuring Source Specific Multicast ” module.

URD is an SSM transition solution for content providers and content aggregators that allows them to deploy receiver applications that are not yet SSM enabled (through support for IGMPv3) by enabling the receiving applications to be started and controlled through a web browser.

For more information about URD, see the see the “ Configuring Source Specific Multicast ” module.

SSM mapping supports SSM transition in cases where neither URD nor IGMP v3lite are available, or when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons.

SSM Mapping Overview

SSM mapping supports SSM transition when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. Using SSM to deliver live streaming video to legacy STBs that do not support IGMPv3 is a typical application of SSM mapping.

In a typical STB deployment, each TV channel uses one separate IP multicast group and has one active server host sending the TV channel. A single server may of course send multiple TV channels, but each to a different group. In this network environment, if a router receives an IGMPv1 or IGMPv2 membership report for a particular group G, the report implicitly addresses the well-known TV server for the TV channel associated with the multicast group.

SSM mapping introduces a means for the last hop router to discover sources sending to groups. When SSM mapping is configured, if a router receives an IGMPv1 or IGMPv2 membership report for a particular group G, the router translates this report into one or more (S, G) channel memberships for the well-known sources associated with this group.

When the router receives an IGMPv1 or IGMPv2 membership report for group G, the router uses SSM mapping to determine one or more source IP addresses for group G. SSM mapping then translates the membership report as an IGMPv3 report INCLUDE (G, [S1, G], [S2, G]...[Sn, G] and continues as if it had received an IGMPv3 report. The router then sends out PIM joins toward (S1, G) to (Sn, G) and continues to be joined to these groups as long as it continues to receive the IGMPv1 or IGMPv2 membership reports and as long as the SSM mapping for the group remains the same. SSM mapping, thus, enables you to leverage SSM for video delivery to legacy STBs that do not support IGMPv3 or for applications that do not take advantage of the IGMPv3 host stack.

SSM mapping enables the last hop router to determine the source addresses either by a statically configured table on the router or by consulting a DNS server. When the statically configured table is changed, or when the DNS mapping changes, the router will leave the current sources associated with the joined groups.

Static SSM Mapping

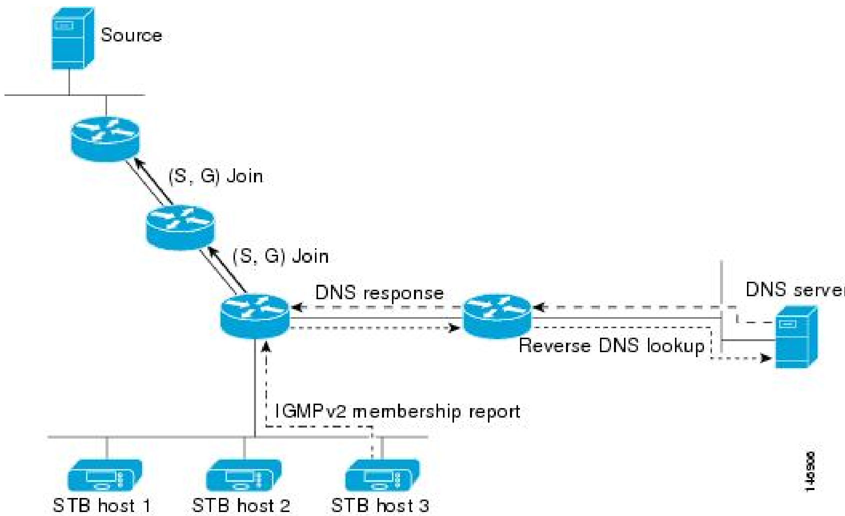
SSM static mapping enables you to configure the last hop router to use a static map to determine the sources sending to groups. Static SSM mapping requires that you configure access lists (ACLs) to define group ranges. The groups permitted by those ACLs then can be mapped to sources using the **ip igmp static ssm-map** global configuration command.

You can configure static SSM mapping in smaller networks when a DNS is not needed or to locally override DNS mappings that may be temporarily incorrect. When configured, static SSM mappings take precedence over DNS mappings.

DNS-Based SSM Mapping

DNS-based SSM mapping enables you to configure the last hop router to perform a reverse DNS lookup to determine sources sending to groups (see the figure below). When DNS-based SSM mapping is configured, the router constructs a domain name that includes the group address G and performs a reverse lookup into the DNS. The router looks up IP address resource records (IP A RRs) to be returned for this constructed domain name and uses the returned IP addresses as the source addresses associated with this group. SSM mapping supports up to 20 sources for each group. The router joins all sources configured for a group.

Figure 58: DNS-Based SSM-Mapping



The SSM mapping mechanism that enables the last hop router to join multiple sources for a group can be used to provide source redundancy for a TV broadcast. In this context, the redundancy is provided by the last hop router using SSM mapping to join two video sources simultaneously for the same TV channel. However, to prevent the last hop router from duplicating the video traffic, it is necessary that the video sources utilize a server-side switchover mechanism where one video source is active while the other backup video source is passive. The passive source waits until an active source failure is detected before sending the video traffic for the TV channel. The server-side switchover mechanism, thus, ensures that only one of the servers is actively sending the video traffic for the TV channel.

To look up one or more source addresses for a group G that includes G1, G2, G3, and G4, the following DNS resource records (RRs) must be configured on the DNS server:

G4.G3.G2.G1 [*multicast-domain*] [*timeout*]	IN A *source-address-1*
	IN A *source-address-2*
	IN A *source-address-n*

The *multicast-domain* argument is a configurable DNS prefix. The default DNS prefix is in-addr.arpa. You should only use the default prefix when your installation is either separate from the internet or if the group names that you map are global scope group addresses (RFC 2770 type addresses that you configure for SSM) that you own.

The *timeout* argument configures the length of time for which the router performing SSM mapping will cache the DNS lookup. This argument is optional and defaults to the timeout of the zone in which this entry is configured. The timeout indicates how long the router will keep the current mapping before querying the DNS server for this group. The timeout is derived from the cache time of the DNS RR entry and can be configured for each group/source entry on the DNS server. You can configure this time for larger values if you want to minimize the number of DNS queries generated by the router. Configure this time for a low value if you want to be able to quickly update all routers with new source addresses.

Note Refer to your DNS server documentation for more information about configuring DNS RRs.

To configure DNS-based SSM mapping in the software, you must configure a few global commands but no per-channel specific configuration is needed. There is no change to the configuration for SSM mapping if additional channels are added. When DNS-based SSM mapping is configured, the mappings are handled entirely by one or more DNS servers. All DNS techniques for configuration and redundancy management can be applied to the entries needed for DNS-based SSM mapping.

SSM Mapping Benefits

- The SSM Mapping feature provides almost the same ease of network installation and management as a pure SSM solution based on IGMPv3. Some additional configuration is necessary to enable SSM mapping.
- The SSM benefit of inhibition of DoS attacks applies when SSM mapping is configured. When SSM mapping is configured the only segment of the network that may still be vulnerable to DoS attacks are receivers on the LAN connected to the last hop router. Since those receivers may still be using IGMPv1 and IGMPv2, they are vulnerable to attacks from unwanted sources on the same LAN. SSM mapping, however, does protect those receivers (and the network path leading towards them) from multicast traffic from unwanted sources anywhere else in the network.
- Address assignment within a network using SSM mapping needs to be coordinated, but it does not need assignment from outside authorities, even if the content from the network is to be transited into other networks.

How to Configure SSM Mapping

Configuring Static SSM Mapping

Perform this task to configure the last hop router in an SSM deployment to use static SSM mapping to determine the IP addresses of sources sending to groups.

Before you begin

- Enable IP multicast routing, enable PIM sparse mode, and configure SSM before performing this task. For more information, see the “Configuring Basic Multicast ”module.
- Before you configure static SSM mapping, you must configure ACLs that define the group ranges to be mapped to source addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp ssm-map enable**
4. **no ip igmp ssm-map query dns**
5. **ip igmp ssm-map static** *access-list source-address*
6. Repeat Step 5 to configure additional static SSM mappings, if required.
7. **end**
8. **show running-config**
9. **copy running-config start-up config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp ssm-map enable** **Example:** `Device(config)# ip igmp ssm-map enable`	Enables SSM mapping for groups in the configured SSM range. **Note** By default, this command enables DNS-based SSM mapping.
Step 4	**no ip igmp ssm-map query dns** **Example:** `Device(config)# no ip igmp ssm-map query dns`	(Optional) Disables DNS-based SSM mapping. **Note** Disable DNS-based SSM mapping if you only want to rely on static SSM mapping. By default, the **ip igmp ssm-map** command enables DNS-based SSM mapping.
Step 5	**ip igmp ssm-map static** *access-list source-address* **Example:** `Device(config)# ip igmp ssm-map static 11 172.16.8.11`	Configures static SSM mapping. • The ACL supplied for the *access-list* argument defines the groups to be mapped to the source IP address entered for the *source-address* argument. **Note** You can configure additional static SSM mappings. If additional SSM mappings are configured and the router receives an IGMPv1 or IGMPv2 membership report for a group in the SSM range, the Cisco IOS XE software determines the source addresses associated with the group by walking each configured **ip igmp ssm-map static** command. The Cisco IOS XE software associates up to 20 sources per group.
Step 6	Repeat Step 5 to configure additional static SSM mappings, if required.	--
Step 7	**end** **Example:** `Device(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 9	**copy running-config start-up config** **Example:** `Device# copy running-config start-up config`	(Optional) Saves your entries in the configuration file.

What to Do Next

Proceed to the Configuring DNS-Based SSM Mapping (CLI) or to the Verifying SSM Mapping Configuration and Operation.

Configuring DNS-Based SSM Mapping (CLI)

Perform this task to configure the last hop router to perform DNS lookups to learn the IP addresses of sources sending to a group.

Before you begin

- Enable IP multicast routing, enable PIM sparse mode, and configure SSM before performing this task. For more information, see the "Configuring Basic Multicast"module.
- Before you can configure and use SSM mapping with DNS lookups, you need to be able to add records to a running DNS server. If you do not already have a DNS server running, you need to install one.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp ssm-map enable**
4. **ip igmp ssm-map query dns**
5. **ip domain multicast** *domain-prefix*
6. **ipname-server** *server-address1* [*server-address2server-address6*]
7. Repeat Step 6 to configure additional DNS servers for redundancy, if required.
8. **end**
9. **show running-config**
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
	`Device# enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp ssm-map enable** **Example:** `Device(config)# ip igmp ssm-map enable`	Enables SSM mapping for groups in a configured SSM range.
Step 4	**ip igmp ssm-map query dns** **Example:** `Device(config)# ip igmp ssm-map query dns`	(Optional) Enables DNS-based SSM mapping. • By default, the **ip igmp ssm-map** command enables DNS-based SSM mapping. Only the **no**form of this command is saved to the running configuration. **Note** Use this command to reenable DNS-based SSM mapping if DNS-based SSM mapping is disabled.
Step 5	**ip domain multicast** *domain-prefix* **Example:** `Device(config)# ip domain multicast ssm-map.cisco.com`	(Optional) Changes the domain prefix used by the Cisco IOS XE software for DNS-based SSM mapping. • By default, the software uses the ip-addr.arpa domain prefix.
Step 6	**ipname-server** *server-address1* [*server-address2server-address6*] **Example:** `Device(config)# ip name-server 10.48.81.21`	Specifies the address of one or more name servers to use for name and address resolution.
Step 7	Repeat Step 6 to configure additional DNS servers for redundancy, if required.	--
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 9	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 10	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

What to Do Next

Configuring Static Traffic Forwarding with SSM Mapping

Perform this task to configure static traffic forwarding with SSM mapping on the last hop router. Static traffic forwarding can be used in conjunction with SSM mapping to statically forward SSM traffic for certain groups. When static traffic forwarding with SSM mapping is configured, the last hop router uses DNS-based SSM mapping to determine the sources associated with a group. The resulting (S, G) channels are then statically forwarded.

Before you begin

This task does not include the steps for configuring DNS-based SSM mapping. See the Configuring DNS-Based SSM Mapping (CLI), on page 536 task for more information about configuring DNS-based SSM mapping.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp static-group** *group-address* **source ssm-map**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Selects an interface on which to statically forward traffic for a multicast group using SSM mapping and enters interface configuration mode. **Note** Static forwarding of traffic with SSM mapping works with either DNS-based SSM mapping or statically-configured SSM mapping.
Step 4	**ip igmp static-group** *group-address* **source ssm-map** **Example:** `Router(config-if)# ip igmp static-group 232.1.2.1 source ssm-map`	Configures SSM mapping to be used to statically forward a (S, G) channel out of the interface. • Use this command if you want to statically forward SSM traffic for certain groups, but you want to use DNS-based SSM mapping to determine the source addresses of the channels.

What to Do Next

Proceed to the Verifying SSM Mapping Configuration and Operation.

Verifying SSM Mapping Configuration and Operation

Perform this optional task to verify SSM mapping configuration and operation.

SUMMARY STEPS

1. **enable**
2. **show ip igmp ssm-mapping**
3. **show ip igmp ssm-mapping** *group-address*
4. **show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**]
5. **show host**
6. **debug ip igmp** *group-address*

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
> enable
```

Step 2 **show ip igmp ssm-mapping**

(Optional) Displays information about SSM mapping.

The following example shows how to display information about SSM mapping configuration. In this example, SSM static mapping and DNS-based SSM mapping are enabled.

Example:

```
# show ip igmp ssm-mapping
SSM Mapping  : Enabled
DNS Lookup   : Enabled
Mcast domain : ssm-map.cisco.com
Name servers : 10.0.0.3
               10.0.0.4
```

Step 3 **show ip igmp ssm-mapping** *group-address*

(Optional) Displays the sources that SSM mapping uses for a particular group.

The following example shows how to display information about the configured DNS-based SSM mapping. In this example, the router has used DNS-based mapping to map group 232.1.1.4 to sources 172.16.8.5 and 172.16.8.6. The timeout for this entry is 860000 milliseconds (860 seconds).

Example:

```
# show ip igmp ssm-mapping 232.1.1.4
```

```
Group address: 232.1.1.4
Database     : DNS
DNS name     : 4.1.1.232.ssm-map.cisco.com
Expire time  : 860000
Source list  : 172.16.8.5
             : 172.16.8.6
```

Step 4 **show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**]

(Optional) Displays the multicast groups with receivers that are directly connected to the router and that were learned through IGMP.

The following is sample output from the **show ip igmp groups** command with the *group-address* argument and **detail** keyword. In this example the "M" flag indicates that SSM mapping is configured.

Example:

```
# show ip igmp group 232.1.1.4 detail
            Interface:      GigabitEthernet2/0/0
            Group:          232.1.1.4 SSM
            Uptime:         00:03:20
            Group mode:     INCLUDE
            Last reporter:  0.0.0.0
            CSR Grp Exp:    00:02:59
            Group source list: (C - Cisco Src Report, U - URD, R - Remote,
                                 S - Static, M - SSM Mapping)
              Source Address   Uptime    v3 Exp   CSR Exp   Fwd  Flags
              172.16.8.3       00:03:20  stopped   00:02:59  Yes  CM
              172.16.8.4       00:03:20  stopped   00:02:59  Yes  CM
              172.16.8.5       00:03:20  stopped   00:02:59  Yes  CM
              172.16.8.6       00:03:20  stopped   00:02:59  Yes  CM
```

Step 5 **show host**

(Optional) Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of hostnames and addresses.

The following is sample output from the **show host**command. Use this command to display DNS entries as they are learned by the router.

Example:

```
# show host
Default domain is cisco.com
Name/address lookup uses domain service
Name servers are 10.48.81.21
Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined
Host                      Port   Flags      Age   Type    Address(es)
10.0.0.0.ssm-map.cisco.c  None  (temp, OK)   0     IP     172.16.8.5
                                                          172.16.8.6
                                                          172.16.8.3
```

172.16.8.4

Step 6 **debug ip igmp** *group-address*

(Optional) Displays the IGMP packets received and sent and IGMP host-related events.

The following is sample output from the **debug ip igmp**command when SSM static mapping is enabled. The following output indicates that the router is converting an IGMPv2 join for group G into an IGMPv3 join:

Example:

```
IGMP(0): Convert IGMPv2 report (*,232.1.2.3) to IGMPv3 with 2 source(s) using STATIC.
```

The following is sample output from the **debug ip igmp** command when DNS-based SSM mapping is enabled. The following output indicates that a DNS lookup has succeeded:

Example:

```
IGMP(0): Convert IGMPv2 report (*,232.1.2.3) to IGMPv3 with 2 source(s) using DNS.
```

The following is sample output from the **debug ip igmp** command when DNS-based SSM mapping is enabled and a DNS lookup has failed:

IGMP(0): DNS source lookup failed for (*, 232.1.2.3), IGMPv2 report failed

Configuration Examples for SSM Mapping

SSM Mapping Example

The following configuration example shows a router configuration for SSM mapping. This example also displays a range of other IGMP and SSM configuration options to show compatibility between features. Do not use this configuration example as a model unless you understand all of the features used in the example.

> **Note** Address assignment in the global SSM range 232.0.0.0/8 should be random. If you copy parts or all of this sample configuration, make sure to select a random address range but not 232.1.1.x as shown in this example. Using a random address range minimizes the possibility of address collision and may prevent conflicts when other SSM content is imported while SSM mapping is used.

```
!
no ip domain lookup
ip domain multicast ssm.map.cisco.com
ip name-server 10.48.81.21
!
!
ip multicast-routing distributed
ip igmp ssm-map enable
ip igmp ssm-map static 10 172.16.8.10
ip igmp ssm-map static 11 172.16.8.11
!
!
.
.
.
!
interface GigabitEthernet0/0/0
 description Sample IGMP Interface Configuration for SSM-Mapping Example
 ip address 10.20.1.2 255.0.0.0
 ip pim sparse-mode
 ip igmp last-member-query-interval 100
 ip igmp static-group 232.1.2.1 source ssm-map
```

```
 ip igmp version 3
 ip igmp explicit-tracking
 ip igmp limit 2
 ip igmp v3lite
 ip urd
!
.
.
.
!
ip pim ssm default
!
access-list 10 permit 232.1.2.10
access-list 11 permit 232.1.2.0 0.0.0.255
!
```

This table describes the significant commands shown in the SSM mapping configuration example.

Table 30: SSM Mapping Configuration Example Command Descriptions

Command	Description
no ip domain lookup	Disables IP DNS-based hostname-to-address translation. **Note** The **no ip domain-list** command is shown in the configuration only to demonstrate that disabling IP DNS-based hostname-to-address translation does not conflict with configuring SSM mapping. If this command is enabled, the Cisco IOS XE software will try to resolve unknown strings as hostnames.
ip domain multicast ssm-map.cisco.com	Specifies ssm-map.cisco.com as the domain prefix for SSM mapping.
ip name-server 10.48.81.21	Specifies 10.48.81.21 as the IP address of the DNS server to be used by SSM mapping and any other service in the software that utilizes DNS.
ip multicast-routing	Enables IP multicast routing.
ip igmp ssm-map enable	Enables SSM mapping.
ip igmp ssm-map static 10 172.16.8.10	Configures the groups permitted by ACL 10 to use source address 172.16.8.10. • In this example, ACL 10 permits all groups in the 232.1.2.0/25 range except 232.1.2.10.
ip igmp ssm-map static 11 172.16.8.11	Configures the groups permitted by ACL 11 to use source address 172.16.8.11. • In this example, ACL 11 permits group 232.1.2.10.
ip pim sparse-mode	Enables PIM sparse mode.

Command	Description
ip igmp last-member-query-interval 100	Reduces the leave latency for IGMPv2 hosts. **Note** This command is not required for configuring SSM mapping; however, configuring this command can be beneficial for IGMPv2 hosts relying on SSM mapping.
ip igmp static-group 232.1.2.1 source ssm-map	Configures SSM mapping to be used to determine the sources associated with group 232.1.2.1. The resulting (S, G) channels are statically forwarded.
ip igmp version 3	Enables IGMPv3 on this interface. **Note** This command is shown in the configuration only to demonstrate that IGMPv3 can be configured simultaneously with SSM mapping; however, it is not required.
ip igmp explicit-tracking	Minimizes the leave latency for IGMPv3 host leaving a multicast channel. **Note** This command is not required for configuring SSM mapping.
ip igmp limit 2	Limits the number of IGMP states resulting from IGMP membership states on a per-interface basis. **Note** This command is not required for configuring SSM mapping.
ip igmp v3lite	Enables the acceptance and processing of IGMP v3lite membership reports on this interface. **Note** This command is shown in the configuration only to demonstrate that IGMP v3lite can be configured simultaneously with SSM mapping; however, it is not required.
ip urd	Enables interception of TCP packets sent to the reserved URD port 465 on an interface and processing of URD channel subscription reports. **Note** This command is shown in the configuration only to demonstrate that URD can be configured simultaneously with SSM mapping; however, it is not required.
ip pim ssm default	Configures SSM service. The **default** keyword defines the SSM range access list as 232/8.
access-list 10 permit 232.1.2.10 **access-list 11 permit 232.1.2.0 0.0.0.255**	Configures the ACLs to be used for static SSM mapping. **Note** These are the ACLs that are referenced by the **ip igmp ssm-map static** commands in this configuration example.

DNS Server Configuration Example

To configure DNS-based SSM mapping, you need to create a DNS server zone or add records to an existing zone. If the routers that are using DNS-based SSM mapping are also using DNS for other purposes besides SSM mapping, you should use a normally-configured DNS server. If DNS-based SSM mapping is the only DNS implementation being used on the router, you can configure a fake DNS setup with an empty root zone, or a root zone that points back to itself.

The following example shows how to create a zone and import the zone data using Network Registrar:

```
Router> zone 1.1.232.ssm-map.cisco.com. create primary file=named.ssm-map
100 Ok
Router> dns reload
100 Ok
```

The following example shows how to import the zone files from a named.conf file for BIND 8:

```
Router> ::import named.conf /etc/named.conf
Router> dns reload
100 Ok:
```

Note Network Registrar version 8.0 and later support import BIND 8 format definitions.

Additional References

Related Documents

Related Topic	Document Title
SSM concepts and configuration	"Configuring Basic IP Multicast" module
Cisco IOS IP multicast commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2365	*Administratively Scoped IP Multicast*
RFC 2770	*GLOP Addressing in 233/8*
RFC 3569	*An Overview of Source-Specific Multicast*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for SSM Mapping

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 31: Feature Information for SSM Mapping

Feature Name	Releases	Feature Information
Source Specific Multicast (SSM) Mapping	12.3(2)T 12.2(18)S 12.2(18)SXD3 12.2(27)SBC 15.0(1)S Cisco IOS XE 3.1.0SG	This feature was introduced. The following commands were introduced or modified: **debug ip igmp, ip domain multicast, ip igmp ssm-map enable, ip igmp ssm-map query dns, ip igmp ssm-map static, ip igmp static-group, show ip igmp groups, show ip igmp ssm-mapping.**

CHAPTER 41

IGMP Snooping

This module describes how to enable and configure the Ethernet Virtual Connection (EVC)-based IGMP Snooping feature globally and on bridge domains.

Information About IGMP Snooping

IGMP Snooping

Multicast traffic becomes flooded because a device usually learns MAC addresses by looking into the source address field of all the frames that it receives. A multicast MAC address is never used as the source address for a packet. Such addresses do not appear in the MAC address table, and the device has no method for learning them.

IP Multicast Internet Group Management Protocol (IGMP), which runs at Layer 3 on a multicast device, generates Layer 3 IGMP queries in subnets where the multicast traffic must be routed. IGMP (on a device) sends out periodic general IGMP queries.

IGMP Snooping is an Ethernet Virtual Circuit (EVC)-based feature set. EVC decouples the concept of VLAN and broadcast domain. An EVC is an end-to-end representation of a single instance of a Layer 2 service being offered by a provider. In the Cisco EVC framework, bridge domains are made up of one or more Layer 2 interfaces known as service instances. A service instance is the instantiation of an EVC on a given port on a given device. A service instance is associated with a bridge domain based on the configuration.

Traditionally, a VLAN is a broadcast domain, and physical ports are assigned to VLANs as access ports; the VLAN tag in a packet received by a trunk port is the same number as the internal VLAN broadcast domain. With EVC, an Ethernet Flow Point (EFP) is configured and associated with a broadcast domain. The VLAN tag is used to identify the EFP only and is no longer used to identify the broadcast domain.

When you enable EVC-based IGMP snooping on a bridge domain, the bridge domain interface responds at Layer 2 to the IGMP queries with only one IGMP join request per Layer 2 multicast group. Each bridge domain represents a Layer 2 broadcast domain. The bridge domain interface creates one entry per subnet in the Layer 2 forwarding table for each Layer 2 multicast group from which it receives an IGMP join request. All hosts interested in this multicast traffic send IGMP join requests and are added to the forwarding table entry. During a Layer 2 lookup on a bridge domain to which the bridge domain interface belongs, the bridge domain forwards the packets to the correct EFP. When the bridge domain interface hears the IGMP Leave group message from a host, it removes the table entry of the host.

Layer 2 multicast groups learned through IGMP snooping are dynamic. However, you can statically configure Layer 2 multicast groups. If you specify group membership for a multicast group address statically, your static setting supersedes any automatic manipulation by IGMP snooping. Multicast group membership lists can consist of both user-defined and IGMP snooping-learned-settings.

Restrictions for IGMP Snooping

- IGMP snooping is only supported on a Bridge Domain when OTV is enabled on ASR 1000 routers.
- If IGMP snooping is configured on a Bridge Domain with OTV enabled, then the IGMP snooping process limits the multicast traffic. In this scenario, the snooping tables are populated.
- If IGMP snooping is configured on a Bridge Domain without OTV, the IGMP snooping process does not limit multicast traffic. In this scenario, the snooping tables are not populated and the multicast traffic floods the entire VLAN.

How to Configure IGMP Snooping

Enabling IGMP Snooping

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping**
4. **bridge-domain** *bridge-id*
5. **ip igmp snooping**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip igmp snooping** **Example:** `Device(config)# ip igmp snooping`	Globally enables IGMP snooping after it has been disabled.
Step 4	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 100`	(Optional) Enters bridge domain configuration mode.
Step 5	**ip igmp snooping** **Example:** `Device(config-bdomain)# ip igmp snooping`	(Optional) Enables IGMP snooping on the bridge domain interface being configured. • Required only if IGMP snooping was previously explicitly disabled on the specified bridge domain.
Step 6	**end** **Example:** `Device(config-bdomain)# end`	Returns to privileged EXEC mode.

Configuring IGMP Snooping Globally

Perform this task to modify the global configuration for IGMP snooping.

Before you begin

IGMP snooping must be enabled. IGMP snooping is enabled by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping robustness-variable** *variable*
4. **ip igmp snooping tcn query solicit**
5. **ip igmp snooping tcn flood query count** *count*
6. **ip igmp snooping report-suppression**
7. **ip igmp snooping explicit-tracking-limit** *limit*
8. **ip igmp snooping last-member-query-count** *count*
9. **ip igmp snooping last-member-query-interval** *interval*
10. **ip igmp snooping check** { **ttl** | **rtr-alert-option** }
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping robustness-variable** *variable* **Example:** `Device(config)# ip igmp snooping robustness-variable 3`	(Optional) Configures IGMP snooping robustness variable.
Step 4	**ip igmp snooping tcn query solicit** **Example:** `Device(config)# ip igmp snooping tcn query solicit`	(Optional) Enables device to send TCN query solicitation even if it is not the spanning-tree root.
Step 5	**ip igmp snooping tcn flood query count** *count* **Example:** `Device(config)# ip igmp snooping tcn flood query count 4`	(Optional) Configures the TCN flood query count for IGMP snooping.
Step 6	**ip igmp snooping report-suppression** **Example:** `Device(config)# ip igmp snooping report-suppression`	(Optional) Enables report suppression for IGMP snooping.
Step 7	**ip igmp snooping explicit-tracking-limit** *limit* **Example:** `Device(config)# ip igmp snooping explicit-tracking-limit 200`	(Optional) Limits the number of reports in the IGMP snooping explicit-tracking database.
Step 8	**ip igmp snooping last-member-query-count** *count* **Example:** `Device (config)# ip igmp snooping last-member-query-count 5`	(Optional) Configures how often Internet Group Management Protocol (IGMP) snooping will send query messages in response to receiving an IGMP leave message. The default is 2 milliseconds.

	Command or Action	Purpose
Step 9	**ip igmp snooping last-member-query-interval** *interval* **Example:** `Device (config)# ip igmp snooping last-member-query-interval 200`	(Optional) Configures the length of time after which the group record is deleted if no reports are received. The default is 1000 milliseconds.
Step 10	**ip igmp snooping check** { **ttl** | **rtr-alert-option** } **Example:** `Device (config)# ip igmp snooping check ttl`	(Optional) Enforces IGMP snooping check.
Step 11	**exit** **Example:** `Device (config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring IGMP Snooping on a Bridge Domain Interface

Perform this task to modify the IGMP snooping configuration on a bridge domain interface.

Before you begin

- The bridge domain interface must be created. See the "Configuring Bridge Domain Interfaces" section of the *Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide*.
- IGMP snooping must be enabled on the interface to be configured. IGMP snooping is enabled by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge-domain** *bridge-id*
4. **ip igmp snooping immediate-leave**
5. **ip igmp snooping robustness-variable** *variable*
6. **ip igmp snooping report-suppression**
7. **ip igmp snooping explicit-tracking**
8. **ip igmp snooping explicit-tracking-limit** *limit*
9. **ip igmp snooping last-member-query-count** *count*
10. **ip igmp snooping last-member-query-interval** *interval*
11. **ip igmp snooping access-group** {*acl-number* | *acl-name*}
12. **ip igmp snooping limit** *num* [**except** {*acl-number* | *acl-name*}]
13. **ip igmp snooping minimum-version** {**2** | **3**}
14. **ip igmp snooping check** { **ttl** | **rtr-alert-option** }
15. **ip igmp snooping static source** *source-address* **interface** *port-type port-number*
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 100`	Enters bridge domain configuration mode.
Step 4	**ip igmp snooping immediate-leave** **Example:** `Device(config-bdomain)# ip igmp snooping immediate-leave`	(Optional) Enables IGMPv2 immediate-leave processing. **Note** When both immediate-leave processing and the query count are configured, fast-leave processing takes precedence.
Step 5	**ip igmp snooping robustness-variable** *variable* **Example:** `Device(config-bdomain)# ip igmp snooping robustness-variable 3`	(Optional) Configures the IGMP snooping robustness variable. The default is 2.
Step 6	**ip igmp snooping report-suppression** **Example:** `Device(config-bdomain)# ip igmp snooping report-suppression`	(Optional) Enables report suppression for all hosts on the bridge domain interface.
Step 7	**ip igmp snooping explicit-tracking** **Example:** `Device(config-bdomain)#  ip igmp snooping explicit-tracking`	(Optional) Enables IGMP snooping explicit tracking. Explicit tracking is enabled by default.
Step 8	**ip igmp snooping explicit-tracking-limit** *limit* **Example:** `Device(config-bdomain)# ip igmp snooping explicit-tracking-limit 200`	(Optional) Limits the number of reports in the IGMP snooping explicit-tracking database.

	Command or Action	Purpose
Step 9	**ip igmp snooping last-member-query-count** *count* **Example:** `Device(config-bdomain)# ip igmp snooping last-member-query-count 5`	(Optional) Configures the interval for snooping query messages sent in response to receiving an IGMP leave message. The default is 2 milliseconds. **Note** When both immediate-leave processing and the query count are configured, fast-leave processing takes precedence.
Step 10	**ip igmp snooping last-member-query-interval** *interval* **Example:** `Device(config-bdomain)#  ip igmp snooping last-member-query-interval 2000`	(Optional) Configures the length of time after which the group record is deleted if no reports are received. The default is 1000 milliseconds.
Step 11	**ip igmp snooping access-group** {*acl-number* | *acl-name*} **Example:** `Device(config-bdomain)# ip igmp snooping access-group 1300`	Configures ACL-based filtering on a bridge domain.
Step 12	**ip igmp snooping limit** *num* [**except** {*acl-number* | *acl-name*}] **Example:** `Device(config-bdomain)# ip igmp snooping 4400 except test1`	(Optional) Limits the number of groups or channels allowed on a bridge domain.
Step 13	**ip igmp snooping minimum-version** {**2** | **3**} **Example:** `Device(config-bdomain)# ip igmp snooping minimum-version 2`	(Optional) Configures IGMP protocol filtering.
Step 14	**ip igmp snooping check** { **ttl** | **rtr-alert-option** } **Example:** `Device(config-bdomain)# ip igmp snooping check ttl`	(Optional) Enforces IGMP snooping check.
Step 15	**ip igmp snooping static source** *source-address* **interface** *port-type port-number* **Example:** `Device(config-bdomain)# ip igmp snooping static source 192.0.2.1 interface gigbitethernet 1/1/1`	(Optional) Configures a host statically for a Layer 2 LAN port.
Step 16	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-bdomain)# end`	

Configuring an EFP

Perform this task to configure IGMP snooping features on an EFP.

Before you begin

The EFP and bridge domain must be previously configured. Configuring a service instance on a Layer 2 port creates a pseudoport or Ethernet Flow Point (EFP) on which you configure Ethernet Virtual Connection (EVC) features. See the "Configuring Ethernet Virtual Connections on the Cisco ASR 1000 Router" section of the *Carrier Ethernet Configuration Guide* for configuration information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router-guard ip multicast efps**
4. **interface** *type number*
5. **service instance** *id* **ethernet**
6. **router-guard multicast**
7. **ip igmp snooping tcn flood**
8. **ip igmp snooping access-group** {*acl-number* | *acl-name*}
9. **ip igmp snooping limit** *num* [**except** {*acl-number* | *acl-name*}]
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router-guard ip multicast efps** **Example:** `Device(config)# router-guard ip multicast efps`	(Optional) Enables the router guard for all EFPs.
Step 4	**interface** *type number* **Example:**	(Optional) Specifies the bridge domain interface to be configured.

	Command or Action	Purpose
	`Device(config)# interface BDI100`	
Step 5	**service instance** *id* **ethernet** **Example:** `Device(config-if)# service instance 333 ethernet`	(Optional) Enters Ethernet service configuration mode for configuring the EFP.
Step 6	**router-guard multicast** **Example:** `Device(config-if-srv)# router-guard multicast`	(Optional) Configures a router guard on an EFP.
Step 7	**ip igmp snooping tcn flood** **Example:** `Device(config-if-srv)# no ip igmp snooping tcn flood`	(Optional) Disables TCN flooding on an EFP. TCN flooding is enabled by default.
Step 8	**ip igmp snooping access-group** {*acl-number* | *acl-name*} **Example:** `Device(config-if-srv)# ip igmp snooping access-group 44`	(Optional) Configures ACL-based filtering on an EFP.
Step 9	**ip igmp snooping limit** *num* [**except** {*acl-number* | *acl-name*}] **Example:** `Device(config-if-srv)# ip igmp snooping limit 1300 except test1`	(Optional) Limits the number of IGMP groups or channels allowed on an EFP.
Step 10	**end** **Example:** `Device(config-if-srv)# end`	Returns to privileged EXEC mode.

Verifying IGMP Snooping

SUMMARY STEPS

1. **enable**
2. **show igmp snooping** [**count** [**bd** *bd-id*]]
3. **show igmp snooping groups bd** *bd-id* [**count** | *ip-address* [**verbose**] [**hosts** | **sources** | **summary**]]
4. **show igmp snooping membership bd** *bd-id*
5. **show igmp snooping mrouter** [**bd** *bd-id*]

6. **show igmp snooping counters** [**bd** *bd-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show igmp snooping** [**count** [**bd** *bd-id*]] **Example:** `Device(config)# show igmp snooping`	Displays configuration for IGMP snooping, globally or by bridge domain.
Step 3	**show igmp snooping groups bd** *bd-id* [**count** | *ip-address* [**verbose**] [**hosts** | **sources** | **summary**]] **Example:** `Device(config)# show igmp snooping groups bd 100`	Displays snooping information for groups by bridge domain.
Step 4	**show igmp snooping membership bd** *bd-id* **Example:** `Device(config)# show igmp snooping membership bd 100`	Displays IGMPv3 host membership information.
Step 5	**show igmp snooping mrouter** [**bd** *bd-id*] **Example:** `Device(config)# show igmp snooping mrouter`	Displays multicast ports, globally or by bridge domain.
Step 6	**show igmp snooping counters** [**bd** *bd-id*] **Example:** `Device(config)# show snooping counters`	Displays IGMP snooping counters, globally or by bridge domain.

Additional References

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IGMP Snooping

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 32: Feature Information for Configuring IGMP Snooping

Feature Name	Releases	Feature Information
IGMP Snooping	Cisco IOS XE Release 3.5S 15.2(4)S	IGMP snooping is an IP multicast constraining mechanism based on the Ethernet Virtual Connection (EVC) infrastructure. IGMP snooping examines Layer 3 information (IGMP Join/Leave messages) in the IGMP packets sent between hosts and routers. The following commands were introduced or modified: **ip igmp snooping**, **ip igmp snooping check**, **ip igmp snooping explicit-track ing limit**, **ip igmp snooping immediate leave**, **ip igmp snooping last-member-query count**, **ip igmp snooping last-member-query interval**, **ip igmp snooping report-suppression**, **ip igmp snooping robustness-variable**, **ip igmp snooping static**, **ip igmp snooping tcn flood** (if-srv), **ip igmp snooping tcn flood query**, **ip igmp snooping tcn flood query solicit**, **router guard ip multicast efps**

CHAPTER 42

Constraining IP Multicast in a Switched Ethernet Network

This module describes how to configure devices to use the Cisco Group Management Protocol (CGMP) in switched Ethernet networks to control multicast traffic to Layer 2 switch ports and the Router-Port Group Management Protocol (RGMP) to constrain IP multicast traffic on routing device-only network segments.

The default behavior for a Layer 2 switch is to forward all multicast traffic to every port that belongs to the destination LAN on the switch. This behavior reduces the efficiency of the switch, whose purpose is to limit traffic to the ports that need to receive the data. This behavior requires a constraining mechanism to reduce unnecessary multicast traffic, which improves switch performance.

Prerequisites for Constraining IP Multicast in a Switched Ethernet Network

Before using the tasks in this module, you should be familiar with the concepts described in the “IP Multicast Technology Overview” module.

Information About IP Multicast in a Switched Ethernet Network

IP Multicast Traffic and Layer 2 Switches

The default behavior for a Layer 2 switch is to forward all multicast traffic to every port that belongs to the destination LAN on the switch. This behavior reduces the efficiency of the switch, whose purpose is to limit traffic to the ports that need to receive the data. This behavior requires a constraining mechanism to reduce unnecessary multicast traffic, which improves switch performance.

Cisco Group Management Protocol (CGMP), Router Group Management Protocol (RGMP), and IGMP snooping efficiently constrain IP multicast in a Layer 2 switching environment.

- CGMP and IGMP snooping are used on subnets that include end users or receiver clients.
- RGMP is used on routed segments that contain only routers, such as in a collapsed backbone.
- RGMP and CGMP cannot interoperate. However, Internet Group Management Protocol (IGMP) can interoperate with CGMP and RGMP snooping.

CGMP on Catalyst Switches for IP Multicast

CGMP is a Cisco-developed protocol used on device connected to Catalyst switches to perform tasks similar to those performed by IGMP. CGMP is necessary for those Catalyst switches that do not distinguish between IP multicast data packets and IGMP report messages, both of which are addressed to the same group address at the MAC level. The switch can distinguish IGMP packets, but would need to use software on the switch, greatly impacting its performance.

You must configure CGMP on the multicast device and the Layer 2 switches. The result is that, with CGMP, IP multicast traffic is delivered only to those Catalyst switch ports that are attached to interested receivers. All other ports that have not explicitly requested the traffic will not receive it unless these ports are connected to a multicast router. Multicast router ports must receive every IP multicast data packet.

Using CGMP, when a host joins a multicast group, it multicasts an unsolicited IGMP membership report message to the target group. The IGMP report is passed through the switch to the router for normal IGMP processing. The router (which must have CGMP enabled on this interface) receives the IGMP report and processes it as it normally would, but also creates a CGMP Join message and sends it to the switch. The Join message includes the MAC address of the end station and the MAC address of the group it has joined.

The switch receives this CGMP Join message and then adds the port to its content-addressable memory (CAM) table for that multicast group. All subsequent traffic directed to this multicast group is then forwarded out the port for that host.

The Layer 2 switches are designed so that several destination MAC addresses could be assigned to a single physical port. This design allows switches to be connected in a hierarchy and also allows many multicast destination addresses to be forwarded out a single port.

The device port also is added to the entry for the multicast group. Multicast device must listen to all multicast traffic for every group because IGMP control messages are also sent as multicast traffic. The rest of the multicast traffic is forwarded using the CAM table with the new entries created by CGMP.

IGMP Snooping

IGMP snooping is an IP multicast constraining mechanism that runs on a Layer 2 LAN switch. IGMP snooping requires the LAN switch to examine, or “snoop,” some Layer 3 information (IGMP Join/Leave messages) in the IGMP packets sent between the hosts and the router. When the switch receives the IGMP host report from a host for a particular multicast group, the switch adds the port number of the host to the associated multicast table entry. When the switch hears the IGMP Leave group message from a host, the switch removes the table entry of the host.

Because IGMP control messages are sent as multicast packets, they are indistinguishable from multicast data at Layer 2. A switch running IGMP snooping must examine every multicast data packet to determine if it contains any pertinent IGMP control information. IGMP snooping implemented on a low-end switch with a slow CPU could have a severe performance impact when data is sent at high rates. The solution is to implement

IGMP snooping on high-end switches with special application-specific integrated circuits (ASICs) that can perform the IGMP checks in hardware. CGMP is a better option for low-end switches without special hardware.

Router-Port Group Management Protocol (RGMP)

CGMP and IGMP snooping are IP multicast constraining mechanisms designed to work on routed network segments that have active receivers. They both depend on IGMP control messages that are sent between the hosts and the routers to determine which switch ports are connected to interested receivers.

Switched Ethernet backbone network segments typically consist of several routers connected to a switch without any hosts on that segment. Because routers do not generate IGMP host reports, CGMP and IGMP snooping will not be able to constrain the multicast traffic, which will be flooded to every port on the VLAN. Routers instead generate Protocol Independent Multicast (PIM) messages to Join and Prune multicast traffic flows at a Layer 3 level.

Router-Port Group Management Protocol (RGMP) is an IP multicast constraining mechanism for router-only network segments. RGMP must be enabled on the routers and on the Layer 2 switches. A multicast router indicates that it is interested in receiving a data flow by sending an RGMP Join message for a particular group. The switch then adds the appropriate port to its forwarding table for that multicast group--similar to the way it handles a CGMP Join message. IP multicast data flows will be forwarded only to the interested router ports. When the router no longer is interested in that data flow, it sends an RGMP Leave message and the switch removes the forwarding entry.

If there are any routers that are not RGMP-enabled, they will continue to receive all multicast data.

How to Constrain Multicast in a Switched Ethernet Network

Configuring Switches for IP Multicast

If you have switching in your multicast network, consult the documentation for the switch you are working with for information about how to configure IP multicast.

Configuring IGMP Snooping

No configuration is required on the router. Consult the documentation for the switch you are working with to determine how to enable IGMP snooping and follow the provided instructions.

Enabling CGMP

CGMP is a protocol used on devices connected to Catalyst switches to perform tasks similar to those performed by IGMP. CGMP is necessary because the Catalyst switch cannot distinguish between IP multicast data packets and IGMP report messages, which are both at the MAC level and are addressed to the same group address.

Note

- CGMP should be enabled only on 802 or ATM media, or LAN emulation (LANE) over ATM.
- CGMP should be enabled only on devices connected to Catalyst switches.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip cgmp** [**proxy** | **router-only**]
5. **end**
6. **clear ip cgmp** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.
Step 4	**ip cgmp** [**proxy** | **router-only**] **Example:** `Device(config-if)# ip cgmp proxy`	Enables CGMP on an interface of a device connected to a Cisco Catalyst 5000 family switch. • The **proxy** keyword enables the CGMP proxy function. When enabled, any device that is not CGMP-capable will be advertised by the proxy router. The proxy router advertises the existence of other non-CGMP-capable devices by sending a CGMP Join message with the MAC address of the non-CGMP-capable device and group address of 0000.0000.0000.
Step 5	**end** **Example:** `Device(config-if)# end`	Ends the current configuration session and returns to EXEC mode.
Step 6	**clear ip cgmp** [*interface-type interface-number*] **Example:** `Device# clear ip cgmp`	(Optional) Clears all group entries from the caches of Catalyst switches.

Configuring IP Multicast in a Layer 2 Switched Ethernet Network

Perform this task to configure IP multicast in a Layer 2 Switched Ethernet network using RGMP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rgmp**
5. **end**
6. **debug ip rgmp**
7. **show ip igmp interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Selects an interface that is connected to hosts.
Step 4	**ip rgmp** **Example:** `Device(config-if)# ip rgmp`	Enables RGMP on Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces.
Step 5	**end** **Example:** `Device(config-if)# end`	Ends the current configuration session and returns to EXEC mode.
Step 6	**debug ip rgmp** **Example:** `Device# debug ip rgmp`	(Optional) Logs debug messages sent by an RGMP-enabled device.

	Command or Action	Purpose
Step 7	**show ip igmp interface** **Example:** `Device# show ip igmp interface`	(Optional) Displays multicast-related information about an interface.

Configuration Examples for Constraining IP Multicast in a Switched Ethernet Network

Example: CGMP Configuration

The following example is for a basic network environment where multicast source(s) and multicast receivers are in the same VLAN. The desired behavior is that the switch will constrain the multicast forwarding to those ports that request the multicast stream.

A 4908G-L3 router is connected to the Catalyst 4003 on port 3/1 in VLAN 50. The following configuration is applied on the GigabitEthernet1 interface. Note that there is no **ip multicast-routing** command configured because the router is not routing multicast traffic across its interfaces.

```
interface GigabitEthernet1
 ip address 192.168.50.11 255.255.255.0
 ip pim dense-mode
 ip cgmp
```

RGMP Configuration Example

The following example shows how to configure RGMP on a router:

```
ip multicast-routing
ip pim sparse-mode
interface ethernet 0
 ip rgmp
```

Additional References

The following sections provide references related to constraining IP multicast in a switched Ethernet network.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP Multicast Command Reference

Related Topic	Document Title
IGMP snooping	The "IGMP Snooping" module of the *IP Multicast: IGMP Configuration Guide*
RGMP	The "Configuring Router-Port Group Management Protocol" module of the *IP Multicast: IGMP Configuration Guide*

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Feature Information for Constraining IP Multicast in a Switched Ethernet Network

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 33: Feature Information for Constraining IP Multicast in a Switched Ethernet Network

Feature Name	Releases	Feature Configuration Information
Cisco IOS	--	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.

CHAPTER 43

Configuring Router-Port Group Management Protocol

Router-Port Group Management Protocol (RGMP) is 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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Prerequisites for RGMP

Before you enable RGMP, ensure that the following features are enabled on your router:

- IP routing
- IP multicast
- PIM in sparse mode, sparse-dense mode, source specific mode, or bidirectional mode

If your router is in a bidirectional group, make sure to enable RGMP only on interfaces that do not function as a designated forwarder (DF). If you enable RGMP on an interface that functions as a DF, the interface will not forward multicast packets up the bidirectional shared tree to the rendezvous point (RP).

You must have the following features enabled on your switch:

- IP multicast
- IGMP snooping

Note Refer to the Catalyst switch software documentation for RGMP switch configuration tasks and command information.

Information About RGMP

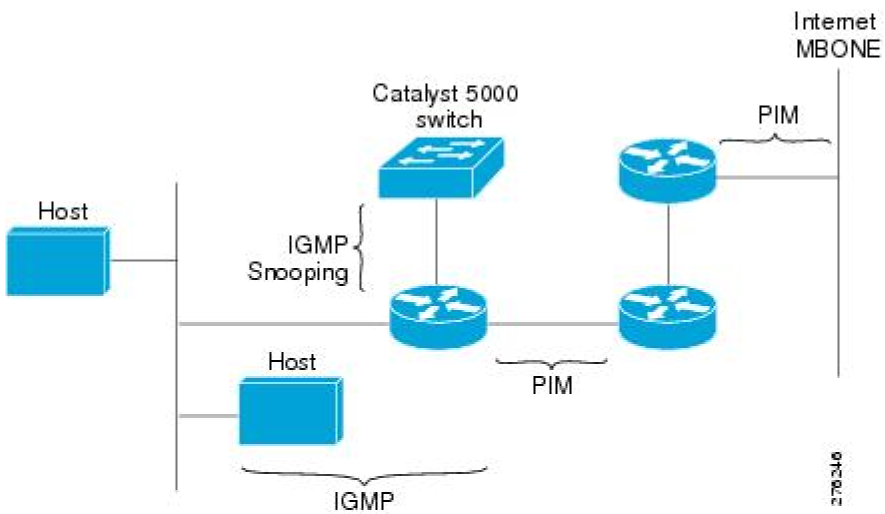
IP Multicast Routing Overview

The software supports the following protocols to implement IP multicast routing:

- Internet Group Management Protocol (IGMP) is used between hosts on a LAN and the routers on that LAN to track the multicast groups of which hosts are members.
- Protocol Independent Multicast (PIM) is used between routers so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- Cisco Group Management Protocol (CGMP) is a protocol used on routers connected to Catalyst switches to perform tasks similar to those performed by IGMP.
- RGMP is a protocol used on routers connected to Catalyst switches or networking devices functioning as Layer 2 switches to restrict IP multicast traffic. Specifically, the protocol enables a router to communicate to a switch the IP multicast group for which the router would like to receive or forward traffic.

The figure shows where these protocols operate within the IP multicast environment.

Figure 59: IP Multicast Routing Protocols



> **Note** CGMP and RGMP cannot interoperate on the same switched network. If RGMP is enabled on a switch or router interface, CGMP is automatically disabled on that switch or router interface; if CGMP is enabled on a switch or router interface, RGMP is automatically disabled on that switch or router interface.

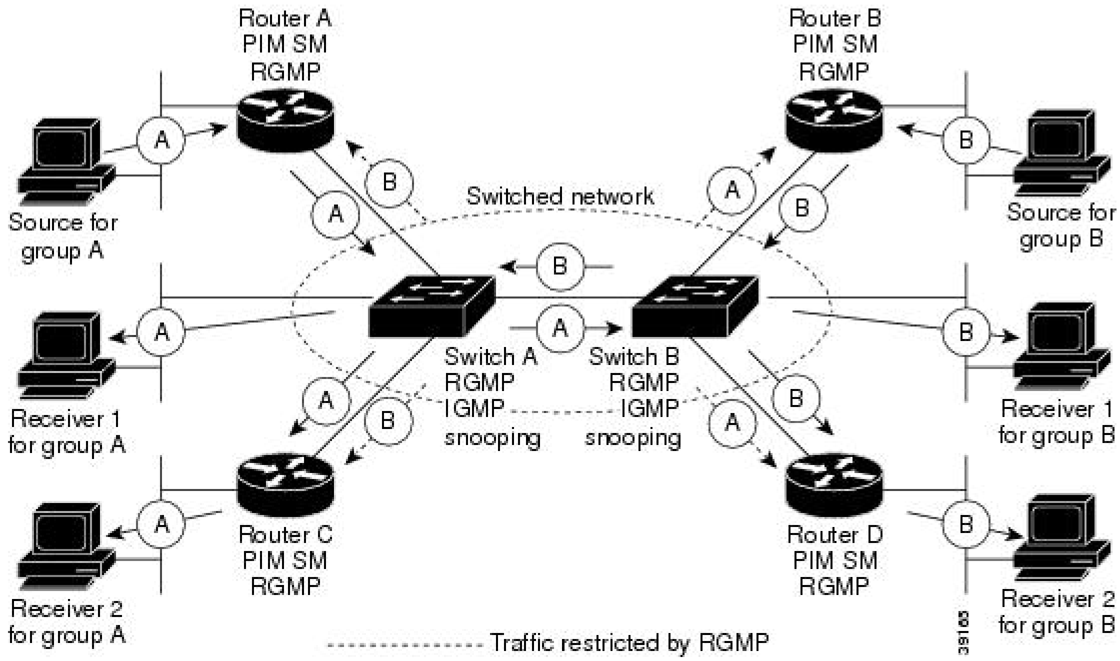
RGMP Overview

RGMP enables a router to communicate to a switch the IP multicast group for which the router would like to receive or forward traffic. RGMP is designed for switched Ethernet backbone networks running PIM sparse mode (PIM-SM) or sparse-dense mode.

> **Note** RGMP-enabled switches and router interfaces in a switched network support directly connected, multicast-enabled hosts that receive multicast traffic. RGMP-enabled switches and router interfaces in a switched network do not support directly connected, multicast-enabled hosts that source multicast traffic. A multicast-enabled host can be a PC, a workstation, or a multicast application running in a router.

The figure shows a switched Ethernet backbone network running PIM in sparse mode, RGMP, and IGMP snooping.

Figure 60: RGMP in a Switched Network

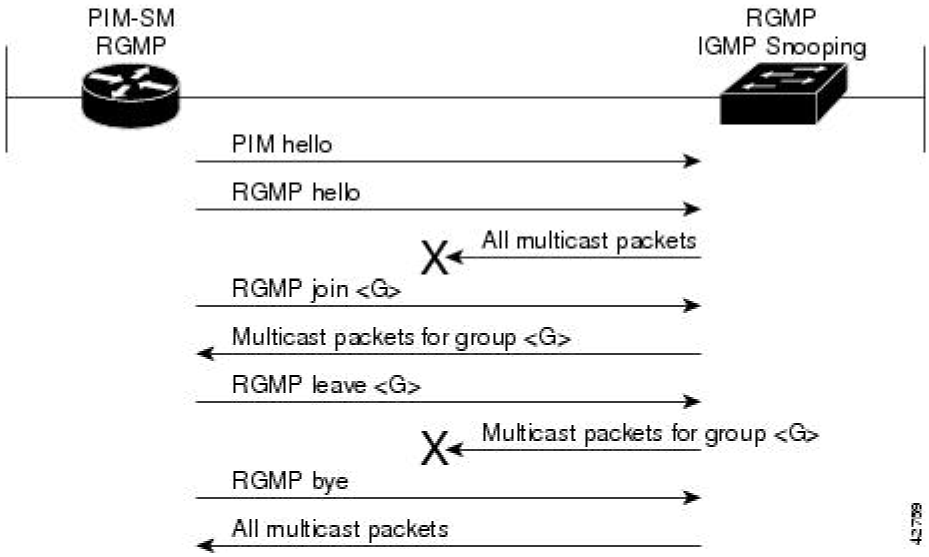


In the figure, the sources for the two different multicast groups (the source for group A and the source for group B) send traffic into the same switched network. Without RGMP, traffic from source A is unnecessarily flooded from switch A to switch B, then to router B and router D. Also, traffic from source B is unnecessarily flooded from switch B to switch A, then to router A and router C. With RGMP enabled on all routers and switches in this network, traffic from source A would not flood router B and router D. Also, traffic from source B would not flood router A and router C. Traffic from both sources would still flood the link between switch A and switch B. Flooding over this link would still occur because RGMP does not restrict traffic on links toward other RGMP-enabled switches with routers behind them.

By restricting unwanted multicast traffic in a switched network, RGMP increases the available bandwidth for all other multicast traffic in the network and saves the processing resources of the routers.

The figure shows the RGMP messages sent between an RGMP-enabled router and an RGMP-enabled switch.

Figure 61: RGMP Messages



The router sends simultaneous PIM hello (or a PIM query message if PIM Version 1 is configured) and RGMP hello messages to the switch. The PIM hello message is used to locate neighboring PIM routers. The RGMP hello message instructs the switch to restrict all multicast traffic on the interface from which the switch received the RGMP hello message.

Note RGMP messages are sent to the multicast address 224.0.0.25, which is the local-link multicast address reserved by the Internet Assigned Numbers Authority (IANA) for sending IP multicast traffic from routers to switches. If RGMP is not enabled on both the router and the switch, the switch automatically forwards all multicast traffic out the interface from which the switch received the PIM hello message.

The router sends the switch an RGMP join <G> message (where G is the multicast group address) when the router wants to receive traffic for a specific multicast group. The RGMP join message instructs the switch to forward multicast traffic for group <G> out the interface from which the switch received the RGMP hello message.

Note The router sends the switch an RGMP join <G> message for a multicast group even if the router is only forwarding traffic for the multicast group into a switched network. By joining a specific multicast group, the router can determine if another router is also forwarding traffic for the multicast group into the same switched network. If two routers are forwarding traffic for a specific multicast group into the same switched network, the two routers use the PIM assert mechanism to determine which router should continue forwarding the multicast traffic into the network.

The router sends the switch an RGMP leave <G> message when the router wants to stop receiving traffic for a specific multicast group. The RGMP leave message instructs the switch to stop forwarding the multicast traffic on the port from which the switch received the PIM and RGMP hello messages.

Note An RGMP-enabled router cannot send an RGMP leave <G> message until the router does not receive or forward traffic from any source for a specific multicast group (if multiple sources exist for a specific multicast group).

The router sends the switch an RGMP bye message when RGMP is disabled on the router. The RGMP bye message instructs the switch to forward the router all IP multicast traffic on the port from which the switch received the PIM and RGMP hello messages, as long as the switch continues to receive PIM hello messages on the port.

How to Configure RGMP

Enabling RGMP

To enable RGMP, use the following commands on all routers in your network beginning in global configuration mode:

Note CGMP and RGMP cannot interoperate on the same switched network. If RGMP is enabled on a switch or router interface, CGMP is automatically disabled on that switch or router interface; if CGMP is enabled on a switch or router interface, RGMP is automatically disabled on that switch or router interface.

SUMMARY STEPS

1. **interface** *type number*
2. **ip rgmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *type number*	Specifies the router interface on which you want to configure RGMP and enters interface configuration mode.
Step 2	**ip rgmp**	Enables RGMP on a specified interface.

What to do next

See the "RGMP_Configuration_Example" section for an example of how to configure RGMP.

Verifying RGMP Configuration

To verify that RGMP is enabled on the correct interfaces, use the **show ip igmp interface**command:

```
Router> show ip igmp interface
gigabitethernet1/0 is up, line protocol is up
```

```
Internet address is 10.0.0.0/24
  IGMP is enabled on interface
Current IGMP version is 2
  RGMP is enabled
IGMP query interval is 60 seconds
IGMP querier timeout is 120 seconds
IGMP max query response time is 10 seconds
Last member query response interval is 1000 ms
Inbound IGMP access group is not set
IGMP activity: 1 joins, 0 leaves
Multicast routing is enabled on interface
Multicast TTL threshold is 0
Multicast designated router (DR) is 10.0.0.0 (this system)
IGMP querying router is 10.0.0.0 (this system)
Multicast groups joined (number of users):
    224.0.1.40(1)
```

Note If RGMP is not enabled on an interface, no RGMP information is displayed in the **show ip igmp interface** command output for that interface.

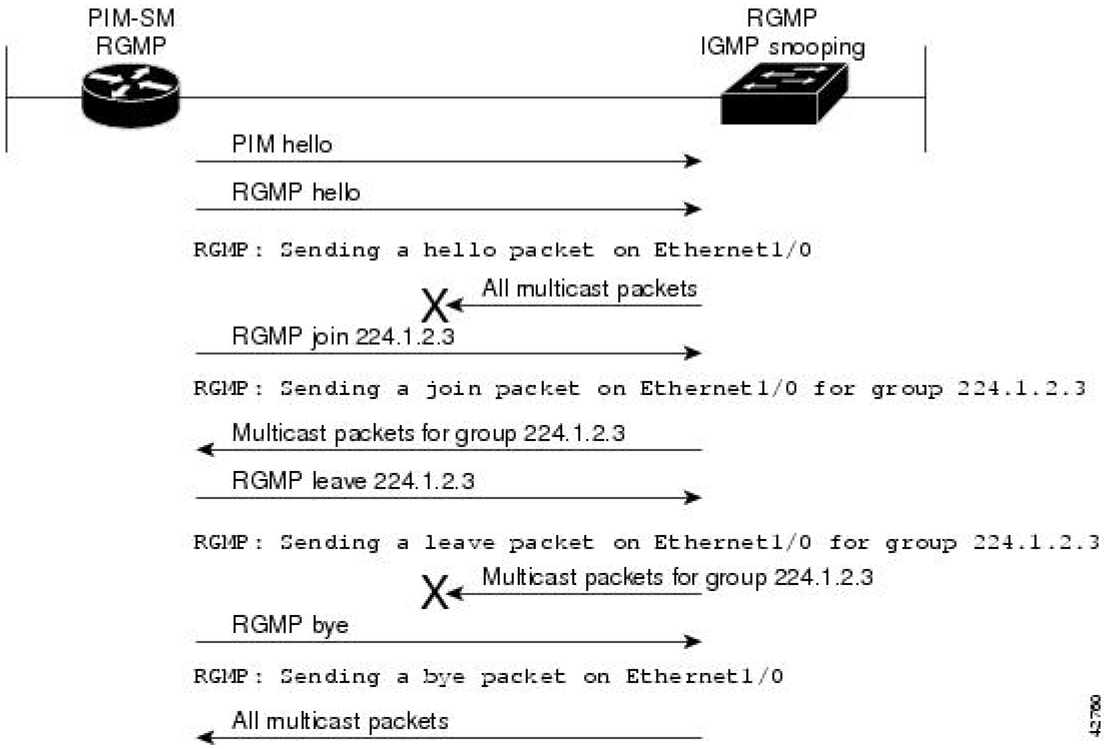
Monitoring and Maintaining RGMP

To enable RGMP debugging, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug ip rgmp`	Logs debug messages sent by an RGMP-enabled router. Using the command without arguments logs RGMP Join <G> and RGMP leave <G> messages for all multicast groups configured on the router. Using the command with arguments logs RGMP join <G> and RGMP leave <G> messages for the specified group.

The figure shows the debug messages that are logged by an RGMP-enabled router as the router sends RGMP join <G> and RGMP leave <G> messages to an RGMP-enabled switch.

Figure 62: RGMP Debug Messages

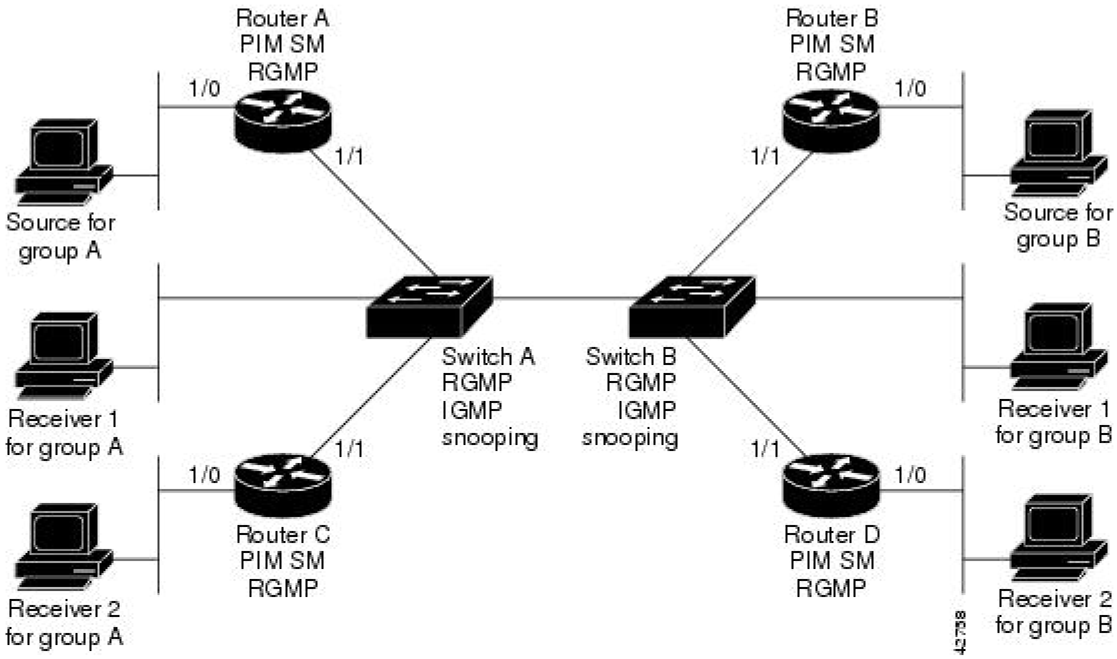


Configuration Examples for RGMP

RGMP Configuration Example

This section provides an RGMP configuration example that shows the individual configurations for the routers and switches shown in the figure.

Figure 63: RGMP Configuration Example



Router A Configuration

```
ip routing
ip multicast-routing distributed
interface gigabitethernet 1/0/0
  ip address 10.0.0.1 255.0.0.0
  ip pim sparse-dense-mode
  no shutdown
interface gigabitethernet 1/1/0
  ip address 10.1.0.1 255.0.0.0
  ip pim sparse-dense-mode
  ip rgmp
  no shutdown
```

Router B Configuration

```
ip routing
ip multicast-routing distributed
interface gigabitethernet 1/0/0
  ip address 10.2.0.1 255.0.0.0
  ip pim sparse-dense-mode
  no shutdown
interface gigabitethernet 1/1/0
  ip address 10.3.0.1 255.0.0.0
  ip pim sparse-dense-mode
  ip rgmp
  no shutdown
```

Router C Configuration

```
ip routing
```

```
ip multicast-routing distributed
interface gigabitethernet 1/0/0
  ip address 10.4.0.1 255.0.0.0
  ip pim sparse-dense-mode
  no shutdown
interface gigabitethernet 1/1/0
  ip address 10.5.0.1 255.0.0.0
  ip pim sparse-dense-mode
  ip rgmp
  no shutdown
```

Router D Configuration

```
ip routing
ip multicast-routing distributed
interface gigabitethernet 1/0/0
  ip address 10.6.0.1 255.0.0.0
  ip pim sparse-dense-mode
  no shutdown
interface gigabitethernet 1/1/0
  ip address 10.7.0.1 255.0.0.0
  ip pim sparse-dense-mode
  ip rgmp
  no shutdown
```

Switch A Configuration

```
Switch> (enable) set igmp enable
Switch> (enable) set rgmp enable
```

Switch B Configuration

```
Switch> (enable) set igmp enable
Switch> (enable) set rgmp enable
```

Additional References

The following sections provide references related to RGMP.

Related Documents

Related Topic	Document Title
PIM-SM and SSM concepts and configuration examples	“ Configuring Basic IP Multicast ” module
IP multicast commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Router-Port Group Management Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 34: Feature Information for Router-Port Group Management Protocol

Feature Name	Releases	Feature Information
Router-Port Group Management Protocol	Cisco IOS XE Release 2.1	Router-Port Group Management Protocol (RGMP) is 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

CHAPTER 44

Configuring IP Multicast over Unidirectional Links

IP multicast requires bidirectional communication, yet some networks include broadcast satellite links, which are unidirectional. Unidirectional link routing (UDLR) provides three mechanisms for a router to emulate a bidirectional link to enable the routing of unicast and multicast packets over a physical unidirectional interface, such as a broadcast satellite link. The mechanisms are a UDLR tunnel, Internet Group Management Protocol (IGMP) UDLR, and IGMP proxy. This document describes a UDLR tunnel and IGMP UDLR. IGMP proxy is described in the " Customizing IGMP " module. The three mechanisms may be used independently or in combination.

Prerequisites for UDLR

- You understand the concepts in the " IP Multicast Technology Overview " module.
- You have IP multicast configured in your network. Refer to the " Configuring Basic IP Multicast " module.

Information About UDLR

UDLR Overview

Both unicast and multicast routing protocols forward data on interfaces from which they have received routing control information. This model requires a bidirectional link. However, some network links are unidirectional. For networks that are unidirectional (such as broadcast satellite links), a method of communication that allows for control information to operate in a unidirectional environment is necessary. (Note that IGMP is not a routing protocol.)

Specifically, in unicast routing, when a router receives an update message on an interface for a prefix, it forwards data for destinations that match that prefix out that same interface. This is the case in distance vector routing protocols. Similarly, in multicast routing, when a router receives a Join message for a multicast group on an interface, it forwards copies of data destined for that group out that same interface. Based on these principles, unicast and multicast routing protocols cannot be supported over UDLs without the use of UDLR. UDLR is designed to enable the operation of routing protocols over UDLs without changing the routing protocols themselves.

UDLR enables a router to emulate the behavior of a bidirectional link for IP operations over UDLs. UDLR has three complementary mechanisms for bidirectional link emulation, which are described in the following sections:

- UDLR Tunnel--A mechanism for routing unicast and multicast traffic.
- Internet Group Management Protocol (IGMP) UDLR--Mechanism for routing multicast traffic. This method scales well for many broadcast satellite links.
- IGMP Proxy--Mechanism for routing multicast traffic.

You can use each mechanism independently or in conjunction with the others. IGMP proxy is described in the " Customizing IGMP " module.

UDLR Tunnel

The UDLR tunnel mechanism enables IP and its associated unicast and multicast routing protocols to treat the unidirectional link (UDL) as being logically bidirectional. A packet that is destined on a receive-only interface is picked up by the UDLR tunnel mechanism and sent to an upstream router using a generic routing encapsulation (GRE) tunnel. The control traffic flows in the opposite direction of the user data flow. When the upstream router receives this packet, the UDLR tunnel mechanism makes it appear that the packet was received on a send-only interface on the UDL.

The purpose of the unidirectional GRE tunnel is to move control packets from a downstream node to an upstream node. The one-way tunnel is mapped to a one-way interface (that goes in the opposite direction). Mapping is performed at the link layer, so the one-way interface appears bidirectional. When the upstream node receives packets over the tunnel, it must make the upper-layer protocols act as if the packets were received on the send-capable UDL.

A UDLR tunnel supports the following functionality:

- Address Resolution Protocol (ARP) and Next Hop Resolution Protocol (NHRP) over a UDL
- Emulation of bidirectional links for all IP traffic (as opposed to only control-only broadcast/multicast traffic)
- Support for IP GRE multipoint at a receive-only tunnel

Note A UDL router can have many routing peers (for example, routers interconnected via a broadcast satellite link). As with bidirectional links, the number of peer routers a router has must be kept relatively small to limit the volume of routing updates that must be processed. For multicast operation, we recommend using the IGMP UDLR mechanism when interconnecting more than 20 routers.

IGMP UDLR

In addition to a UDLR tunnel, another mechanism that enables support of multicast routing protocols over UDLs is using IP multicast routing with IGMP, which accommodates UDLR. This mechanism scales well for many broadcast satellite links.

With IGMP UDLR, an upstream router sends periodic queries for members on the UDL. The queries include a unicast address of the router that is not the unicast address of the unidirectional interface. The downstream routers forward IGMP reports received from directly connected members (on interfaces configured to helper forward IGMP reports) to the upstream router. The upstream router adds the unidirectional interface to the (*, G) outgoing interface list, thereby enabling multicast packets to be forwarded down the UDL.

In a large enterprise network, it is not possible to be able to receive IP multicast traffic via satellite and forward the traffic throughout the network. This limitation exists because receiving hosts must be directly connected to the downstream router. However, you can use the IGMP proxy mechanism to overcome this limitation. Refer to the " Customizing IGMP " module for more information on this mechanism.

How to Route IP Multicast over Unidirectional Links

This section includes the following procedures. You can do either or both in your network.

Configuring a UDLR Tunnel

To configure a UDLR tunnel, perform the task in this section. The tunnel mode defaults to GRE. You need not assign an IP address to the tunnel (you need not use the **ip address** or **ip unnumbered** commands). You must configure the tunnel endpoint addresses.

You must configure both the upstream and downstream routers to meet the following conditions:

- On the upstream router, where the UDL can only send, you must configure the tunnel to receive. When packets are received over the tunnel, the upper-layer protocols treat the packet as though it is received over the unidirectional, send-only interface.
- On the downstream router, where the UDL can only receive, you must configure the tunnel to send. When packets are sent by upper-layer protocols over the interface, they will be redirected and sent over this GRE tunnel.

Before you begin

Before configuring UDLR tunnel, ensure that all routers on the UDL have the same subnet address. If all routers on the UDL cannot have the same subnet address, the upstream router must be configured with secondary addresses to match all the subnets that the downstream routers are attached to.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **interface tunnel** *number*
5. **tunnel udlr receive-only** *type number*
6. **tunnel source** {ip-address | *type number*}

7. **tunnel destination** {*hostname*| ip-address}
8. Move to the downstream router.
9. **enable**
10. **configure terminal**
11. **interface** *type number*
12. **interface tunnel** *number*
13. **tunnel udlr send-only** *type number*
14. **tunnel source** {ip-address | *type number*}
15. **tunnel destination** {*hostname*| ip-address}
16. **tunnel udlr address-resolution**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted. • Do this step on the upstream router.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures the unidirectional send-only interface.
Step 4	**interface tunnel** *number* **Example:** `Router(config-if)# interface tunnel 0`	Configures the receive-only tunnel interface.
Step 5	**tunnel udlr receive-only** *type number* **Example:** `Router(config-if)# tunnel udlr receive-only fastethernet 0/0/0`	Configures the UDLR tunnel. • Use the same *type* and *number* values as the unidirectional send-only interface *type* and *number* values specified with the **interface** *type number* command in Step 3.
Step 6	**tunnel source** {ip-address | *type number*} **Example:** `Router(config-if)# tunnel source 10.3.4.5`	Configures the tunnel source.
Step 7	**tunnel destination** {*hostname*| ip-address} **Example:**	Configures the tunnel destination.

	Command or Action	Purpose
	`Router(config-if)# tunnel destination 10.8.2.3`	
Step 8	Move to the downstream router.	--
Step 9	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 10	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 11	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures the unidirectional receive-only interface.
Step 12	**interface tunnel** *number* **Example:** `Router(config-if)# interface tunnel 0`	Configures the send-only tunnel interface.
Step 13	**tunnel udlr send-only** *type number* **Example:** `Router(config-if)# tunnel udlr send-only ethernet 0`	Configures the UDLR tunnel. • Use the same *type* and *number* values as the unidirectional receive-only interface *type* and *number* values specified with the **interface** *type number* command in Step 3.
Step 14	**tunnel source** {ip-address | *type number*} **Example:** `Router(config-if)# tunnel source 11.8.2.3`	Configures the tunnel source.
Step 15	**tunnel destination** {*hostname*| ip-address} **Example:** `Router(config-if)# tunnel destination 10.3.4.5`	Configures the tunnel destination.
Step 16	**tunnel udlr address-resolution** **Example:** `Router(config-if)# tunnel udlr address-resolution`	Enables the forwarding of ARP and NHRP.

Configuring IGMP UDLR

To configure an IGMP UDL, you must configure both the upstream and downstream routers. You need not specify whether the direction is sending or receiving; IGMP learns the direction by the nature of the physical connection.

When the downstream router receives an IGMP report from a host, the router sends the report to the IGMP querier associated with the UDL interface identified in the **ip igmp helper-address** command.

Before you begin

- All routers on the UDL have the same subnet address. If all routers on the UDL cannot have the same subnet address, the upstream router must be configured with secondary addresses to match all the subnets that the downstream routers are attached to.
- Multicast receivers are directly connected to the downstream routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp unidirectional-link**
5. Move to the downstream router.
6. **enable**
7. **configure terminal**
8. **ip multicast default-rpf-distance** *distance*
9. **interface** *type number*
10. **ip igmp unidirectional-link**
11. **ip igmp helper-address udl** *type number*
12. **exit**
13. **show ip igmp udlr** [*group-name*| *group-address* | *type number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted. • Begin on the upstream router.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Configures the interface.

	Command or Action	Purpose
	`Router(config)# interface gigabitethernet 0/1/1`	
Step 4	**ip igmp unidirectional-link** **Example:** `Router(config-if)# ip igmp unidirectional-link`	Configures IGMP on the interface to be unidirectional.
Step 5	Move to the downstream router.	--
Step 6	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted. • Begin on the upstream router.
Step 7	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 8	**ip multicast default-rpf-distance** *distance* **Example:** `Router# ip multicast default-rpf-distance 10`	(Optional) Sets the distance for the default RPF interface. By default, the distance for the default reverse path forwarding (RPF) interface is 15. Any explicit sources learned by routing protocols will take preference if their distance is less than the distance configured by the **ip multicast default-rpf-distance** command. Use this command on downstream routers if you want some sources to use RPF to reach the UDLR link and others to use the terrestrial paths. • If you want IGMP to prefer the UDL, set the distance to be less than the distances of the unicast routing protocols. • If you want IGMP to prefer the non-UDL, set the distance to be greater than the distances of the unicast routing protocols.
Step 9	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures the interface.
Step 10	**ip igmp unidirectional-link** **Example:** `Router(config-if)# ip igmp unidirectional-link`	Configures IGMP on the interface to be unidirectional.
Step 11	**ip igmp helper-address udl** *type number*	Configures the interface to be an IGMP helper.

	Command or Action	Purpose
	Example: `Router(config-if)# ip igmp helper-address udl ethernet 0`	• Use this command on every downstream router, on every interface to specify the *type* and *number* values that identify the UDL interface.
Step 12	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode and returns to EXEC mode.
Step 13	**show ip igmp udlr** [*group-name*| *group-address* | *type number*] **Example:** `Router(config)# show ip igmp udlr`	(Optional) Displays UDLR information for directly connected multicast groups on interfaces that have a UDL helper address configured.

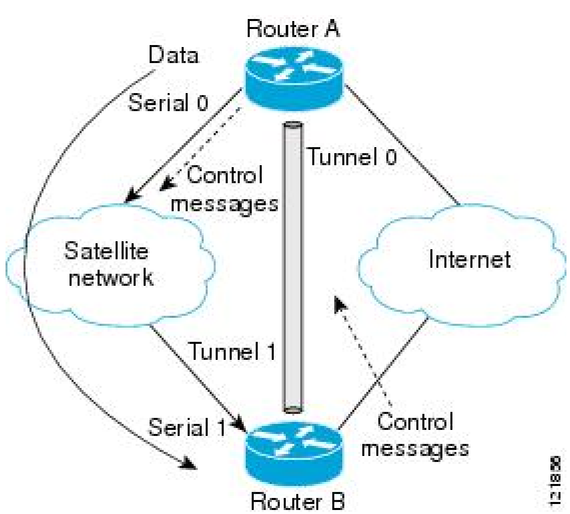
Configuration Examples for UDLR

UDLR Tunnel Example

The following example shows how to configure a UDLR tunnel. In the example, Router A (the upstream router) is configured with Open Shortest Path First (OSPF) and PIM. Serial interface 0 has send-only capability. Therefore, the UDLR tunnel is configured as receive only, and points to serial 0.

Router B (the downstream router) is configured with OSPF and PIM. Serial interface 1 has receive-only capability. Therefore, the UDLR tunnel is configured as send-only, and points to serial 1. The forwarding of ARP and NHRP is enabled. The figure below illustrates the example.

Figure 64: UDLR Tunnel Example



Router A Configuration

```
ip multicast-routing
!
! Serial0/0/0 has send-only capability
!
interface serial 0/0/0
 encapsulation hdlc
 ip address 10.1.0.1 255.255.0.0
 ip pim sparse-dense-mode
!
! Configure tunnel as receive-only UDLR tunnel.
!
interface tunnel 0
 tunnel source 10.20.0.1
 tunnel destination 10.41.0.2
 tunnel udlr receive-only serial 0/0/0
!
! Configure OSPF.
!
router ospf
 network 10.0.0.0 0.255.255.255 area 0
```

Router B Configuration

```
ip multicast-routing
!
! Serial1 has receive-only capability
!
interface serial 1/0/0
 encapsulation hdlc
 ip address 10.1.0.2 255.255.0.0
 ip pim sparse-dense-mode
!
! Configure tunnel as send-only UDLR tunnel.
!
interface tunnel 0
 tunnel source 10.41.0.2
 tunnel destination 10.20.0.1
 tunnel udlr send-only serial 1/0/0
 tunnel udlr address-resolution
!
! Configure OSPF.
!
router ospf
 network 10.0.0.0 0.255.255.255 area 0
```

IGMP UDLR Example

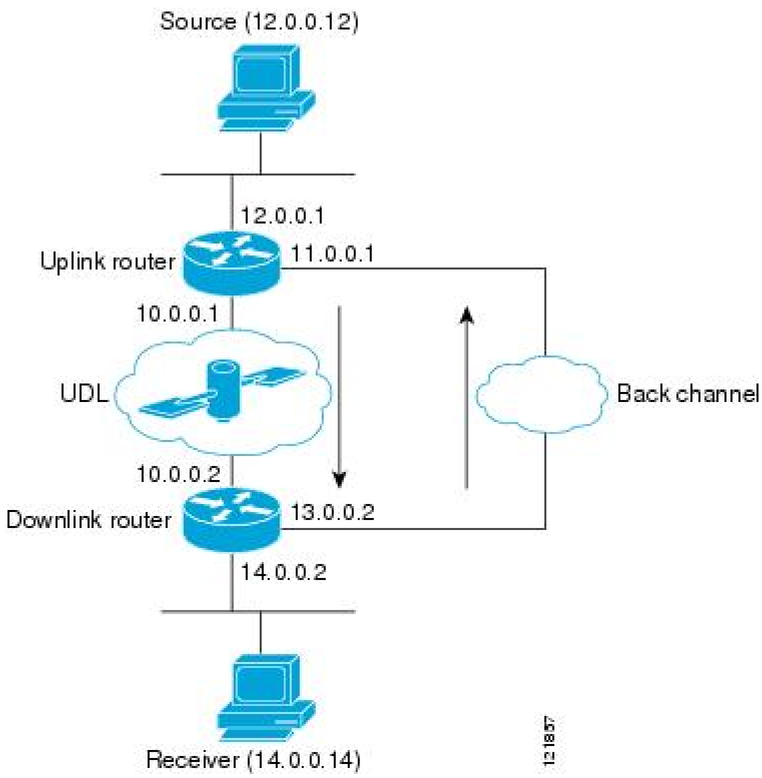
The following example shows how to configure IGMP UDLR. In this example, uplink-rtr is the local upstream router and downlink-rtr is the downstream router.

Both routers are also connected to each other by a back channel connection. Both routers have two IP addresses: one on the UDL and one on the interface that leads to the back channel. The back channel is any return route and can have any number of routers.

> **Note**
> Configuring PIM on the back channel interfaces on the uplink router and downlink router is optional.

All routers on a UDL must have the same subnet address. If all routers on a UDL cannot have the same subnet address, the upstream router must be configured with secondary addresses to match all the subnets that the downstream routers are attached to.

Figure 65: IGMP Unidirectional Link Routing Example



Uplink Router (uplink-rtr) Configuration

```
ip multicast-routing
!
! Interface that source is attached to
!
interface gigabitethernet 0/0/0
 description Typical IP multicast enabled interface
 ip address 12.0.0.1 255.0.0.0
 ip pim sparse-dense-mode
!
! Back channel
!
interface gigabitethernet 1/0/0
 description Back channel which has connectivity to downlink-rtr
 ip address 11.0.0.1 255.0.0.0
 ip pim sparse-dense-mode
!
! Unidirectional link
!
```

```
interface serial 0/0/0
 description Unidirectional to downlink-rtr
 ip address 10.0.0.1 255.0.0.0
 ip pim sparse-dense-mode
 ip igmp unidirectional-link
 no keepalive
```

Downlink Router (downlink-rtr) Configuration

```
ip multicast-routing
!
! Interface that receiver is attached to, configure for IGMP reports to be
! helpered for the unidirectional interface.
!
interface gigabitethernet 0/0/0
 description Typical IP multicast-enabled interface
 ip address 14.0.0.2 255.0.0.0
 ip pim sparse-dense-mode
 ip igmp helper-address udl serial 0/0/0
!
! Back channel
!
interface gigabitethernet 1/0/0
 description Back channel that has connectivity to downlink-rtr
 ip address 13.0.0.2 255.0.0.0
 ip pim sparse-dense-mode
!
! Unidirectional link
!
interface serial 0/0/0
 description Unidirectional to uplink-rtr
 ip address 10.0.0.2 255.0.0.0
 ip pim sparse-dense-mode
 ip igmp unidirectional-link
 no keepalive
```

Integrated UDLR Tunnel IGMP UDLR and IGMP Proxy Example

The following example shows how to configure UDLR tunnels, IGMP UDLR, and IGMP proxy on both the upstream and downstream routers sharing a UDL.

Upstream Configuration

```
ip multicast-routing
!
interface Tunnel0
  ip address 9.1.89.97 255.255.255.252
  no ip directed-broadcast
  tunnel source 9.1.89.97
  tunnel mode gre multipoint
  tunnel key 5
  tunnel udlr receive-only GigabitEthernet2/3/0
!
interface GigabitEthernet2/0/0
  no ip address
  shutdown
!
! user network
interface GigabitEthernet2/1/0
```

```
  ip address 9.1.89.1 255.255.255.240
  no ip directed-broadcast
  ip pim dense-mode
  ip cgmp
  fair-queue 64 256 128
  no cdp enable
  ip rsvp bandwidth 1000 100
!
interface GigabitEthernet2/2/0
  ip address 9.1.95.1 255.255.255.240
  no ip directed-broadcast
!
! physical send-only interface
interface GigabitEthernet2/3/0
  ip address 9.1.92.100 255.255.255.240
  no ip directed-broadcast
  ip pim dense-mode
  ip nhrp network-id 5
  ip nhrp server-only
  ip igmp unidirectional-link
  fair-queue 64 256 31
  ip rsvp bandwidth 1000 100
!
router ospf 1
  network 9.1.92.96 0.0.0.15 area 1
!
ip classless
ip route 9.1.90.0 255.255.255.0 9.1.92.99
```

Downstream Configuration

```
ip multicast-routing
!
interface Loopback0
  ip address 9.1.90.161 255.255.255.252
  ip pim sparse-mode
  ip igmp helper-address udl GigabitEthernet2/3/0
  ip igmp proxy-service
!
interface Tunnel0
  ip address 9.1.90.97 255.255.255.252
  ip access-group 120 out
  no ip directed-broadcast
  no ip mroute-cache
  tunnel source 9.1.90.97
  tunnel destination 9.1.89.97
  tunnel key 5
  tunnel udlr send-only GigabitEthernet2/3/0
  tunnel udlr address-resolution
!
interface GigabitEthernet2/0/0
  no ip address
  no ip directed-broadcast
  shutdown
  no cdp enable
!
! user network
interface GigabitEthernet2/1/0
  ip address 9.1.90.1 255.255.255.240
  no ip directed-broadcast
  ip pim sparse-mode
  ip igmp mroute-proxy Loopback0
  no cdp enable
```

```
!
! Backchannel
interface GigabitEthernet2/2/0
  ip address 9.1.95.3 255.255.255.240
  no ip directed-broadcast
  no cdp enable
!
! physical receive-only interface
interface GigabitEthernet2/3/0
  ip address 9.1.92.99 255.255.255.240
  no ip directed-broadcast
  ip pim sparse-mode
  ip igmp unidirectional-link
  no keepalive
  no cdp enable
!
router ospf 1
  network 9.1.90.0 0.0.0.255 area 1
  network 9.1.92.96 0.0.0.15 area 1
!
ip classless
ip route 0.0.0.0 0.0.0.0 9.1.95.1
! set rpf to be the physical receive-only interface
ip mroute 0.0.0.0 0.0.0.0 9.1.92.96
ip pim rp-address 9.1.90.1
!
! permit ospf, ping and rsvp, deny others
access-list 120 permit icmp any any
access-list 120 permit 46 any any
access-list 120 permit ospf any any
```

Additional References

Related Documents

Related Topic	Document Title
IP multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*
Tunnel interfaces	" Implementing Tunnels " module
IGMP and IGMP Proxy	" Customizing IGMP " module

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring IP Multicast over Unidirectional Links

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 35: Feature Information for Configuring IP Multicast over Unidirectional Links

Feature Name	Releases	Feature Configuration Information
UDLR Tunnel ARP and IGMP Proxy	12.2(8)T	This feature enables arp over a unidirectional link and overcomes the existing limitation of requiring downstream multicast receivers to be directly connected to the unidirectional link downstream router.
Uni-Directional Link Routing (UDLR)	12.2(2)T 12.2(17d)SXB1	Unidirectional link routing is used to allow routing protocols to function in environments where routers are connected through unidirectional links. Unidirectional link routing enables layer 3 connectivity by tunneling routing information to the router on the upstream side of a unidirectional link.

PART IV

Multicast Optimization

CHAPTER 45

Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

This module describes how to optimize Protocol Independent Multicast (PIM) sparse mode for a large deployment of IP multicast. You can set a limit on the rate of PIM register messages sent in order to limit the load on the designated router and RP, you can reduce the PIM router query message interval to achieve faster convergence, and you can delay or prevent the use of the shortest path tree.

Prerequisites for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

- You must have PIM sparse mode running in your network.
- If you plan to use a group list to control to which groups the shortest-path tree (SPT) threshold applies, you must have configured your access list before performing the task.

Information About Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

PIM Registering Process

IP multicast sources do not use a signaling mechanism to announce their presence. Sources just send their data into the attached network, as opposed to receivers that use Internet Group Management Protocol (IGMP) to announce their presence. If a source sends traffic to a multicast group configured in PIM sparse mode

(PIM-SM), the Designated Router (DR) leading toward the source must inform the rendezvous point (RP) about the presence of this source. If the RP has downstream receivers that want to receive the multicast traffic (natively) from this source and has not joined the shortest path leading toward the source, then the DR must send the traffic from the source to the RP. The PIM registering process, which is individually run for each (S, G) entry, accomplishes these tasks between the DR and RP.

The registering process begins when a DR creates a new (S, G) state. The DR encapsulates all the data packets that match the (S, G) state into PIM register messages and unicasts those register messages to the RP.

If an RP has downstream receivers that want to receive register messages from a new source, the RP can either continue to receive the register messages through the DR or join the shortest path leading toward the source. By default, the RP will join the shortest path, because delivery of native multicast traffic provides the highest throughput. Upon receipt of the first packet that arrives natively through the shortest path, the RP will send a register-stop message back to the DR. When the DR receives this register-stop message, it will stop sending register messages to the RP.

If an RP has no downstream receivers that want to receive register messages from a new source, the RP will not join the shortest path. Instead, the RP will immediately send a register-stop message back to the DR. When the DR receives this register-stop message, it will stop sending register messages to the RP.

Once a routing entry is established for a source, a periodic reregistering takes place between the DR and RP. One minute before the multicast routing table state times out, the DR will send one dataless register message to the RP each second that the source is active until the DR receives a register-stop message from the RP. This action restarts the timeout time of the multicast routing table entry, typically resulting in one reregistering exchange every 2 minutes. Reregistering is necessary to maintain state, to recover from lost state, and to keep track of sources on the RP. It will take place independently of the RP joining the shortest path.

PIM Version 1 Compatibility

If an RP is running PIM Version 1, it will not understand dataless register messages. In this case, the DR will not send dataless register messages to the RP. Instead, approximately every 3 minutes after receipt of a register-stop message from the RP, the DR encapsulates the incoming data packets from the source into register messages and sends them to the RP. The DR continues to send register messages until it receives another register-stop message from the RP. The same behavior occurs if the DR is running PIM Version 1.

When a DR running PIM Version 1 encapsulates data packets into register messages for a specific (S, G) entry, the entry is process-switched, not fast-switched or hardware-switched. On platforms that support these faster paths, the PIM registering process for an RP or DR running PIM Version 1 may lead to periodic out-of-order packet delivery. For this reason, we recommend upgrading your network from PIM Version 1 to PIM Version 2.

PIM Designated Router

Devices configured for IP multicast send PIM hello messages to determine which device will be the designated router (DR) for each LAN segment (subnet). The hello messages contain the device's IP address, and the device with the highest IP address becomes the DR.

The DR sends Internet Group Management Protocol (IGMP) host query messages to all hosts on the directly connected LAN. When operating in sparse mode, the DR sends source registration messages to the rendezvous point (RP).

By default, multicast devices send PIM router query messages every 30 seconds. By enabling a device to send PIM hello messages more often, the device can discover unresponsive neighbors more quickly. As a result, the device can implement failover or recovery procedures more efficiently. It is appropriate to make this change only on redundant devices on the edge of the network.

PIM Sparse-Mode Register Messages

Dataless register messages are sent at a rate of one message per second. Continuous high rates of register messages might occur if a DR is registering bursty sources (sources with high data rates) and if the RP is not running PIM Version 2.

By default, PIM sparse-mode register messages are sent without limiting their rate. Limiting the rate of register messages will limit the load on the DR and RP, at the expense of dropping those register messages that exceed the set limit. Receivers may experience data packet loss within the first second in which packets are sent from bursty sources.

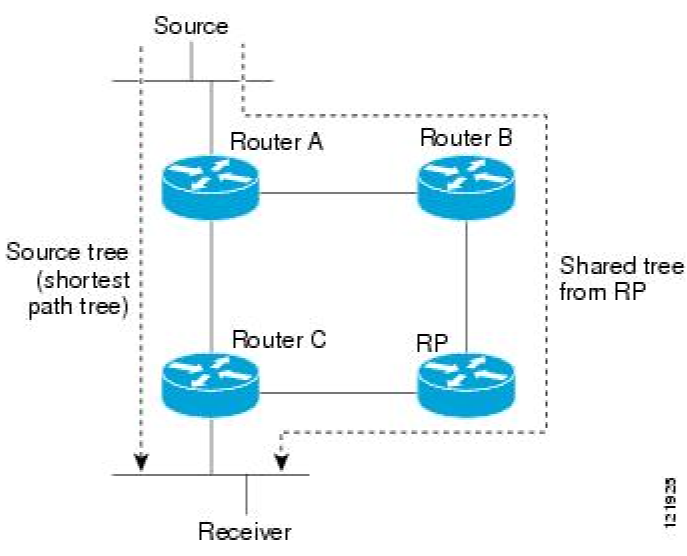
Preventing Use of Shortest-Path Tree to Reduce Memory Requirement

Understanding PIM shared tree and source tree will help you understand how preventing the use of the shortest-path tree can reduce memory requirements.

PIM Shared Tree and Source Tree - Shortest-Path Tree

By default, members of a multicast group receive data from senders to the group across a single data distribution tree rooted at the rendezvous point (RP). This type of distribution tree is called shared tree, as shown in the figure. Data from senders is delivered to the RP for distribution to group members joined to the shared tree.

Figure 66: Shared Tree versus Source Tree (Shortest-Path Tree)



If the data rate warrants, leaf routers on the shared tree may initiate a switch to the data distribution tree rooted at the source. This type of distribution tree is called a shortest-path tree (SPT) or source tree. By default, the software switches to a source tree upon receiving the first data packet from a source.

The following process describes the move from shared tree to source tree in more detail:

1. Receiver joins a group; leaf Router C sends a Join message toward the RP.

2. The RP puts the link to Router C in its outgoing interface list.
3. Source sends data; Router A encapsulates data in a register message and sends it to the RP.
4. The RP forwards data down the shared tree to Router C and sends a Join message toward the source. At this point, data may arrive twice at Router C, once encapsulated and once natively.
5. When data arrives natively (through multicast) at the RP, the RP sends a register-stop message to Router A.
6. By default, reception of the first data packet prompts Router C to send a Join message toward the source.
7. When Router C receives data on (S, G), it sends a Prune message for the source up the shared tree.
8. The RP deletes the link to Router C from the outgoing interface of (S, G). The RP triggers a Prune message toward the source.

Join and Prune messages are sent for sources and RPs. They are sent hop-by-hop and are processed by each PIM router along the path to the source or RP. Register and register-stop messages are not sent hop-by-hop. They are sent by the designated router that is directly connected to a source and are received by the RP for the group.

Multiple sources sending to groups use the shared tree.

Benefit of Preventing or Delaying the Use of the Shortest-Path Tree

The switch from shared to source tree happens upon the arrival of the first data packet at the last hop device (Router C in the figure *Shared Tree and Source Tree (Shortest-Path Tree)*). This switch occurs because the **ip pim spt-threshold** command controls that timing, and its default setting is 0 kbps.

The shortest-path tree requires more memory than the shared tree, but reduces delay. You might want to prevent or delay its use to reduce memory requirements. Instead of allowing the leaf device to move to the shortest-path tree immediately, you can prevent use of the SPT or specify that the traffic must first reach a threshold.

You can configure when a PIM leaf device should join the shortest-path tree for a specified group. If a source sends at a rate greater than or equal to the specified *kbps* rate, the device triggers a PIM Join message toward the source to construct a source tree (shortest-path tree). If the **infinity** keyword is specified, all sources for the specified group use the shared tree, never switching to the source tree.

How to Optimize PIM Sparse Mode in a Large IP Multicast Deployment

Optimizing PIM Sparse Mode in a Large Deployment

Consider performing this task if your deployment of IP multicast is large.

Steps 3, 5, and 6 in this task are independent of each other and are therefore considered optional. Any one of these steps will help optimize PIM sparse mode. If you are going to perform Step 5 or 6, you must perform Step 4. Step 6 applies only to a designated router; changing the PIM query interval is only appropriate on redundant routers on the edge of the PIM domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip pim register-rate-limit** *rate*
4. **ip pim spt-threshold** {*kbps*| **infinity**}[**group-list** *access-list*]
5. **interface** *type number*
6. **ip pim query-interval** *period* [**msec**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip pim register-rate-limit** *rate* **Example:** `Router(config)# ip pim register-rate-limit 10`	(Optional) Sets a limit on the maximum number of PIM sparse mode register messages sent per second for each (S, G) routing entry. • Use this command to limit the number of register messages that the designated router (DR) will allow for each (S, G) entry. • By default, there is no maximum rate set. • Configuring this command will limit the load on the DR and RP at the expense of dropping those register messages that exceed the set limit. • Receivers may experience data packet loss within the first second in which register messages are sent from bursty sources.
Step 4	**ip pim spt-threshold** {*kbps*| **infinity**}[**group-list** *access-list*] **Example:** `Router(config)# ip pim spt-threshold infinity group-list 5`	(Optional) Specifies the threshold that must be reached before moving to the shortest-path tree. • The default value is **0**, which causes the router to join the SPT immediately upon the first data packet it receives. • Specifying the **infinity** keyword causes the router never to move to the shortest-path tree; it remains on the shared tree. This keyword applies to a multicast environment of “many-to-many” communication.

	Command or Action	Purpose
		• The group list is a standard access list that controls which groups the SPT threshold applies to. If a value of 0 is specified or the group list is not used, the threshold applies to all groups. • In the example, group-list 5 is already configured to permit the multicast groups 239.254.2.0 and 239.254.3.0: access-list 5 permit 239.254.2.0 0.0.0.255 access-list 5 permit 239.254.3.0 0.0.0.255
Step 5	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Configures an interface. • If you do not want to change the default values of the PIM SPT threshold or the PIM query interval, do not perform this step; you are done with this task.
Step 6	**ip pim query-interval** *period* [**msec**] **Example:** `Router(config-if)# ip pim query-interval 1`	(Optional) Configures the frequency at which multicast routers send PIM router query messages. • Perform this step only on redundant routers on the edge of a PIM domain. • The default query interval is 30 seconds. • The *period* argument is in seconds unless the **msec** keyword is specified. • Set the query interval to a smaller number of seconds for faster convergence, but keep in mind the trade-off between faster convergence and higher CPU and bandwidth usage.

Configuration Examples for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

Optimizing PIM Sparse Mode in a Large IP Multicast Deployment Example

The following example shows how to:

- Set the query interval to 1 second for faster convergence.
- Configure the router to never move to the SPT but to remain on the shared tree.
- Set a limit of 10 PIM sparse mode register messages sent per second for each (S, G) routing entry.

```
interface ethernet 0
 ip pim query-interval 1
.
```

```
.
.
!
ip pim spt-threshold infinity
ip pim register-rate-limit 10
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP SLAs commands	Cisco IOS IP Multicast Command Reference
PIM Sparse Mode concepts and configuration	"Configuring Basic IP Multicast" module or "Configuring IP Multicast in IPv6 Networks" module

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by these features, and support for existing MIBs has not been modified by these features.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 36: Feature Information for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

Feature Name	Releases	Feature Configuration Information
PIM Version 2	12.2(4)T	Protocol Independent Multicast (PIM) version 2 builds upon the success of the existing PIMv1 base, has two basic operating modes: sparse-mode and dense-mode, and is suitable for large networks with heterogeneous links and devices.

CHAPTER 46

Multicast Subsecond Convergence

The Multicast Subsecond Convergence feature comprises a comprehensive set of features and protocol enhancements that provide for improved scalability and convergence in multicast-based services. This feature set provides for the ability to scale to larger services levels and to recover multicast forwarding after service failure in subsecond time frames.

Prerequisites for Multicast Subsecond Convergence

Service providers must have a multicast-enabled core in order to use the Cisco Multicast Subsecond Convergence feature.

Restrictions for Multicast Subsecond Convergence

Devices that use the subsecond designated router (DR) failover enhancement must be able to process hello interval information arriving in milliseconds. Devices that are congested or do not have enough CPU cycles to process the hello interval can assume that the Protocol Independent Multicast (PIM) neighbor is disconnected, although this may not be the case.

Information About Multicast Subsecond Convergence

Benefits of Multicast Subsecond Convergence

- The scalability components improve on the efficiency of handling increases (or decreases) in service users (receivers) and service load (sources or content).

- New algorithms and processes (such as aggregated join messages, which deliver up to 1000 individual messages in a single packet) reduce the time to reach convergence by a factor of 10.
- Multicast subsecond convergence improves service availability for large multicast networks.
- Multicast users such as financial services firms and brokerages receive better quality of service (QoS), because multicast functionality is restored in a fraction of the time previously required.

Multicast Subsecond Convergence Scalability Enhancements

The Multicast Subsecond Convergence feature provides scalability enhancements that improve on the efficiency of handling increases (or decreases) in service users (receivers) and service load (sources or content). Scalability enhancements in this release include the following:

- Improved Internet Group Management Protocol (IGMP) and PIM state maintenance through new timer management techniques
- Improved scaling of the Multicast Source Discovery Protocol (MSDP) Source-Active (SA) cache

The scalability enhancements provide the following benefits:

- Increased potential PIM multicast route (mroute), IGMP, and MSDP SA cache state capacity
- Decreased CPU usage

PIM Router Query Messages

Multicast subsecond convergence allows you to send PIM router query messages (PIM hellos) every few milliseconds. The PIM hello message is used to locate neighboring PIM devices. Before the introduction of this feature, the device could send the PIM hellos only every few seconds. By enabling a device to send PIM hello messages more often, this feature allows the device to discover unresponsive neighbors more quickly. As a result, the device can implement failover or recovery procedures more efficiently.

Reverse Path Forwarding

Unicast Reverse Path Forwarding (RPF) helps to mitigate problems caused by the introduction of malformed or forged IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. Malformed or forged source addresses can indicate denial-of-service (DoS) attacks based on source IP address spoofing.

RPF uses access control lists (ACLs) in determining whether to drop or forward data packets that have malformed or forged IP source addresses. An option in the ACL commands allows system administrators to log information about dropped or forwarded packets. Logging information about forged packets can help in uncovering information about possible network attacks.

Per-interface statistics can help system administrators quickly discover the interface serving as the entry point for an attack on the network.

RPF Checks

PIM is designed to forward IP multicast traffic using the standard unicast routing table. PIM uses the unicast routing table to decide if the source of the IP multicast packet has arrived on the optimal path from the source. This process, the RPF check, is protocol-independent because it is based on the contents of the unicast routing table and not on any particular routing protocol.

Triggered RPF Checks

Multicast subsecond convergence provides the ability to trigger a check of RPF changes for mroute states. This check is triggered by unicast routing changes. By performing a triggered RPF check, users can set the periodic RPF check to a relatively high value (for example, 10 seconds) and still fail over quickly.

The triggered RPF check enhancement reduces the time needed for service to be restored after disruption, such as for single service events (for example, in a situation with one source and one receiver) or as the service scales along any parameter (for example, many sources, many receivers, and many interfaces). This enhancement decreases in time-to-converge PIM (mroute), IGMP, and MSDP (SA cache) states.

RPF Failover

In an unstable unicast routing environment that uses triggered RPF checks, the environment could be constantly triggering RPF checks, which places a burden on the resources of the device. To avoid this problem, use the **ip multicast rpf backoff** command to prevent a second triggered RPF check from occurring for the length of time configured. That is, the PIM “backs off” from another triggered RPF check for a minimum amount of milliseconds as configured by the user.

If the backoff period expires without further routing table changes, PIM then scans for routing changes and accordingly establishes multicast RPF changes. However, if more routing changes occur during the backoff period, PIM doubles the backoff period to avoid overloading the device with PIM RPF changes while the routing table is still converging.

Topology Changes and Multicast Routing Recovery

The Multicast Subsecond Convergence feature set enhances both enterprise and service provider network backbones by providing almost instantaneous recovery of multicast paths after unicast routing recovery.

Because PIM relies on the unicast routing table to calculate its RPF when a change in the network topology occurs, unicast protocols first need to calculate options for the best paths for traffic, and then multicast can determine the best path.

Multicast subsecond convergence allows multicast protocol calculations to finish almost immediately after the unicast calculations are completed. As a result, multicast traffic forwarding is restored substantially faster after a topology change.

How to Configure Multicast Subsecond Convergence

Modifying the Periodic RPF Check Interval

Perform this optional task to modify the intervals at which periodic RPF checks occur.

> **Note** Cisco recommends that you do *not* change the default values for the **ip rpf interval**command. The default values allow subsecond RPF failover. The default interval at which periodic RPF checks occur is 10 seconds.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast rpf interval** *seconds* [**list** *access-list* | **route-map** *route-map*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast rpf interval** *seconds* [**list** *access-list* | **route-map** *route-map*] **Example:** `Device(config)# ip multicast rpf interval 10`	Configures the periodic RPF check intervals to occur at a specified interval, in seconds.

What to Do Next

Proceed to the Configuring PIM RPF Failover Intervals, on page 606 to configure the intervals at which PIM RPF failover will be triggered by changes in the routing tables. Proceed to the Modifying the PIM Router Query Message Interval, on page 607 to modify the interval at which IGMP host query messages are sent. Proceed to the *Verifying Multicast Subsecond Convergence Configurations* to display information about and to verify information regarding the Multicast Subsecond Convergence feature.

Configuring PIM RPF Failover Intervals

Perform this optional task to configure the intervals at which PIM RPF failover will be triggered by changes in the routing tables.

> **Note** Cisco recommends that you do *not* modify the default values for the **ip multicast rpf backoff** command. The default values allow subsecond RPF failover.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast rpf backoff** *minimum maximum* [**disable**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast rpf backoff** *minimum maximum* [**disable**] **Example:** `Device(config)# ip multicast rpf backoff 100 2500`	Configures the minimum and the maximum backoff intervals.

What to Do Next

Proceed to the Modifying the PIM Router Query Message Interval, on page 607 to modify the interval at which IGMP host query messages are sent. Proceed to the *Verifying Multicast Subsecond Convergence Configurations* to display information about and to verify information regarding the Multicast Subsecond Convergence feature.

Modifying the PIM Router Query Message Interval

Perform this task to modify the PIM router query message interval.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *subslot* / *port*
4. **ip pim query-interval** *period* [**msec**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *subslot* / *port* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Specifies the interface and enters interface configuration mode.
Step 4	**ip pim query-interval** *period* [**msec**] **Example:** `Device(config-if)# ip pim query-interval 45`	Configures the frequency at which multicast routers send PIM router query messages.

What to Do Next

Proceed to the *Verifying Multicast Subsecond Convergence Configurations* to display and verify information about the Multicast Subsecond Convergence feature.

Verifying Multicast Subsecond Convergence Configurations

Perform this task to display detailed information about and to verify information regarding the Multicast Subsecond Convergence feature.

SUMMARY STEPS

1. **enable**
2. **show ip pim interface** *type number*
3. **show ip pim neighbor**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ip pim interface** *type number*

Use this command to display information about interfaces configured for PIM.

The following is sample output from the **show ip pim interface** command:

Example:

```
Device# show ip pim interface GigabitEthernet 1/0/0
Address          Interface                Ver/   Nbr    Query  DR     DR
                                          Mode   Count  Intvl  Prior
172.16.1.4       GigabitEthernet1/0/0     v2/S   1      100 ms 1      172.16.1.4
```

Step 3 **show ip pim neighbor**

Use this command to display the PIM neighbors discovered by the Cisco IOS XE software.

The following is sample output from the **show ip pim neighbor** command:

Example:

```
Device# show ip pim neighbor
PIM Neighbor Table
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
172.16.1.3        GigabitEthernet1/0/0     00:03:41/250 msec v2    1 / S
```

Configuration Examples for Multicast Subsecond Convergence

Example Modifying the Periodic RPF Check Interval

In the following example, the **ip multicast rpf interval** has been set to 10 seconds. This command does not show up in **show running-config** output unless the interval value has been configured to be the nondefault value.

```
!
ip multicast-routing
ip multicast rpf interval 10
.
.
.
interface Ethernet0/0
 ip address 172.16.2.1 255.255.255.0
.
.
.
ip pim sparse-mode
!
```

Example Configuring PIM RPF Failover Intervals

In the following example, the **ip multicast rpf backoff** command has been configured with a minimum backoff interval value of 100 and a maximum backoff interval value of 2500. This command does not show up in **show running-config** command output unless the interval value has been configured to be the nondefault value.

```
!
ip multicast-routing
.
.
```

```
.
ip multicast rpf backoff 100 2500
!
!
interface Ethernet0/0
 ip address 172.16.2.1 255.255.255.0
.
.
.
ip pim sparse-mode
!
```

Modifying the PIM Router Query Message Interval Example

In the following example, the **ip pim query-interval** command has been set to 100 milliseconds. This command does not show up in **show running-config** command output unless the interval value has been configured to be the nondefault value.

```
!
interface gigabitethernet0/0/1
 ip address 172.16.2.1 255.255.255.0
ip pim query-interval 100 msec
ip pim sparse-mode
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS IP Multicast commands	Cisco IOS IP Multicast Command Reference

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing standards has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multicast Subsecond Convergence

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 37: Feature Information for Multicast Subsecond Convergence

Feature Name	Releases	Feature Information
Multicast Subsecond Convergence	12.0(22)S 12.2(14)S 12.2(15)T Cisco IOS XE Release 2.1 15.0(1)S Cisco IOS XE Release 3.1.0SG Cisco IOS XE Release 3.2SE	The Multicast Subsecond Convergence feature comprises a comprehensive set of features and protocol enhancements that provide for improved scalability and convergence in multicast-based services. This feature set provides for the ability to scale to larger services levels and to recover multicast forwarding after service failure in subsecond time frames. The following commands were introduced or modified: **debug ip mrouting**, **debug ip pim**, **ip multicast rpf backoff**, **ip multicast rpf interval**, **ip pim query-interval**, **show ip pim interface**, **show ip pim neighbor**, **show ip rpf events**.

CHAPTER 47

IP Multicast Load Splitting across Equal-Cost Paths

This module describes how to load split IP multicast traffic from different sources, or from different sources and groups, over Equal Cost Multipath (ECMP) to take advantage of multiple paths through the network.

Prerequisites for IP Multicast Load Splitting across Equal-Cost Paths

IP multicast is enabled on the device using the tasks described in the “Configuring Basic IP Multicast” module of the *IP Multicast: PIM Configuration Guide*.

Information About IP Multicast Load Splitting across Equal-Cost Paths

Load Splitting Versus Load Balancing

Load splitting and load balancing are not the same. Load splitting provides a means to randomly distribute (*, G) and (S, G) traffic streams across multiple equal-cost reverse path forwarding (RPF) paths, which does not necessarily result in a balanced IP multicast traffic load on those equal-cost RPF paths. By randomly distributing (*, G) and (S, G) traffic streams, the methods used for load splitting IP multicast traffic attempt to distribute an equal amount of traffic flows on each of the available RPF paths not by counting the flows, but, rather, by making a pseudorandom decision. These methods are collectively referred to as equal-cost multipath (ECMP) multicast load splitting methods and result in better load-sharing in networks where there are many traffic streams that utilize approximately the same amount of bandwidth.

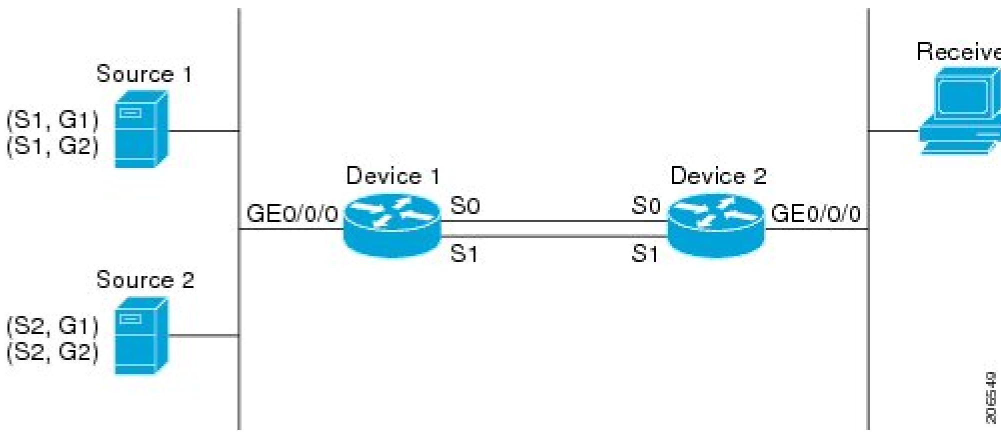
If there are just a few (S, G) or (*, G) states flowing across a set of equal-cost links, the chance that they are well balanced is quite low. To overcome this limitation, precalculated source addresses--for (S, G) states or rendezvous point (RP) addresses for (*, G) states, can be used to achieve a reasonable form of load balancing. This limitation applies equally to the per-flow load splitting in Cisco Express Forwarding (CEF) or with EtherChannels: As long as there are only a few flows, those methods of load splitting will not result in good load distribution without some form of manual engineering.

Default Behavior for IP Multicast When Multiple Equal-Cost Paths Exist

By default, for Protocol Independent Multicast sparse mode (PIM-SM), Source Specific Multicast (PIM-SSM), bidirectional PIM (bidir-PIM), and PIM dense mode (PIM-DM) groups, if multiple equal-cost paths are available, Reverse Path Forwarding (RPF) for IPv4 multicast traffic is based on the PIM neighbor with the highest IP address. This method is referred to as the highest PIM neighbor behavior. This behavior is in accordance with RFC 2362 for PIM-SM, but also applies to PIM-SSM, PIM-DM, and bidir-PIM.

The figure illustrates a sample topology that is used in this section to explain the default behavior for IP multicast when multiple equal-cost paths exist.

Figure 67: Default Behavior for IP Multicast When Multiple Equal-Cost Paths Exist



In the figure, two sources, S1 and S2, are sending traffic to IPv4 multicast groups, G1 and G2. Either PIM-SM, PIM-SSM, or PIM-DM can be used in this topology. If PIM-SM is used, assume that the default of 0 for the **ip pim spt-threshold** command is being used on Device 2, that an Interior Gateway Protocol (IGP) is being run, and that the output of the **show ip route** command for S1 and for S2 (when entered on Device 2) displays serial interface 0 and serial interface 1 on Device 1 as equal-cost next-hop PIM neighbors of Device 2.

Without further configuration, IPv4 multicast traffic in the topology illustrated in the figure would always flow across one serial interface (either serial interface 0 or serial interface 1), depending on which interface has the higher IP address. For example, suppose that the IP addresses configured on serial interface 0 and serial interface 1 on Device 1 are 10.1.1.1 and 10.1.2.1, respectively. Given that scenario, in the case of PIM-SM and PIM-SSM, Device 2 would always send PIM join messages towards 10.1.2.1 and would always receive IPv4 multicast traffic on serial interface 1 for all sources and groups shown in the figure. In the case of PIM-DM, Device 2 would always receive IP multicast traffic on serial Interface 1, only that in this case, PIM join messages are not used in PIM-DM; instead Device 2 would prune the IP multicast traffic across serial interface 0 and would receive it through serial interface 1 because that interface has the higher IP address on Device 1.

IPv4 RPF lookups are performed by intermediate multicast device to determine the RPF interface and RPF neighbor for IPv4 (*,G) and (S, G) multicast routes (trees). An RPF lookup consists of RPF route-selection and route-path-selection. RPF route-selection operates solely on the IP unicast address to identify the root of the multicast tree. For (*, G) routes (PIM-SM and Bidir-PIM), the root of the multicast tree is the RP address for the group G; for (S, G) trees (PIM-SM, PIM-SSM and PIM-DM), the root of the multicast tree is the source S. RPF route-selection finds the best route towards the RP or source in the routing information base (RIB), and, if configured (or available), the Distance Vector Multicast Routing Protocol (DVMRP) routing table, the Multiprotocol Border Gateway Protocol (MBGP) routing table or configured static mroutes. If the resulting route has only one available path, then the RPF lookup is complete, and the next-hop device and interface of the route become the RPF neighbor and RPF interface of this multicast tree. If the route has more than one path available, then route-path-selection is used to determine which path to choose.

For IP multicast, the following route-path-selection methods are available:

Note All methods but the default method of route-path-selection available in IP multicast enable some form of ECMP multicast load splitting.

- Highest PIM neighbor--This is the default method; thus, no configuration is required. If multiple equal-cost paths are available, RPF for IPv4 multicast traffic is based on the PIM neighbor with the highest IP address; as a result, without configuration, ECMP multicast load splitting is disabled by default.
- ECMP multicast load splitting method based on source address--You can configure ECMP multicast load splitting using the **ip multicast multipath** command. Entering this form of the **ip multicast multipath** command enables ECMP multicast load splitting based on source address using the S-hash algorithm. For more information, see the ECMP Multicast Load Splitting Based on Source Address Using the S-Hash Algorithm, on page 616 section.
- ECMP multicast load splitting method based on source and group address--You can configure ECMP multicast load splitting using the **ip multicast multipath** command with the **s-g-hash** and **basic** keywords. Entering this form of the **ip multicast multipath** command enables ECMP multicast load splitting based on source and group address using the basic S-G-hash algorithm. For more information, see the ECMP Multicast Load Splitting Based on Source and Group Address Using the Basic S-G-Hash Algorithm, on page 616 section.
- ECMP multicast load splitting method based on source, group, and next-hop address--You can configure ECMP multicast load splitting using the **ip multicast multipath** command with the **s-g-hash** and **next-hop-based** keywords. Entering this form of the command enables ECMP multicast load splitting based on source, group, and next-hop address using the next-hop-based S-G-hash algorithm. For more information, see the ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address, on page 617 section.

The default behavior (the highest PIM neighbor behavior) does not result in any form of ECMP load-splitting in IP multicast, but instead selects the PIM neighbor that has the highest IP address among the next-hop PIM neighbors for the available paths. A next hop is considered to be a PIM neighbor when it displays in the output of the **show ip pim neighbor** command, which is the case when PIM hello messages have been received from it and have not timed out. If none of the available next hops are PIM neighbors, then simply the next hop with the highest IP address is chosen.

Methods to Load Split IP Multicast Traffic

In general, the following methods are available to load split IP multicast traffic:

- You can enable ECMP multicast load splitting based on source address, based on source and group address, or based on source, group, and next-hop address. After the equal-cost paths are recognized, ECMP multicast load splitting operates on a per (S, G) basis, rather than a per packet basis as in unicast traffic.
- Alternative methods to load split IP multicast are to consolidate two or more equal-cost paths into a generic routing encapsulation (GRE) tunnel and allow the unicast routing protocol to perform the load splitting, or to load split across bundle interfaces, such as Fast or Gigabit EtherChannel interfaces, Multilink PPP (MLPPP) link bundles, or Multilink Frame Relay (FR.16) link bundles.

Overview of ECMP Multicast Load Splitting

By default, ECMP multicast load splitting of IPv4 multicast traffic is disabled. ECMP multicast load splitting can be enabled using the **ip multicast multipath** command.

ECMP Multicast Load Splitting Based on Source Address Using the S-Hash Algorithm

ECMP multicast load splitting traffic based on source address uses the S-hash algorithm, enabling the RPF interface for each (*, G) or (S, G) state to be selected among the available equal-cost paths, depending on the RPF address to which the state resolves. For an (S, G) state, the RPF address is the source address of the state; for a (*, G) state, the RPF address is the address of the RP associated with the group address of the state.

When ECMP multicast load splitting based on source address is configured, multicast traffic for different states can be received across more than just one of the equal-cost interfaces. The method applied by IPv4 multicast is quite similar in principle to the default per-flow load splitting in IPv4 CEF or the load splitting used with Fast and Gigabit EtherChannels. This method of ECMP multicast load splitting, however, is subject to polarization.

ECMP Multicast Load Splitting Based on Source and Group Address Using the Basic S-G-Hash Algorithm

ECMP multicast load splitting based on source and group address uses a simple hash, referred to as the basic S-G-hash algorithm, which is based on source and group address. The basic S-G-hash algorithm is predictable because no randomization is used in coming up with the hash value. The S-G-hash mechanism, however, is subject to polarization because for a given source and group, the same hash is always picked irrespective of the device this hash is being calculated on.

Note The basic S-G-hash algorithm ignores bidir-PIM groups.

Predictability As a By-Product of Using the S-Hash and Basic S-G-Hash Algorithms

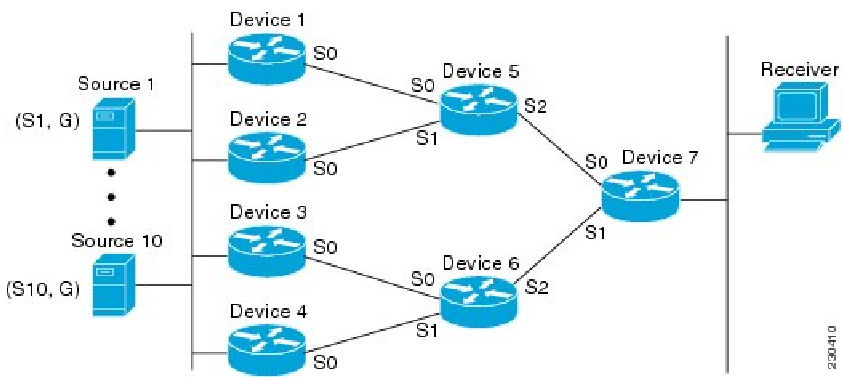
The method used by ECMP multicast load splitting in IPv4 multicast allows for consistent load splitting in a network where the same number of equal-cost paths are present in multiple places in a topology. If an RP address or source addresses are calculated once to have flows split across N paths, then they will be split across those N paths in the same way in all places in the topology. Consistent load splitting allows for predictability, which, in turn, enables load splitting of IPv4 multicast traffic to be manually engineered.

Polarization As a By-Product of Using the S-Hash and Basic S-G-Hash Algorithms

The hash mechanism used in IPv4 multicast to load split multicast traffic by source address or by source and group address is subject to a problem usually referred to as polarization. A by-product of ECMP multicast load splitting based on source address or on source and group address, polarization is a problem that prevents routers in some topologies from effectively utilizing all available paths for load splitting.

The figure illustrates a sample topology that is used in this section to explain the problem of polarization when configuring ECMP multicast load splitting based on source address or on source and group address.

Figure 68: Polarization Topology



In the topology illustrated in the figure, notice that Router 7 has two equal-cost paths towards the sources, S1 to S10, through Router 5 and Router 6. For this topology, suppose that ECMP multicast load splitting is enabled with the **ip multicast multipath** command on all routers in the topology. In that scenario, Router 7 would apply equal-cost load splitting to the 10 (S, G) states. The problem of polarization in this scenario would affect Router 7 because that router would end up choosing serial interface 0 on Router 5 for sources S1 to S5 and serial interface 1 on Router 6 for sources S6 to S10. The problem of polarization, furthermore, would also affect Router 5 and Router 6 in this topology. Router 5 has two equal-cost paths for S1 to S5 through serial interface 0 on Router 1 and serial interface 1 on Router 2. Because Router 5 would apply the same hash algorithm to select which of the two paths to use, it would end up using just one of these two upstream paths for sources S1 to S5; that is, either all the traffic would flow across Router 1 and Router 5 or across Router 2 and Router 5. It would be impossible in this topology to utilize Router 1 and Router 5 and Router 2 and Router 5 for load splitting. Likewise, the polarization problem would apply to Router 3 and Router 6 and Router 4 and Router 6; that is, it would be impossible in this topology to utilize both Router 3 and Router 6 and Router 4 and Router 6 for load splitting.

ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address

Configuring ECMP multicast load splitting based on source, group, and next-hop address enables a more complex hash, the next-hop-based S-G-hash algorithm, which is based on source, group, and next-hop address. The next-hop-based S-G-hash algorithm is predictable because no randomization is used in calculating the hash value. Unlike the S-hash and basic S-G-hash algorithms, the hash mechanism used by the next-hop-based S-G-hash algorithm is not subject to polarization.

Note The next-hop-based S-G-hash algorithm in IPv4 multicast is the same algorithm used in IPv6 ECMP multicast load splitting, which, in turn, utilizes the same hash function used for PIM-SM bootstrap device (BSR).

The next-hop-based hash mechanism does not produce polarization and also maintains better RPF stability when paths fail. These benefits come at the cost that the source or RP IP addresses cannot be used to reliably predict and engineer the outcome of load splitting when the next-hop-based S-G-hash algorithm is used. Because many customer networks have implemented equal-cost multipath topologies, the manual engineering of load splitting, thus, is not a requirement in many cases. Rather, it is more of a requirement that the default behavior of IP multicast be similar to IP unicast; that is, it is expected that IP multicast use multiple equal-cost paths on a best-effort basis. Load splitting for IPv4 multicast, therefore, could not be enabled by default because of the anomaly of polarization.

Note Load splitting for CEF unicast also uses a method that does not exhibit polarization and likewise cannot be used to predict the results of load splitting or engineer the outcome of load splitting.

The next-hop-based hash function avoids polarization because it introduces the actual next-hop IP address of PIM neighbors into the calculation, so the hash results are different for each device, and in effect, there is no problem of polarization. In addition to avoiding polarization, this hash mechanism also increases stability of the RPF paths chosen in the face of path failures. Consider a device with four equal-cost paths and a large number of states that are load split across these paths. Suppose that one of these paths fails, leaving only three available paths. With the hash mechanism used by the polarizing hash mechanisms (the hash mechanism used by the S-hash and basic S-G-hash algorithms), the RPF paths of all states would likely reconverge and thus change between those three paths, especially those paths that were already using one of those three paths. These states, therefore, may unnecessarily change their RPF interface and next-hop neighbor. This problem exists simply because the chosen path is determined by taking the total number of paths available into consideration by the algorithm, so once a path changes, the RPF selection for all states is subject to change too. For the next-hop-based hash mechanism, only the states that were using the changed path for RPF would need to reconverge onto one of the three remaining paths. The states that were already using one of those paths would not change. If the fourth path came back up, the states that initially used it would immediately reconverge back to that path without affecting the other states.

Note The next-hop-based S-G-hash algorithm ignores bidir-PIM groups.

Effect of ECMP Multicast Load Splitting on PIM Neighbor Query and Hello Messages for RPF Path Selection

If load splitting of IP multicast traffic over ECMP is *not* enabled and there are multiple equal-cost paths towards an RP or a source, IPv4 multicast will first elect the highest IP address PIM neighbor. A PIM neighbor is a device from which PIM hello (or PIMv1 query) messages are received. For example, consider a device that has two equal-cost paths learned by an IGP or configured through two static routes. The next hops of these two paths are 10.1.1.1 and 10.1.2.1. If both of these next-hop devices send PIM hello messages, then 10.1.2.1 would be selected as the highest IP address PIM neighbor. If only 10.1.1.1 sends PIM hello messages, then 10.1.1.1 would be selected. If neither of these devices sends PIM hello messages, then 10.1.2.1 would be selected. This deference to PIM hello messages allows the construction of certain types of dynamic failover scenarios with only static multicast routes (mroutes); it is otherwise not very useful.

Note For more information about configuring static mroutes, see the Configuring Multiple Static Mroutes in Cisco IOS configuration note on the Cisco IOS IP multicast FTP site, which is available at: ftp://ftpeng.cisco.com/ipmulticast /config-notes/static-mroutes.txt.

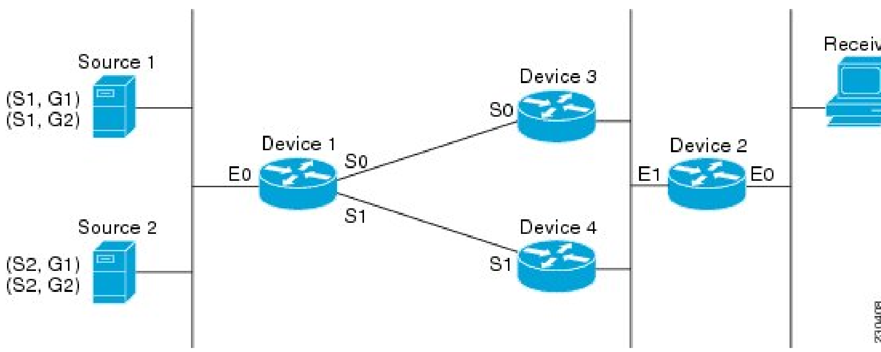
When load splitting of IP multicast traffic over ECMP is enabled, the presence of PIM hello message from neighbors is not considered; that is, the chosen RPF neighbor does not depend on whether or not PIM hello messages are received from that neighbor--it only depends on the presence or absence of an equal-cost route entry.

Effect of ECMP Multicast Loading Splitting on Assert Processing in PIM-DM and DF Election in Bidir-PIM

The **ip multicast multipath** command only changes the RPF selection on the downstream device; it does not have an effect on designated forwarder (DF) election in bidir-PIM or the assert processing on upstream devices in PIM-DM.

The figure illustrates a sample topology that is used in this section to explain the effect of ECMP multicast load splitting on assert processing in PIM-DM and DF election in bidir-PIM.

Figure 69: ECMP Multicast Load Splitting and Assert Processing in PIM-DM and DF Election in Bidir-PIM



In the figure, Device 2 has two equal-cost paths to S1 and S2 and the RP addresses on Device 1. Both paths are across Gigabit Ethernet interface 1/0/0: one path towards Device 3 and one path towards Device 4. For PIM-SM and PIM-SSM (*, G) and (S, G) RPF selection, there is no difference in the behavior of Device 2 in this topology versus Device 2 in the topology illustrated in the figure. There is, however, a difference when using PIM-DM or bidir-PIM.

If PIM-DM is used in the topology illustrated in the figure, Device 3 and Device 4 would start flooding traffic for the states onto Gigabit Ethernet interface 1/0/0 and would use the PIM assert process to elect one device among them to forward the traffic and to avoid traffic duplication. As both Device 3 and Device 4 would have the same route cost, the device with the higher IP address on Gigabit Ethernet interface 1/0/0 would always win the assert process. As a result, if PIM-DM is used in this topology, traffic would not be load split across Device 3 and Device 4.

If bidir-PIM is used in the topology illustrated in the figure, a process called DF election would take place between Device 2, Device 3, and Device 4 on Gigabit Ethernet interface 1/0/0. The process of DF election would elect one device for each RP to forward traffic across Gigabit Ethernet interface 1/0/0 for any groups using that particular RP, based on the device with the highest IP address configured for that interface. Even if multiple RPs are used (for example one for G1 and another one for G2), the DF election for those RPs would always be won by the device that has the higher IP address configured on Gigabit Ethernet interface 1/0/0 (either Device 3 or Device 4 in this topology). The election rules used for DF election are virtually the same as the election rules used for the PIM assert process, only the protocol mechanisms to negotiate them are more refined for DF election (in order to return the results more expediently). As a result, when bidir-PIM is used in this topology, load splitting would always occur across Gigabit Ethernet interface 1/0/0.

The reason that ECMP multicast load splitting does influence the RPF selection but not the assert process in PIM-DM or DF election in bidir-PIM is because both the assert process and DF election are cooperative processes that need to be implemented consistently between participating devices. Changing them would require some form of protocol change that would also need to be agreed upon by the participating devices. RPF selection is a purely device local policy and, thus, can be enabled or disabled without protocol changes individually on each device.

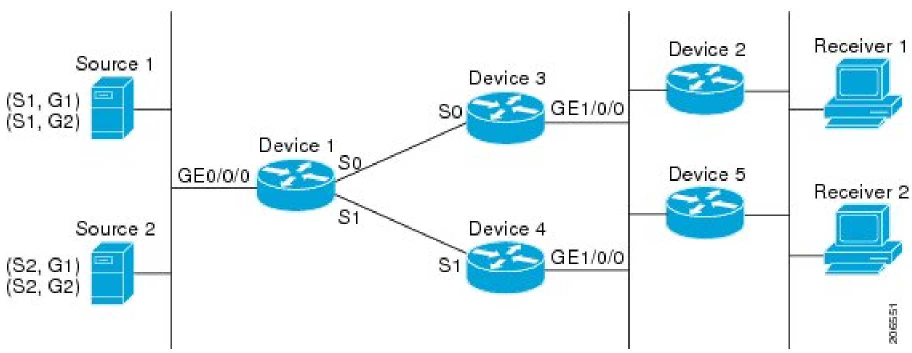
For PIM-DM and bidir-PIM, configuring ECMP multicast load splitting with the **ip multicast multipath** command is only effective in topologies where the equal-cost paths are not upstream PIM neighbors on the same LAN, but rather neighbors on different LANs or point-to-point links.

Effect of ECMP Multicast Load Splitting on the PIM Assert Process in PIM-SM and PIM-SSM

There are also cases where ECMP multicast load splitting with the **ip multicast multipath** command can become ineffective due to the PIM assert process taking over, even when using PIM-SM with (*, G) or (S, G) forwarding or PIM-SSM with (S, G) forwarding.

The figure illustrates a sample topology that is used in this section to explain the effect of ECMP multicast load splitting on the PIM assert process in PIM-SM and PIM-SSM.

Figure 70: ECMP Multicast Load Splitting and the PIM Assert Process in PIM-SM and PIM-SSM



In the topology illustrated in the figure, if both Device 2 and Device 5 are Cisco devices and are consistently configured for ECMP multicast load splitting with the **ip multicast multipath** command, then load splitting would continue to work as expected; that is, both devices would have Device 3 and Device 4 as equal-cost next hops and would sort the list of equal-cost paths in the same way (by IP address). When applying the multipath hash function, for each (S, G) or (*, G) state, they would choose the same RPF neighbor (either Device 3 or Device 4) and send their PIM joins to this neighbor.

If Device 5 and Device 2 are inconsistently configured with the **ip multicast multipath** command, or if Device 5 is a third-party device, then Device 2 and Device 5 may choose different RPF neighbors for some (*, G) or (S, G) states. For example Device 2 could choose Device 3 for a particular (S, G) state or Device 5 could choose Device 4 for a particular (S, G) state. In this scenario, Device 3 and Device 4 would both start to forward traffic for that state onto Gigabit Ethernet interface 1/0/0, see each other's forwarded traffic, and--to avoid traffic duplication--start the assert process. As a result, for that (S, G) state, the device with the higher IP address for Gigabit Ethernet interface 1/0/0 would forward the traffic. However, both Device 2 and Device 5 would be tracking the winner of the assert election and would send their PIM joins for that state to this assert winner, even if this assert winner is not the same device as the one that they calculated in their RPF selection. For PIM-SM and PIM-SSM, therefore, the operation of ECMP multicast load splitting can only be guaranteed when all downstream devices on a LAN are consistently configured Cisco devices.

ECMP Multicast Load Splitting and Reconvergence When Unicast Routing Changes

When unicast routing changes, all IP multicast routing states reconverge immediately based on the available unicast routing information. Specifically, if one path goes down, the remaining paths reconverge immediately, and if the path comes up again, multicast forwarding will subsequently reconverge to the same RPF paths that were used before the path failed. Reconvergence occurs whether load splitting of IP multicast traffic over ECMP is configured or not.

Use of BGP with ECMP Multicast Load Splitting

ECMP multicast load splitting works with RPF information learned through BGP in the same way as with RPF information learned from other protocols: It chooses one path out of the multiple paths installed by the protocol. The main difference with BGP is that it only installs a single path, by default. For example, when a BGP speaker learns two identical external BGP (eBGP) paths for a prefix, it will choose the path with the lowest device ID as the best path. The best path is then installed in the IP routing table. If BGP multipath support is enabled and the eBGP paths are learned from the same neighboring AS, instead of picking the single best path, BGP installs multiple paths in the IP routing table. By default, BGP will install only one path to the IP routing table.

To leverage ECMP multicast load splitting for BGP learned prefixes, you must enable BGP multipath. Once configured, when BGP installs the remote next-hop information, RPF lookups will execute recursively to find the best next hop towards that BGP next hop (as in unicast). If for example there is only a single BGP path for a given prefix, but there are two IGP paths to reach that BGP next hop, then multicast RPF will correctly load split between the two different IGP paths.

Use of ECMP Multicast Load Splitting with Static Mroutes

If it is not possible to use an IGP to install equal cost routes for certain sources or RPs, static routes can be configured to specify the equal-cost paths for load splitting. You cannot use static mroutes to configure equal-cost paths because the software does not support the configuration of one static mroute per prefix. There are some workarounds for this limitation using recursive route lookups but the workarounds cannot be applied to equal-cost multipath routing.

Note For more information about configuring static mroutes, see the Configuring Multiple Static Mroutes in Cisco IOS configuration note on the Cisco IOS IP multicast FTP site at ftp://ftpeng.cisco.com/ipmulticast/config-notes/static-mroutes.txt.

You can specify only static mroutes for equal-cost multipaths in IPv4 multicast; however, those static mroutes would only apply to multicast, or you can specify that the equal-cost multipaths apply to both unicast and multicast routing. In IPv6 multicast, there is no such restriction. Equal-cost multipath mroutes can be configured for static IPv6 mroutes that apply to only unicast routing, only multicast routing, or both unicast and multicast routing.

Alternative Methods of Load Splitting IP Multicast Traffic

Load splitting of IP multicast traffic can also be achieved by consolidating multiple parallel links into a single tunnel over which the multicast traffic is then routed. This method of load splitting is more complex to configure than ECMP multicast load splitting. One such case where configuring load splitting across equal-cost paths using GRE links can be beneficial is the case where the total number of (S, G) or (*, G) states is so small and the bandwidth carried by each state so variable that even the manual engineering of the source or RP addresses cannot guarantee the appropriate load splitting of the traffic.

Note With the availability of ECMP multicast load splitting, tunnels typically only need to be used if per-packet load sharing is required.

IP multicast traffic can also be used to load split across bundle interfaces, such as Fast or Gigabit EtherChannel interfaces, MLPPP link bundles or Multilink Frame Relay (FRF.16) bundles. GRE or other type of tunnels can also constitute such forms of Layer 2 link bundles. Before using such an Layer 2 mechanism, it is necessary to understand how unicast and multicast traffic is load split.

Before load splitting IP multicast traffic across equal-cost paths over a tunnel, you must configure CEF per-packet load balancing or else the GRE packets will not be load balanced per packet.

How to Load Split IP Multicast Traffic over ECMP

Enabling ECMP Multicast Load Splitting

Perform the following tasks to load split IP multicast traffic across multiple equal-cost paths, based on source address.

If two or more equal-cost paths from a source are available, unicast traffic will be load split across those paths. However, by default, multicast traffic is not load split across multiple equal-cost paths. In general, multicast traffic flows down from the RPF neighbor. According to PIM specifications, this neighbor must have the highest IP address if more than one neighbor has the same metric.

Configuring load splitting with the **ip multicast multipath** command causes the system to load split multicast traffic across multiple equal-cost paths based on source address using the S-hash algorithm. When the **ip multicast multipath** command is configured and multiple equal-cost paths exist, the path in which multicast traffic will travel is selected based on the source IP address. Multicast traffic from different sources will be load split across the different equal-cost paths. Load splitting will not occur across equal-cost paths for multicast traffic from the same source sent to different multicast groups.

Note The **ip multicast multipath**command load splits the traffic and does not load balance the traffic. Traffic from a source will use only one path, even if the traffic far outweighs traffic from other sources.

Prerequisites for IP Multicast Load Splitting - ECMP

- You must have an adequate number of sources (at least more than two sources) to enable ECMP multicast load splitting based on source address.
- You must have multiple paths available to the RP to configure ECMP multicast load splitting.

> **Note** Use the **show ip route** command with either the IP address of the source for the *ip-address*argument or the IP address of the RP to validate that there are multiple paths available to the source or RP, respectively. If you do not see multiple paths in the output of the command, you will not be able to configure ECMP multicast load splitting.

- When using PIM-SM with shortest path tree (SPT) forwarding, the T-bit mus be set for the forwarding of all (S, G) states.
- Before configuring ECMP multicast load splitting, it is best practice to use the **show ip rpf** command to validate whether sources can take advantage of IP multicast multipath capabilities.
- BGP does not install multiple equal-cost paths by default. Use the **maximum-paths** command to configure multipath (for example in BGP). For more information, see the Use of BGP with ECMP Multicast Load Splitting, on page 621 section.

Restrictions

- If two or more equal-cost paths from a source are available, unicast traffic will be load split across those paths. However, by default, multicast traffic is not load split across multiple equal-cost paths. In general, multicast traffic flows down from the RPF neighbor. According to PIM specifications, this neighbor must have the highest IP address if more than one neighbor has the same metric.
- The **ip multicast multipath** command does not support configurations in which the same PIM neighbor IP address is reachable through multiple equal-cost paths. This situation typically occurs if unnumbered interfaces are used. Use different IP addresses for all interfaces when configuring the **ip multicast multipath** command.
- The **ip multicast multipath** command load splits the traffic and does not load balance the traffic. Traffic from a source will use only one path, even if the traffic far outweighs traffic from other sources.

Enabling ECMP Multicast Load Splitting Based on Source Address

Perform this task to enable ECMP multicast load splitting of multicast traffic based on source address (using the S-hash algorithm) to take advantage of multiple paths through the network. The S-hash algorithm is predictable because no randomization is used in calculating the hash value. The S-hash algorithm, however, is subject to polarization because for a given source, the same hash is always picked irrespective of the device on which the hash is being calculated.

> **Note** Enable ECMP multicast load splitting on the device that is to be the receiver for traffic from more than one incoming interfaces, which is opposite to unicast routing. From the perspective of unicast, multicast is active on the sending device connecting to more than one outgoing interfaces.

Before you begin

- You must have an adequate number of sources (at least more than two sources) to enable ECMP multicast load splitting based on source address.
- You must have multiple paths available to the RP to configure ECMP multicast load splitting.

Note Use the **show ip route** command with either the IP address of the source for the *ip-address*argument or the IP address of the RP to validate that there are multiple paths available to the source or RP, respectively. If you do not see multiple paths in the output of the command, you will not be able to configure ECMP multicast load splitting.

- When using PIM-SM with shortest path tree (SPT) forwarding, the T-bit mus be set for the forwarding of all (S, G) states.
- Before configuring ECMP multicast load splitting, it is best practice to use the **show ip rpf** command to validate whether sources can take advantage of IP multicast multipath capabilities.
- BGP does not install multiple equal-cost paths by default. Use the **maximum-paths** command to configure multipath (for example in BGP). For more information, see the Use of BGP with ECMP Multicast Load Splitting, on page 621 section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast multipath**
4. Repeat step 3 on all the devices in a redundant topology.
5. **exit**
6. **show ip rpf** *source-address* [*group-address*]
7. **show ip route** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast multipath** **Example:**	Enables ECMP multicast load splitting based on source address using the S-hash algorithm.

	Command or Action	Purpose
	`Device(config)# ip multicast multipath`	• Because this command changes the way an RPF neighbor is selected, it must be configured consistently on all devices in a redundant topology to avoid looping. • This command does not support configurations in which the same PIM neighbor IP address is reachable through multiple equal-cost paths. This situation typically occurs if unnumbered interfaces are used. Use a different IP address for each interface in a device on which this command is to be configured. • This command load splits the traffic and does not load balance the traffic. Traffic from a source will use only one path, even if the traffic far outweighs traffic from other sources.
Step 4	Repeat step 3 on all the devices in a redundant topology.	--
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip rpf** *source-address* [*group-address*] **Example:** `Device# show ip rpf 10.1.1.2`	(Optional) Displays the information that IP multicast routing uses to perform the RPF check. • Use this command to verify RPF selection so as to ensure that IP multicast traffic is being properly load split.
Step 7	**show ip route** *ip-address* **Example:** `Device# show ip route 10.1.1.2`	(Optional) Displays the current state of the IP routing table. • Use this command to verify that there multiple paths available to a source or RP for ECMP multicast load splitting. • For the *ip-address* argument, enter the IP address of a source to validate that there are multiple paths available to the source (for shortest path trees) or the IP address of an RP to validate that there are multiple paths available to the RP (for shared trees).

Enabling ECMP Multicast Load Splitting Based on Source and Group Address

Perform this task to enable ECMP multicast load splitting of multicast traffic based on source and group address (using the basic S-G-hash algorithm) to take advantage of multiple paths through the network. The basic S-G-hash algorithm is predictable because no randomization is used in calculating the hash value.The basic S-G-hash algorithm, however, is subject to polarization because for a given source and group, the same hash is always picked irrespective of the device on which the hash is being calculated.

The basic S-G-hash algorithm provides more flexible support for ECMP multicast load splitting than the the S-hash algorithm. Using the basic S-G-hash algorithm for load splitting, in particular, enables multicast traffic from devices that send many streams to groups or that broadcast many channels, such as IPTV servers or MPEG video servers, to be more effectively load split across equal-cost paths.

Note Enable ECMP multicast load splitting on the device that is to be the receiver for traffic from more than one incoming interfaces, which is opposite to unicast routing. From the perspective of unicast, multicast is active on the sending device connecting to more than one outgoing interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast multipath s-g-hash basic**
4. Repeat Step 3 on all the devices in a redundant topology.
5. **exit**
6. **show ip rpf** *source-address* [*group-address*]
7. **show ip route** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast multipath s-g-hash basic** **Example:** `Device(config)# ip multicast multipath s-g-hash basic`	Enables ECMP multicast load splitting based on source and group address using the basic S-G-hash algorithm. • Because this command changes the way an RPF neighbor is selected, it must be configured consistently on all devices in a redundant topology to avoid looping.
Step 4	Repeat Step 3 on all the devices in a redundant topology.	--
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show ip rpf** *source-address* [*group-address*] **Example:** `Device# show ip rpf 10.1.1.2`	(Optional) Displays the information that IP multicast routing uses to perform the RPF check. • Use this command to verify RPF selection so as to ensure that IP multicast traffic is being properly load split.
Step 7	**show ip route** *ip-address* **Example:** `Device# show ip route 10.1.1.2`	(Optional) Displays the current state of the IP routing table. • Use this command to verify that there multiple paths available to a source or RP for ECMP multicast load splitting. • For the *ip-address* argument, enter the IP address of a source to validate that there are multiple paths available to the source (for shortest path trees) or the IP address of an RP to validate that there are multiple paths available to the RP (for shared trees).

Enabling ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address

Perform this task to enable ECMP multicast load splitting of multicast traffic based on source, group, and next-hop address (using the next-hop-based S-G-hash algorithm) to take advantage of multiple paths through the network. The next-hop-based S-G-hash algorithm is predictable because no randomization is used in calculating the hash value. Unlike the S-hash and basic S-G-hash algorithms, the hash mechanism used by the next-hop-based S-G-hash algorithm is not subject to polarization.

The next-hop-based S-G-hash algorithm provides more flexible support for ECMP multicast load splitting than S-hash algorithm and eliminates the polarization problem. Using the next-hop-based S-G-hash algorithm for ECMP multicast load splitting enables multicast traffic from devices that send many streams to groups or that broadcast many channels, such as IPTV servers or MPEG video servers, to be more effectively load split across equal-cost paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast multipath s-g-hash next-hop-based**
4. Repeat Steps 1 through 3 on all the routers in a redundant topology.
5. **end**
6. **show ip rpf** *source-address* [*group-address*]
7. **show ip route** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast multipath s-g-hash next-hop-based** **Example:** `Router(config)# ip multicast multipath s-g-hash next-hop-based`	Enables ECMP multicast load splitting based on source, group, and next-hop-address using the next-hop-based S-G-hash algorithm. • Because this command changes the way an RPF neighbor is selected, it must be configured consistently on all routers in a redundant topology to avoid looping. **Note** Be sure to enable the **ip multicast multipath** command on the router that is supposed to be the receiver for traffic from more than one incoming interfaces, which is opposite to unicast routing. From the perspective of unicast, multicast is active on the sending router connecting to more than one outgoing interfaces.
Step 4	Repeat Steps 1 through 3 on all the routers in a redundant topology.	--
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip rpf** *source-address* [*group-address*] **Example:** `Router# show ip rpf 10.1.1.2`	(Optional) Displays the information that IP multicast routing uses to perform the RPF check. • Use this command to verify RPF selection so as to ensure that IP multicast traffic is being properly load split.
Step 7	**show ip route** *ip-address* **Example:** `Router# show ip route 10.1.1.2`	(Optional) Displays the current state of the IP routing table. • Use this command to verify that there multiple paths available to a source or RP for ECMP multicast load splitting. • For the *ip-address* argument, enter the IP address of a source to validate that there are multiple paths available to the source (for shortest path trees) or the IP address of an RP to validate that there are multiple paths available to the RP (for shared trees).

Configuration Examples for Load Splitting IP Multicast Traffic over ECMP

Example Enabling ECMP Multicast Load Splitting Based on Source Address

The following example shows how to enable ECMP multicast load splitting on a router based on source address using the S-hash algorithm:

```
ip multicast multipath
```

Example Enabling ECMP Multicast Load Splitting Based on Source and Group Address

The following example shows how to enable ECMP multicast load splitting on a router based on source and group address using the basic S-G-hash algorithm:

```
ip multicast multipath s-g-hash basic
```

Example Enabling ECMP Multicast Load Splitting Based on Source Group and Next-Hop Address

The following example shows how to enable ECMP multicast load splitting on a router based on source, group, and next-hop address using the next-hop-based S-G-hash algorithm:

```
ip multicast multipath s-g-hash next-hop-based
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS IP Multicast commands	Cisco IOS IP Multicast Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 4601	Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing standards has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Load Splitting IP Multicast Traffic over ECMP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 38: Feature Information for Load Splitting IP Multicast Traffic over ECMP

Feature Name	Releases	Feature Information
IP Multicast Load Splitting across Equal-Cost Paths	--	The IP Multicast Load Splitting Across Equal Paths feature enables load splitting of IP multicast traffic across equal-cost paths. Prior to this feature, when there were equal-cost paths between routers, IP multicast packets traversed only one path. If a tunnel was configured, the same next hop was always used, and no load splitting occurred. The following commands were introduced or modified: **ip multicast multipath**.

Feature Name	Releases	Feature Information
IP Multicast Load Splitting--Equal Cost Multipath (ECMP) Using S, G and Next Hop	12.2(33)SRB 15.0(1)M 15.0(1)S	The IP Multicast Load Splitting--Equal Cost Multipath (ECMP) Using S, G and Next Hop feature introduces more flexible support for ECMP multicast load splitting by adding support for load splitting based on source and group address and on source, group, and next-hop address. This feature enables multicast traffic from devices that send many streams to groups or that broadcast many channels, such as IPTV servers or MPEG video servers, to be more effectively load split across equal-cost paths. Prior to the introduction of this feature, the Cisco IOS software only supported ECMP multicast load splitting based on source address, which restricted multicast traffic sent by a single source to multiple groups from being load split across equal-cost paths. The following commands were introduced or modified: **ip multicast multipath**.

CHAPTER 48

Configuring Multicast Admission Control

This module describes how to implement multicast admission control in an IP multicast network. Multicast admission control features are configured on multicast-enabled routers to prevent control plane overload, ensure proper resource allocation, and provide multicast Call Admission Control (CAC) capabilities.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Prerequisites for Configuring Multicast Admission Control

IP multicast is enabled and the Protocol Independent Multicast (PIM) interfaces are configured using the tasks described in the “Configuring Basic IP Multicast” module.

Information About Configuring Multicast Admission Control

Multicast Admission Control

As the popularity of network video applications grows among consumers, admission control functions--which govern transmission and reception of multicast traffic based on available network resources--are vital. Without admission control, some users may receive degraded multicast streams, rendering programs unwatchable, and others may receive a “Network Busy” message or nothing at all as network resources are overtaxed. Network admission control is important in maintaining a high quality of experience for digital video consumers.

The goals of multicast admission control features, therefore, are as follows:

- Protect the router from control plane overload to ensure that memory and CPU resources on multicast-enabled routers are not overrun by multicast route (mroute) states or denial-of-service (DoS) attacks from multicast packets.
- Enable proper resource allocation (on a global, per MVRF, or per interface basis) to ensure that multicast services are delivered to subscribers per their IP Service Level Agreements (SLAs) and to minimize the effects of DoS attacks on subscribers.
- Provide multicast CAC capabilities to prevent bandwidth resources (interfaces, subnetworks) from being congested and to enable service providers to offer more flexible and refined content and subscriber-based policies.

Multicast Admission Control Features

The Cisco IOS software supports the following multicast admission control features:

- Global and Per MVRF Mroute State Limit

The **ip multicast route-limit** command allows for the configuration of global and per MVRF state limiters, which impose limits on the number of multicast routes (mroutes) that can be added to the global table or to a particular Multicast Virtual Routing and Forwarding (MVRF) table.

- IGMP State Limit

This feature allows for the configuration of IGMP state limiters, which impose limits on mroute states resulting from Internet Group Management Protocol (IGMP) membership reports (IGMP joins).

- Per Interface Mroute State Limit

This feature allows for the configuration of per interface mroute state limiters, which impose mroute state limits for different access control list (ACL)-classified sets of multicast traffic on an interface.

- Bandwidth-Based CAC for IP Multicast

This feature allows for the configuration of bandwidth-based multicast CAC policies, which allow for bandwidth-based CAC on a per interface basis.

These admission control features may be invoked by service providers and enterprise network administrators based on different criteria, including the service package an end user has purchased or the privileges an enterprise user is entitled to.

Global and Per MVRF Mroute State Limit

The **ip multicast route-limit** command allows for the configuration of global and per MVRF mroute state limiters, which impose limits on the number of mroutes that can be added to the global table or to a particular MVRF table, respectively.

Global mroute state limiters are used to limit the number of mroutes that can be added to the global table on a router. Configuring a global mroute state limiter can protect a router in the event of a multicast DoS attack (by preventing mroutes from overrunning the router).

Per VRF mroute state limiters are used to limit the number of mroutes that can be added to an MVRF table on a Multicast VPN (MVPN) provider edge (PE) router. Configuring per MVRF mroute state limits can be used to ensure the fair sharing of mroutes between different MVRFs on an MVPN PE router.

Global and Per MVRF Mroute State Limit Feature Design

Global and per MVRF mroute state limiters are configured using the **ip multicast route-limit** command in global configuration mode. The syntax of the **ip multicast route-limit** command is as follows:

ip multicast [**vrf** *vrf-name*] **route-limit** *limit* [*threshold*]

Issuing the **ip multicast route-limit**command without the optional **vrf** keyword and *vrf-name* arguments configures a global mroute state limiter. The optional **vrf** keyword and *vrf-name* arguments are used with the **ip multicast limit** command to configure per MVRF mroute state limiters.

Note When configuring global and per VRF mroute state limiters, you can only configure one limit for the global table and one limit per MVRF table.

The value specified for the required *limit* argument defines the maximum number of mroutes that can be added to either the global table or a particular MVRF table, respectively.

Note Global and per MVRF mroute state limiters operate independently and can be used alone or together, depending upon the admission control requirements of your network.

In addition, for both global and per MVRF mroute state limiters, the optional *threshold* argument is available to set mroute threshold limits.

Mechanics of Global and Per MVRF Mroute State Limiters

The mechanics of global and per MVRF mroute state limiters are as follows:

- Each time the state for an mroute is created on a router, the Cisco IOS software checks to see if the limit for the global mroute state limiter (if the mroute is associated with the global table) or the limit for the per MVRF mroute state limiter (if the mroute is associated with the MVRF table) has been reached.
- States for mroutes that exceed the configured limit for the global or the per MVRF mroute state limiter are not created on the router, and a warning message in the following format is generated:

```
% MROUTE-4-ROUTELIMIT : <current mroute count> exceeded multicast route-limit of
<mroute limit value>
```

- When an mroute threshold limit is also configured for the global or the per MVRF mroute state limiter, each time the state for an mroute is created on a router, the Cisco IOS software also checks to see if the mroute threshold limit has been reached. If the mroute threshold limit is exceeded, a warning message in the following format is generated:

```
% MROUTE-4-ROUTELIMITWARNING : multicast route-limit warning <current mroute count> threshold
 <mroute threshold value>
```

Warning messages continue to be generated until the number of mroutes exceeds the configured limit or until the number of mroute states falls below the configured mroute threshold limit.

MSDP SA Limit

The **ip msdp sa-limit** command allows for the configuration of MSDP SA limiters, which impose limits on the number of MSDP Source Active (SA) messages that an MSDP-enabled router can accept (can be cached) from an MSDP peer. This command provides a means to protect an MSDP-enabled router from denial of service (DoS) attacks.

MSDP SA Limit Feature Design

MSDP SA limiters are configured using the **ip msdp sa-limit** command in global configuration mode. The syntax of the **ip msdp sa-limit** command is as follows:

ip msdp [**vrf** *vrf-name*] **sa-limit** {*peer-address* | *peer-name*} *sa-limit*

For the required *peer-address* argument or *peer-name* argument, specify either the MSDP peer address or MSDP peer name of the peer to be limited.

For the required *sa-limit* argument, specify the maximum number of SA messages that can be accepted (cached) from the specified peer. The range is from 1 to 2147483646.

Note In an MVPN environment, the optional **vrf** keyword and *vrf-name* argument are used to specify the MVRF associated with the MSDP peer. When an MVRF is specified, the MSDP SA limiter is applied to the specified MSDP peer associated with the specified MVRF.

Mechanics of MSDP SA Limiters

- When MSDP SA limiters are configured, the router maintains a per-peer count of SA messages stored in the SA cache.
- SA messages that exceed the limit configured for an MSDP peer are ignored.
- If the router receives SA messages in excess of the configured limit from an MSDP peer, a warning in the following format is generated (once a minute) until the cache is cleared:

```
%MSDP-4-SA_LIMIT: SA from peer <peer address or name>, RP <RP address> for <mroute> exceeded
 sa-limit of <configured SA limit for MSDP peer>
```

Tips for Configuring MSDP SA Limiters

- We recommended that you configure MSDP SA limiters for all MSDP peerings on the router.

- An appropriately low MSDP SA limit should be configured on peerings with a stub MSDP region (an MSDP peer that may have some further downstream peers but does not act as a transit for SA messages across the rest of the Internet).
- An appropriately high SA limit should be configured for all MSDP peerings that act as transits for MSDP SA messages across the Internet.

IGMP State Limit

The IGMP State Limit feature allows for the configuration of IGMP state limiters, which impose limits on mroute states resulting from IGMP membership reports (IGMP joins) on a global or per interface basis. Membership reports exceeding the configured limits are not entered into the IGMP cache. This feature can be used to prevent DoS attacks or to provide a multicast CAC mechanism in network environments where all the multicast flows roughly utilize the same amount of bandwidth.

Note IGMP state limiters impose limits on the number of mroute states resulting from IGMP, IGMP v3lite, and URL Rendezvous Directory (URD) membership reports on a global or per interface basis.

IGMP State Limit Feature Design

- Configuring IGMP state limiters in global configuration mode specifies a global limit on the number of IGMP membership reports that can be cached.
- Configuring IGMP state limiters in interface configuration mode specifies a limit on the number of IGMP membership reports on a per interface basis.
- Use ACLs to prevent groups or channels from being counted against the interface limit. A standard or an extended ACL can be specified. A standard ACL can be used to define the (*, G) state to be excluded from the limit on an interface. An extended ACLs can be used to define the (S, G) state to be excluded from the limit on an interface. An extended ACL also can be used to define the (*, G) state to be excluded from the limit on an interface, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list.
- You can only configure one global limit per device and one limit per interface.

Mechanics of IGMP State Limiters

The mechanics of IGMP state limiters are as follows:

- Each time a router receives an IGMP membership report for a particular group or channel, the Cisco IOS software checks to see if either the limit for the global IGMP state limiter or the limit for the per interface IGMP state limiter has been reached.
- If only a global IGMP state limiter has been configured and the limit has not been reached, IGMP membership reports are honored. When the configured limit has been reached, subsequent IGMP membership reports are then ignored (dropped) and a warning message in one of the following formats is generated:
    -
    ```
    %IGMP-6-IGMP_GROUP_LIMIT: IGMP limit exceeded for <group (*, group address)> on
    <interface type number> by host <ip address>
    ```

-
  ```
  %IGMP-6-IGMP_CHANNEL_LIMIT: IGMP limit exceeded for <channel (source address, group
   address)> on <interface type number> by host <ip address>
  ```

- If only per interface IGMP state limiters are configured, then each limit is only counted against the interface on which it was configured.
- If both a global IGMP state limiter and per interface IGMP state limiters are configured, the limits configured for the per interface IGMP state limiters are still enforced but are constrained by the global limit.

Per Interface Mroute State Limit

The Per Interface Mroute State Limit feature provides the capability to limit the number of mroute states on an interface for different ACL-classified sets of multicast traffic. This feature can be used to prevent DoS attacks or to provide a multicast CAC mechanism when all the multicast flows roughly utilize the same amount of bandwidth.

The Per Interface Mroute State Limit feature essentially is a complete superset of the IGMP State Limit feature (with the exception that it does not support a global limit). The Per Interface Mroute State Limit feature, moreover, is more flexible and powerful (albeit more complex) than the IGMP State Limit feature but is not intended to be a replacement for it because there are applications that suit both features.

The main differences between the Per Interface Mroute State Limit feature and the IGMP State Limit feature are as follows:

- The Per Interface Mroute State Limit feature allows multiple limits to be configured on an interface, whereas the IGMP State Limit feature allows only one limit to be configured on an interface. The Per Interface Mroute State Limit feature, thus, is more flexible than the IGMP State Limit feature in that it allows multiple limits to be configured for different sets of multicast traffic on an interface.
- The Per Interface Mroute State Limit feature can be used to limit both IGMP and PIM joins, whereas the IGMP State Limit feature can only be used to limit IGMP joins. The IGMP State Limit feature, thus, is more limited in application in that it is best suited to be configured on an edge router to limit the number of groups that receivers can join on an outgoing interface. The Per Interface Mroute State Limit feature has a wider application in that it can be configured to limit IGMP joins on an outgoing interface, to limit PIM joins (for Any Source Multicast [ASM] groups or Source Specific Multicast [SSM] channels) on an outgoing interface connected to other routers, to limit sources behind an incoming interface from sending multicast traffic, or to limit sources directly connected to an incoming interface from sending multicast traffic.

Note Although the PIM Interface Mroute State Limit feature allows you to limit both IGMP and PIM joins, it does not provide the capability to limit PIM or IGMP joins separately because it does not take into account whether the state is created as a result of an IGMP or PIM join. As such, the IGMP State Limit feature is more specific in application because it specifically limits IGMP joins.

- The Per Interface Mroute State Limit feature allows you to specify limits according to the direction of traffic; that is, it allows you to specify limits for outgoing interfaces, incoming interfaces, and for incoming interfaces having directly connected multicast sources. The IGMP State Limit feature, however, only can be used to limit outgoing interfaces. The Per Interface State Mroute State Limit feature, thus, is wider in scope in that it can be used to limit mroute states for both incoming and outgoing interfaces from both sources and receivers, whereas the IGMP State Limit feature is more narrow in scope in that it can only be used to limit mroute states for receivers on an LAN by limiting the number of IGMP joins on an outgoing interface.

Both the IGMP State Limit and Per Interface Mroute State Limit features provide a rudimentary multicast CAC mechanism that can be used to provision bandwidth utilization on an interface when all multicast flows roughly utilize the same amount of bandwidth. The Bandwidth-Based CAC for IP Multicast feature, however, offers a more flexible and powerful alternative for providing multicast CAC in network environments where IP multicast flows utilize different amounts of bandwidth.

Note For more information about the Bandwidth-Based CAC for IP Multicast feature, see the Bandwidth-Based CAC for IP Multicast, on page 641.

Per Interface Mroute State Limit Feature Design

The Per Interface Mroute State Limit feature is configured using the **ip multicast limit** command in interface configuration mode. An **ip multicast limit** command configured on an interface is called an per interface mroute state limiter. A per interface mroute state limiter is defined by direction, ACL, and maximum number of mroutes. Each per interface mroute state limiter maintains a counter to ensure that the maximum number of mroutes is not exceeded.

The following forms of the **ip multicast limit**command are available to configure per interface mroute state limiters:

- **ip multicast limit** *access-list* *max-entries*

This command limits mroute state creation for an ACL-classified set of traffic on an interface when the interface is an outgoing (egress) interface, and limits mroute outgoing interface list (olist) membership when the interface is an incoming (ingress) Reverse Path Forwarding (RPF) interface.

This type of per interface mroute state limiter limits mroute state creation--by accounting each time an mroute permitted by the ACL is created or deleted--and limits mroute olist membership--by accounting each time that an mroute olist member permitted by the ACL is added or removed.

Entering this form of the command (that is, with no optional keywords) is equivalent to specifying the **ip multicast limit rpf**and **ip multicast limit out** forms of the command.

- **ip multicast limit** **connected** *access-list* *max-entries*

This command limits mroute state creation for an ACL-classified set of multicast traffic on an incoming (RPF) interface that is directly connected to a multicast source by accounting each time that an mroute permitted by the ACL is created or deleted.

- **ip multicast limit** **out** *access-list* *max-entries*

This command limits mroute olist membership on an outgoing interface for an ACL-classified set of multicast traffic by accounting each time that an mroute olist member permitted by the ACL is added or removed.

- **ip multicast limit** **rpf** *access-list* *max-entries*

This command limits mroute state creation for an ACL-classified set of multicast traffic on an incoming (RPF) interface by accounting each time an mroute permitted by the ACL is created or deleted.

For the required *access-list* argument, specify the ACL that defines the IP multicast traffic to be limited on an interface. A standard or extended ACL can be specified. Standard ACLs can be used to define the (*, G) state to be limited on an interface. Extended ACLs can be used to define the (S, G) state to be limited on an interface. Extended ACLs also can be used to define the (*, G) state to be limited on an interface, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list.

Mechanics of Per Interface Mroute State Limiters

The mechanics of per interface mroute state limiters are as follows:

- Each time the state for an mroute is created or deleted and each time an olist member is added or removed, the software searches for a corresponding per interface mroute state limiter that matches the mroute.
- When an mroute is created or deleted, the software searches for a per interface mroute state limiter configured on the incoming (RPF) interface that matches the mroute to be created or deleted. When an olist member is added or removed, the software searches for a per interface mroute state limiter configured on the outgoing interface that matches the mroute to be added or removed.
- A top-down search is performed using the list of configured per interface mroute state limiters. Only per interface mroute state limiters that match the direction of traffic are considered. The first per interface mroute state limiter that matches is used for limiting (sometimes referred to as accounting). A match is found when the ACL permits the mroute state.
- When a match is found, the counter of the per interface mroute state limiter is updated (increased or decreased). If no per interface mroute state limiter is found that matches an mroute, no accounting is performed for the mroute (because there is no counter to update).
- The amount with which to update the counter is called the cost (sometimes referred to as the cost multiplier). The default cost is 1.

Note A per interface mroute state limiter always allows the deletion of an mroute or the removal of an interface from the olist. In those cases, the respective per interface mroute state limiter decreases the counter by the value of the cost multiplier. In addition, RPF changes to an existing mroute are always allowed (in order to not affect existing traffic). However, a per interface mroute state limiter only allows the creation of an mroute or the addition of an mroute olist member if adding the cost does not exceed the maximum number of mroutes permitted.

Tips for Configuring Per Interface Mroute State Limiters

- To ensure that all mroutes are accounted, you can configure a per interface mroute state limiter whose ACL contains a permit-any statement and set the value of zero (0) for maximum entries. Configuring an mroute state limiter in this manner effectively denies all fall through states, which may be a way to prevent a multicast DoS attack in and out of the interface.
- When creating an ACL, remember that, by default, the end of the ACL contains an implicit deny-any statement for everything if it did not find a match before reaching the end.
- An explicit deny statement for a specific mroute in an ACL can be used to specify the state that will not match the ACLwhich will prevent the ACL from being accounted. If an mroute matches a deny statement, the search immediately continues to the next configured mroute state limiter. Configuring an explicit

deny statement in an ACL can be more efficient than forcing the mroute to fall through an ACL by using an implicit deny-any statement at the end of the ACL.

Bandwidth-Based CAC for IP Multicast

The Bandwidth-Based CAC for IP Multicast feature enhances the Per Interface Mroute State Limit feature by implementing a way to count per interface mroute state limiters using cost multipliers (referred to as *bandwidth-based multicast CAC policies*). This feature can be used to provide bandwidth-based multicast CAC on a per interface basis in network environments where the multicast flows utilize different amounts of bandwidth.

Bandwidth-Based CAC for IP Multicast Feature Design

Bandwidth-based multicast CAC policies are configured using the **ip multicast limit cost** command in global configuration mode. The syntax of the **ip multicast limit cost** command is as follows:

ip multicast [**vrf** *vrf-name*] **limit cost** *access-list cost-multiplier*

For the required *access-list* argument, specify the ACL that defines the IP multicast traffic for which to apply a cost. A standard or extended ACL can be specified. Standard ACLs can be used to define the (*, G) state. Extended ACLs can be used to define the (S, G) state. Extended ACLs also can be used to define the (*, G) state, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list.

For the required *cost-multiplier* argument, specify the cost value to be applied to mroutes that match the ACL associated with the bandwidth-based multicast CAC policy. The range is 0 to 2147483647.

Note In an MVPN environment, the optional **vrf** keyword and *vrf-name* argument are used to specify that the cost be applied only to mroutes associated with MVRF specified for the *vrf-name* argument.

Mechanics of the Bandwidth-Based Multicast CAC Policies

The mechanics of bandwidth-based multicast CAC policies are as follows:

- Once an mroute matches an ACL configured for a per interface mroute state limiter, the Cisco IOS software performs a top-down search from the global or per MVRF list of configured bandwidth-based multicast CAC policies to determine if a cost should be applied to the mroute.
- A cost is applied to the first bandwidth-based CAC policy that matches the mroute. A match is found when the ACL applied to the bandwidth-based CAC policy permits the mroute state.
- The counter for the mroute state limiter either adds or subtracts the cost configured for the *cost-multiplier* argument. If no costs are configured or if the mroute does not match any of the configured bandwidth-based CAC polices, the default cost of 1 is used.

Tips for Configuring Bandwidth-Based CAC Policies for IP Multicast

- To ensure that a particular cost applies to all mroutes being limited, you can configure a bandwidth-based CAC policy whose ACL contains a **permit any** statement. Configuring a bandwidth-based CAC policy in this manner effectively ensures that the default cost is not applied to any mroutes being limited.

- Configuring a bandwidth-based CAC policy with a cost of 0 for the *cost-multiplier* argument can be used to skip the accounting of certain mroutes (for example, to prevent Auto-RP groups or a specific multicast channel from being accounted).
- An explicit deny statement for a specific mroute in an ACL can be used to specify the state that will not match the ACL (thus, preventing the ACL from being accounted). If an mroute matches a deny statement, the search immediately continues to the next configured bandwidth-based CAC policy. Configuring an explicit deny statement in an ACL can be more efficient than forcing the mroute to fall through an ACL (by means of the implicit **deny any** statement at the end of the ACL).

How to Configure Multicast Admission Control

Configuring Global and Per MVRF Mroute State Limiters

Perform the following optional tasks to configure global and per MVRF mroute state limiters.

Global mroute state limiters are used to limit the number of mroutes that can be added to the global table on a router. Configuring a global mroute state limiter can protect a router in the event of a multicast DoS attack (by preventing mroutes from overrunning the router).

Per VRF mroute state limiters are used to limit the number of mroutes that can be added to an MVRF table on an MVPN PE router. Configuring per MVRF mroute state limits can be used to ensure the fair sharing of mroutes between different MVRFs on an MVPN PE router.

Note Global and per MVRF mroute state limiters operate independently and can be used alone or together, depending upon the admission control requirements of your network.

Note When configuring global and per VRF mroute state limiters, you can only configure one limit for the global table and one limit per MVRF table.

The following tasks explain how to configure global and per MVRF mroute state limiters:

Prerequisites

- These tasks assume that IP multicast has been enabled and that the PIM interfaces have been configured using the tasks described in the “ Configuring Basic IP Multicast ” module.
- Before configuring per MVRF mroute state limiters, the MVRFs on the PE router must be configured using the tasks described in the “ Configuring Multicast VPN ” module.

Configuring a Global Mroute State Limiter

Perform this task to limit the number of mroutes that can be added to the global table. States for mroutes that exceed the global mroute limit will not be created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast route-limit** *limit* [*threshold*]
4. **end**
5. **show ip mroute count**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast route-limit** *limit* [*threshold*] **Example:** `Router(config)# ip multicast route-limit 1500 1460`	Limits the number of mroutes that can be added to the global table. • For the required *limit* argument, specify the limit on the number of mroutes that can be added to the global table. The range is from 1 to 2147483647. • Use the optional *threshold*argument to set an mroute threshold limit. The range is from 1 to 2147483647.
Step 4	**end** **Example:** `Router(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 5	**show ip mroute count** **Example:** `Router# show ip mroute count`	(Optional) Displays mroute data and packet count statistics. • Use this command to verify the number of mroutes in the global table.

What to Do Next

Proceed to the Configuring Per MVRF Mroute State Limiters, on page 643 task to configure per MVRF mroute state limiters on a PE router.

Configuring Per MVRF Mroute State Limiters

Perform this optional task to configure per MVRF mroute state limiters to limit the number of mroutes that can be added to a particular MVRF table. This feature can be configured on a PE router to ensure the fair sharing of mroutes between different MVRFs on the router. States for mroutes that exceed the per MVRF mroute limiter are not created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast vrf** *vrf-name* **route-limit** *limit* [*threshold*]
4. Repeat Step 3 to configure additional per VRF mroute state limiters for other VRFs on an MVPN PE router.
5. **end**
6. **show ip mroute vrf** *vrf-name* **count**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast vrf** *vrf-name* **route-limit** *limit* [*threshold*] **Example:** `Router(config)# ip multicast vrf red route-limit 1500 1460`	Limits the number of mroutes that can be added to a particular MVRF table. • For the **vrf** keyword and *vrf-name* argument, specify the MVRF for which to apply the limit. • For the required *limit* argument, specify the limit on the number of mroutes that can be added to the MVRF table (for the specified MVRF). The range is from 1 to 2147483647. • Use the optional *threshold*argument to set an mroute threshold limit. The range is from 1 to 2147483647
Step 4	Repeat Step 3 to configure additional per VRF mroute state limiters for other VRFs on an MVPN PE router.	--
Step 5	**end** **Example:** `Router(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 6	**show ip mroute vrf** *vrf-name* **count** **Example:**	(Optional) Displays mroute data and packet count statistics related to the specified MVRF.

	Command or Action	Purpose
	`Router# show ip mroute vrf red count`	• Use this command to verify the number of mroutes in a particular MVRF table.

Configuring MSDP SA Limiters

Perform this optional task to limit the overall number of SA messages that the router can accept from specified MSDP peers. Performing this task protects an MSDP-enabled router from distributed DoS attacks.

Note We recommend that you perform this task for all MSDP peerings on the router.

Before you begin

This task assumes that you are running MSDP and have configured MSDP peers using the tasks described in the " Using MSDP to Interconnect Multiple PIM-SM Domains " module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp** [**vrf** *vrf-name*] **sa-limit** {*peer-address* | *peer-name*} *sa-limit*
4. Repeat Step 3 to configure additional per MVRF mroute state limiters for other MVRFs on an MVPN PE router.
5. **end**
6. **show ip msdp count**
7. **show ip msdp peer** [*peer-address* | *peer-name*]
8. **show ip msdp summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp** [**vrf** *vrf-name*] **sa-limit** {*peer-address* | *peer-name*} *sa-limit* **Example:**	Limits the number of SA messages allowed in the SA cache from the specified MSDP. • Use the optional **vrf** keyword and *vrf-name* argument to specify the MVRF associated with the MSDP peer.

	Command or Action	Purpose
	`Router(config)# ip msdp sa-limit 192.168.10.1 100`	When an MVRF is specified, the MSDP SA limiter is applied to the specified MSDP peer associated with the specified MVRF. • For the required *peer-address* argument or *peer-name* argument, specify either the MSDP peer address or MSDP peer name of the peer to be limited. • For the required *sa-limit* argument, specify the maximum number of SA messages that can be accepted (cached) from the specified peer. The range is from 1 to 2147483646.
Step 4	Repeat Step 3 to configure additional per MVRF mroute state limiters for other MVRFs on an MVPN PE router.	--
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip msdp count** **Example:** `Router# show ip msdp count`	(Optional) Displays the number of sources and groups originated in MSDP SA messages and the number of SA messages from an MSDP peer in the SA cache.
Step 7	**show ip msdp peer** [*peer-address* | *peer-name*] **Example:** `Router# show ip msdp peer`	(Optional) Displays detailed information about MSDP peers. **Note** The output of this command displays the number of SA messages received from MSDP peers that are stored in the cache.
Step 8	**show ip msdp summary** **Example:** `Router# show ip msdp summary`	(Optional) Displays MSDP peer status. **Note** The output of this command displays a per-peer "SA Count" field that displays the number of SAs stored in the SA cache.

Configuring IGMP State Limiters

Note IGMP state limiters impose limits on the number of mroute states resulting from IGMP, IGMP v3lite, and URD membership reports on a global or per interface basis.

Prerequisites

- These tasks assume that IP multicast has been enabled and that the PIM interfaces have been configured using the tasks described in the " Configuring Basic IP Multicast " module.

- All ACLs you intend to apply to per interface IGMP state limiters should be configured prior to beginning this configuration task; otherwise, IGMP membership reports for all groups and channels are counted against the configured limits. For information about how to configure ACLs, see the " Creating an IP Access List and Applying It to an Interface " module.

Configuring Global IGMP State Limiters

Perform this optional task to configure one global IGMP state limiter per device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp limit** *number*
4. **end**
5. **show ip igmp groups**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp limit** *number* **Example:** `Device(config)# ip igmp limit 150`	Configures a global limit on the number of mroute states resulting from IGMP membership reports (IGMP joins).
Step 4	**end** **Example:** `Device(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 5	**show ip igmp groups** **Example:** `Device# show ip igmp groups`	(Optional) Displays the multicast groups with receivers that are directly connected to the device and that were learned through IGMP.

What to Do Next

Proceed to the Configuring Per Interface IGMP State Limiters, on page 648 task to configure per interface IGMP state limiters.

Configuring Per Interface IGMP State Limiters

Perform this optional task to configure a per interface IGMP state limiter.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp limit** *number* [**except** *access-list*]
5. Do one of the following:
   - **exit**
   - **end**
6. **show ip igmp interface** [*type number*]
7. **show ip igmp groups**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0`	Enters interface configuration mode. • Specify an interface that is connected to hosts.
Step 4	**ip igmp limit** *number* [**except** *access-list*] **Example:** `Device(config-if)# ip igmp limit 100`	Configures a per interface limit on the number of mroutes states created as a result of IGMP membership reports (IGMP joins).
Step 5	Do one of the following: • **exit** • **end**	• (Optional) Ends the current configuration session and returns to global configuration mode. Repeat steps 3 and 4 to configure a per interface limiter on another interface.

	Command or Action	Purpose

	Example:
	`Device(config-if)# exit`
	`Device(config-if)# end`
		• Ends the current configuration session and returns to privileged EXEC mode.

Step 6	**show ip igmp interface** [*type number*] **Example:** `Device# show ip igmp interface`	(Optional) Displays information about the status and configuration of IGMP and multicast routing on interfaces.
Step 7	**show ip igmp groups** **Example:** `Device# show ip igmp groups`	(Optional) Displays the multicast groups with receivers that are directly connected to the device and that were learned through IGMP.

Configuring Per Interface Mroute State Limiters

Perform this task to prevent DoS attacks or to provide a multicast CAC mechanism for controling bandwidth when all multicast flows utilize approximately the same amount of bandwidth.

Before you begin

All ACLs to be applied to per interface mroute state limiters must be configured prior to beginning this configuration task; otherwise, the limiters are ignored. For information about how to configure ACLs, see the " Creating an IP Access List and Applying It to an Interface " module of the *Security Configuration Guide: Access Control Lists* guide.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip multicast limit** [**connected** | **out** | **rpf**] *access-list max-entries*
5. Repeat Step 4 to configure additional per interface mroute state limiters on this interface.
6. Repeat Steps 3 and Step 4 to configure per interface mroute state limiters on additional interfaces.
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0`	Enters interface configuration mode for the specified interface type and number.
Step 4	**ip multicast limit** [**connected** | **out** | **rpf**] *access-list max-entries* **Example:** `Device(config-if)# ip multicast limit 15 100`	Configures per interface mroute state limiters.
Step 5	Repeat Step 4 to configure additional per interface mroute state limiters on this interface.	--
Step 6	Repeat Steps 3 and Step 4 to configure per interface mroute state limiters on additional interfaces.	--
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

What to Do Next

Proceed to the Monitoring Per Interface Mroute State Limiters and Bandwidth-Based Multicast CAC Policies task to monitor per interface mroute state limiters.

Configuring Bandwidth-Based Multicast CAC Policies

Perform this optional task to configure bandwidth-based multicast CAC policies. Bandwidth-based multicast CAC policies provide the capability to assign costs to mroutes that are being limited by per interface mroute state limiters. This task can be used to provide bandwidth-based multicast CAC on a per interface basis in network environments where the multicast flows utilize different amounts of bandwidth. Bandwidth-based multicast CAC policies can be applied globally or per MVRF.

Before you begin

- This task assumes that IP multicast has been enabled and that the PIM interfaces have been configured using the tasks described in the “ Configuring Basic IP Multicast ” module.
- All ACLs you intend to apply to bandwidth-based multicast CAC policies should be configured prior to beginning this configuration task; otherwise, the limiters are ignored. For information about how to configure ACLs, see the “ Creating an IP Access List and Applying It to an Interface ” module.

> **Note** You can omit Steps 3 to 7 if you have already configured the per interface mroute state limiters for which to apply costs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip multicast limit** [**connected** | **out** | **rpf**] *access-list max-entries*
5.
6.
7. Repeat Step 4 if you want to configure additional mroute state limiters on the interface.
8. Repeat Step 3 and Step 4 if you want to configure mroute state limiters on additional interfaces.
9. **exit**
10. **ip multicast** [**vrf** *vrf-name*] **limit cost** *access-list cost-multiplier*
11. Repeat Step 8 if you want to apply additional costs to mroutes.
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0`	Enters interface configuration mode for the specified interface type and number.
Step 4	**ip multicast limit** [**connected** | **out** | **rpf**] *access-list max-entries* **Example:** `Router(config-if)# ip multicast limit acl-test 100` **Example:**	Configures mroute state limiters on an interface. • Specify the **ip multicast limit** command with no optional keywords to limit mroute state creation for an ACL-classified set of traffic on an interface when the interface is an outgoing (egress) interface, and to limit mroute olist membership when the interface is an incoming (ingress) Reverse Path Forwarding (RPF) interface.

	Command or Action	Purpose
Step 5		• • This type of mroute state limiter limits mroute state creation by accounting each time an mroute permitted by the ACL is created or deleted and limits mroute olist membership by accounting each time that an mroute olist member permitted by the ACL is added or removed. • Entering this form of the command (that is, with no optional keywords) is equivalent to specifying the **ip multicast limit rpf**and **ip multicast limit out** forms of the command.

Step 6		• Use the optional **connected** keyword to configure an mroute state limiter that limits mroute states created for an ACL-classified set of multicast traffic on an incoming (RPF) interface that is directly connected to a multicast source by accounting each time that an mroute permitted by the ACL is created or deleted.
		• Use the optional **out** keyword to configure an mroute state limiter that limits mroute olist membership on an outgoing interface for an ACL-classified set of multicast traffic by accounting each time that an mroute olist member permitted by the ACL is added or removed.
		• Use the optional **rpf** keyword to configure an mroute state limiter that limits the number of mroute states created for an ACL-classified set of multicast traffic on an incoming (RPF) interface by accounting each time an mroute permitted by the ACL is created or deleted.
		• For the required *access-list* argument, specify the ACL that defines the IP multicast traffic to be limited on an interface.
		  • Standard ACLs can be used to define the (*, G) state to be limited on an interface.
		  • Extended ACLs can be used to define the (S, G) state to be limited on an interface. Extended ACLs also can be used to define the (*, G) state to be limited on an interface, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list.
		• For the required *max-entries* argument, specify the maximum number of mroutes permitted by the per interface mroute state limiter. The range is from 0 to 2147483647.

	Command or Action	Purpose
Step 7	Repeat Step 4 if you want to configure additional mroute state limiters on the interface.	--
Step 8	Repeat Step 3 and Step 4 if you want to configure mroute state limiters on additional interfaces.	--
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode, and returns to global configuration mode.

Step 10	**ip multicast** [**vrf** *vrf-name*] **limit cost** *access-list cost-multiplier*
	Example:
	`Router(config)# ip multicast limit cost acl-MP2SD-channels 4000`
		Applies costs to per interface mroute state limiters.
		• Use the optional **vrf** keyword and *vrf-name*argument to specify that the cost be applied only to mroutes associated with MVRF specified for the *vrf-name* argument.
		• For the required *access-list* argument, specify the ACL that defines the IP multicast traffic for which to apply a cost.
		  • Standard ACLs can be used to define the (*, G) state.
		  • Extended ACLs can be used to define the (S, G) state. Extended ACLs also can be used to define the (*, G) state, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list.
		• For the required *cost-multiplier* argument, specify the cost value to be applied to mroutes that match the ACL associated with the bandwidth-based multicast CAC policy. The range is 0 to 2147483647.

Step 11	Repeat Step 8 if you want to apply additional costs to mroutes.	--
Step 12	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode, and enters privileged EXEC mode.

What to Do Next

Proceed to the Monitoring Per Interface Mroute State Limiters and Bandwidth-Based Multicast CAC Policies task to monitor bandwidth-based multicast CAC policies.

Monitoring Per Interface Mroute State Limiters and Bandwidth-Based Multicast CAC Policies

Perform this optional task to monitor per interface mroute state limiters and bandwidth-based multicast CAC policies.

SUMMARY STEPS

1. **enable**
2. **debug ip mrouting limits** [*group-address*]
3. **show ip multicast limit** *type* *number*
4. **clear ip multicast limit** [*type number*]

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **debug ip mrouting limits** [*group-address*]

Displays debugging information about configured per interface mroute state limiters and bandwidth-based multicast CAC policies.

The following output is from the **debug ip mrouting limits** command. The output displays the following events:

- An mroute state being created and the corresponding per interface mroute state limiter counter being increased by the default cost of 1 on incoming Ethernet interface 1/0.
- An mroute olist member being removed from the olist and the corresponding per interface mroute limiter being decreased by the default cost of 1 on outgoing Ethernet interface 1/0.
- An mroute being denied by the per interface mroute state limiter because the maximum number of mroute states has been reached.
- An mroute state being created and the corresponding per interface mroute state limiter counter being increased by the cost of 2 on incoming Ethernet interface 1/0.
- An mroute olist member being removed from the olist and the corresponding per interface mroute limiter being decreased by a cost of 2 on outgoing Ethernet interface 1/0.

Example:

```
device# debug ip mrouting limits

MRL(0): incr-ed acl ‘rpf-list’ to (13 < max 32), [n:0,p:0], (main) GigabitEthernet0/0, (10.41.0.41,
 225.30.200.60)
MRL(0): decr-ed acl ‘out-list’ to (10 < max 32), [n:0,p:0], (main) GigabitEthernet0/0, (*,
```

```
225.40.202.60)
MRL(0): Add mroute (10.43.0.43, 225.30.200.60) denied for GigabitEthernet0/2, acl std-list, (16 =
max 16)
MRL(0): incr-ed limit-acl `rpf-list' to (12 < max 32), cost-acl 'cost-list' cost 2, [n:0,p:0], (main)
 GigabitEthernet0/0, (10.41.0.41, 225.30.200.60)
MRL(0): decr-ed limit-acl `out-list' to (8 < max 32), cost-acl 'cost-list'' cost 2, [n:0,p:0], (main)
 GigabitEthernet0/0, (*, 225.40.202.60)
```

Step 3 **show ip multicast limit** *type* *number*

Displays counters related to mroute state limiters configured on the interfaces on the router.

For each per interface mroute state limiter shown in the output, the following information is displayed:

- The direction of traffic that the per mroute state limiter is limiting.
- The ACL referenced by the per interface mroute state limiter that defines the IP multicast traffic being limited.
- Statistics, enclosed in parenthesis, which track the current number of mroutes being limited less the configured limit. Each time the state for an mroute is created or deleted and each time an outgoing interface list (olist) member is added or removed, the counters for matching per interface mroute state limiters are increased or decreased accordingly.
- The exceeded counter, which tracks the total number of times that the limit configured for the per interface mroute state limiter has been exceeded. Each time an mroute is denied due to the configured limit being reached, the exceeded counter is increased by a value of 1.

The following is sample output from the **show ip multicast limit** command with the *type number* arguments. In this example, information about mroute state limiters configured on Gigabit Ethernet interface 0/0 is displayed.

Example:

```
Device# show ip multicast limit GigabitEthernet 0/0

Interface GigabitEthernet 0/0
 Multicast Access Limits
out acl out-list (1 < max 32) exceeded 0
rpf acl rpf-list (6 < max 32) exceeded 0
con acl conn-list (0 < max 32) exceeded 0
```

Step 4 **clear ip multicast limit** [*type number*]

Resets the exceeded counter for per interface mroute state limiters.

The following example shows how to reset exceeded counters for per interface mroute state limiters configured on Gigabit Ethernet interface 0/0:

Example:

```
Device# clear ip multicast limit interface GigabitEthernet 0/0
```

Configuration Examples for Configuring Multicast Admission Control

Configuring Global and Per MVRF Mroute State Limiters Example

The following example shows how to configure a global mroute state limiter. In this example, a global mroute state limiter is configured with an mroute limit of 1500 and an mroute threshold limit of 1460.

```
ip multicast route-limit 1500 1460
```

The following is a sample mroute threshold warning message. The output shows that the configured mroute threshold limit of 1460 has been exceeded by one mroute.

```
%MROUTE-4-ROUTELIMITWARNING : multicast route-limit warning 1461 threshold 1460
```

The following is a sample mroute exceeded warning message. The output shows that the configured mroute limit of 1500 has been exceeded by one mroute. States for mroutes that exceed the configured limit for the global mroute state limiter are not created on the router.

```
%MROUTE-4-ROUTELIMIT : 1501 routes exceeded multicast route-limit of 1500
```

Configuring MSDP SA Limiters Example

The following example shows how to configure an MSDP SA limiter. In this example, an MSDP SA limiter is configured that imposes a limit of 100 SA messages from the MSDP peer at 192.168.10.1.

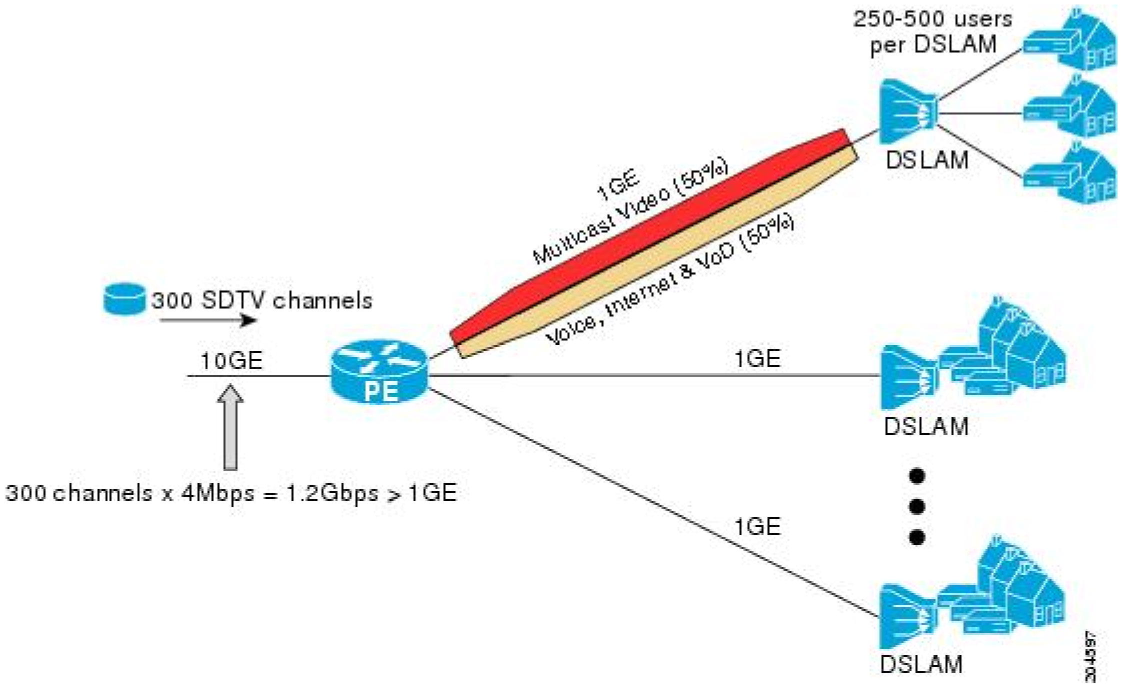
```
ip msdp sa-limit 192.168.10.1 100
```

Example: Configuring IGMP State Limiters

The following example shows how to configure IGMP state limiters to provide multicast CAC in a network environment where all the multicast flows roughly utilize the same amount of bandwidth.

This example uses the topology illustrated in the figure.

Figure 71: IGMP State Limit Example Topology



In this example, a service provider is offering 300 Standard Definition (SD) TV channels. Each SD channel utilizes approximately 4 Mbps.

The service provider must provision the Gigabit Ethernet interfaces on the PE device connected to the Digital Subscriber Line Access Multiplexers (DSLAMs) as follows: 50% of the link's bandwidth (500 Mbps) must be available to subscribers of the Internet, voice, and video on demand (VoD) service offerings while the remaining 50% (500 Mbps) of the link's bandwidth must be available to subscribers of the SD channel offerings.

Because each SD channel utilizes the same amount of bandwidth (4 Mbps), per interface IGMP state limiters can be used to provide the necessary CAC to provision the services being offered by the service provider. To determine the required CAC needed per interface, the total number of channels is divided by 4 (because each channel utilizes 4 Mbps of bandwidth). The required CAC needed per interface, therefore, is as follows:

500Mbps / 4Mbps = 125 mroutes

Once the required CAC is determined, the service provider uses the results to configure the per IGMP state limiters required to provision the Gigabit Ethernet interfaces on the PE device. Based on the network's CAC requirements, the service provider must limit the SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 125. Configuring a per interface IGMP state limit of 125 for the SD channels provisions the interface for 500 Mbps of bandwidth, the 50% of the link's bandwidth that must always be available (but never exceeded) for the SD channel offerings.

The following configuration shows how the service provider uses a per interface mroute state limiter to provision interface Gigabit Ethernet 0/0 for the SD channels and Internet, Voice, and VoD services being offered to subscribers:

```
interface GigabitEthernet0/0
description --- Interface towards the DSLAM ---
.
.
```

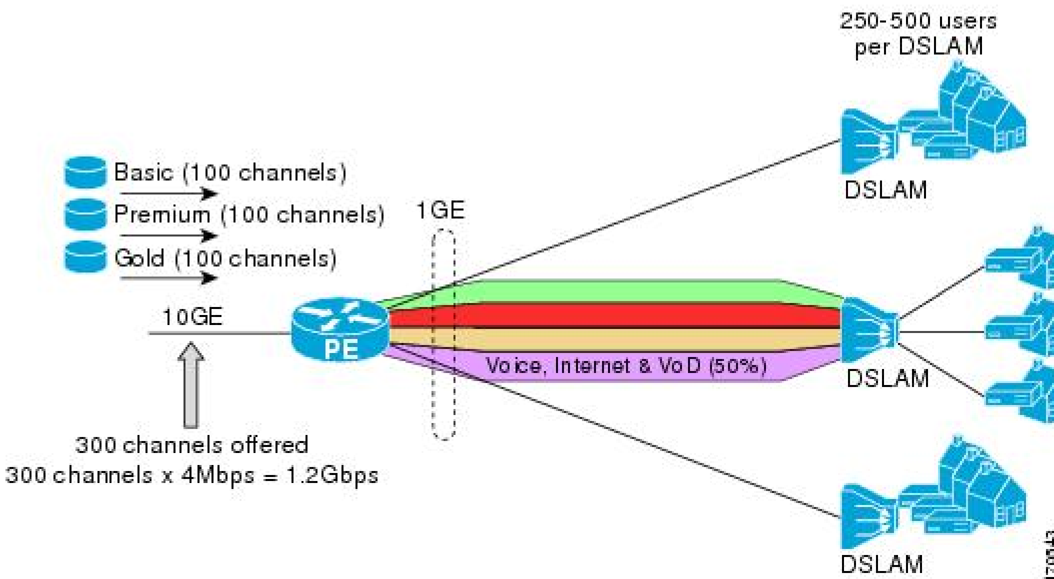
```
.
ip igmp limit 125
```

Example Configuring Per Interface Mroute State Limiters

The following example shows how to configure per interface mroute state limiters to provide multicast CAC in a network environment where all the multicast flows roughly utilize the same amount of bandwidth.

This example uses the topology illustrated in the figure.

Figure 72: Per Interface Mroute State Limit Example Topology



In this example, a service provider is offering 300 SD TV channels. The SD channels are being offered to customers in three service bundles (Basic, Premium, and Gold), which are available to customers on a subscription basis. Each bundle offers 100 channels to subscribers, and each channel utilizes approximately 4 Mbps of bandwidth.

The service provider must provision the Gigabit Ethernet interfaces on the PE device connected to DSLAMs as follows: 50% of the link's bandwidth (500 Mbps) must be available to subscribers of their Internet, voice, and VoD service offerings while the remaining 50% (500 Mbps) of the link's bandwidth must be available to subscribers of their SD channel bundle service offerings.

For the 500 Mbps of the link's bandwidth that must always be available to (but must never be exceeded by) the subscribers of the SD channel bundles, the interface must be further provisioned as follows:

- 60% of the bandwidth must be available to subscribers of the basic service (300 Mbps).
- 20% of the bandwidth must be available to subscribers of the premium service (100 Mbps).
- 20% of the bandwidth must be available to subscribers of the gold service (100 Mbps).

Because each SD channel utilizes the same amount of bandwidth (4 Mbps), per interface mroute state limiters can be used to provide the necessary CAC to provision the services being offered by the service provider. To determine the required CAC needed per interface, the number of channels for each bundle is divided by 4 (because each channel utilizes 4 Mbps of bandwidth). The required CAC needed per interface, therefore, is as follows:

- Basic Services: 300 / 4 = 75
- Premium Services: 100 / 4 = 25
- Gold Services: 100 / 4 = 25

Once the required CAC required per SD channel bundle is determined, the service provider uses the results to configure the mroute state limiters required to provision the Gigabit Ethernet interfaces on the PE device for the services being offered to subscribers behind the DSLAMs:

- For the Basic Services bundle, the service provider must limit the number of Basic Service SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 75. Configuring an mroute state limit of 75 for the SD channels offered in the Basic Service bundle provisions the interface for 300 Mbps of bandwidth (the 60% of the link's bandwidth that must always be available to [but never exceeded by] the subscribers of the Basic Services bundle).
- For the Premium Services bundle, the service provider must limit the number of Premium Service SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 25. Configuring an mroute state limit of 25 for the SD channels offered in the Premium Service bundle provisions the interface for 100 Mbps of bandwidth (the 20% of the link's bandwidth that must always be available to [but never exceeded by] the subscribers of the Premium Service bundle).
- For the Gold Services bundle, the service provider must limit the number of Gold Service SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 25. Configuring an mroute state limit of 25 for the SD channels offered in the Gold Service bundle provisions the interface for 100 Mbps of bandwidth (the 20% of the link's bandwidth that must always be available to [but never exceeded by] the subscribers of the Gold Service bundle).

The service provider then configures three ACLs to be applied to per interface mroute state limiters. Each ACL defines the SD channels for each SD channel bundle to be limited on an interface:

- acl-basic--The ACL that defines the SD channels offered in the basic service.
- acl-premium--The ACL that defines the SD channels offered in the premium service.
- acl-gold--The ACL that defines the SD channels offered in the gold service.

These ACLs are then applied to per interface mroute state limiters configured on the PE device's Gigabit Ethernet interfaces.

For this example, three per interface mroute state limiters are configured on Gigabit Ethernet interface 0/0 to provide the multicast CAC needed to provision the interface for the SD channel bundles being offered to subscribers:

- An mroute state limit of 75 for the SD channels that match acl-basic.
- An mroute state limit of 25 for the SD channels that match acl-premium.
- An mroute state limit of 25 for the SD channels that match acl-gold.

The following configuration shows how the service provider uses per interface mroute state limiters to provision Gigabit Ethernet interface 0/0 for the SD channel bundles and Internet, Voice, and VoD services being offered to subscribers:

```
interface GigabitEthernet0/0
description --- Interface towards the DSLAM ---
.
```

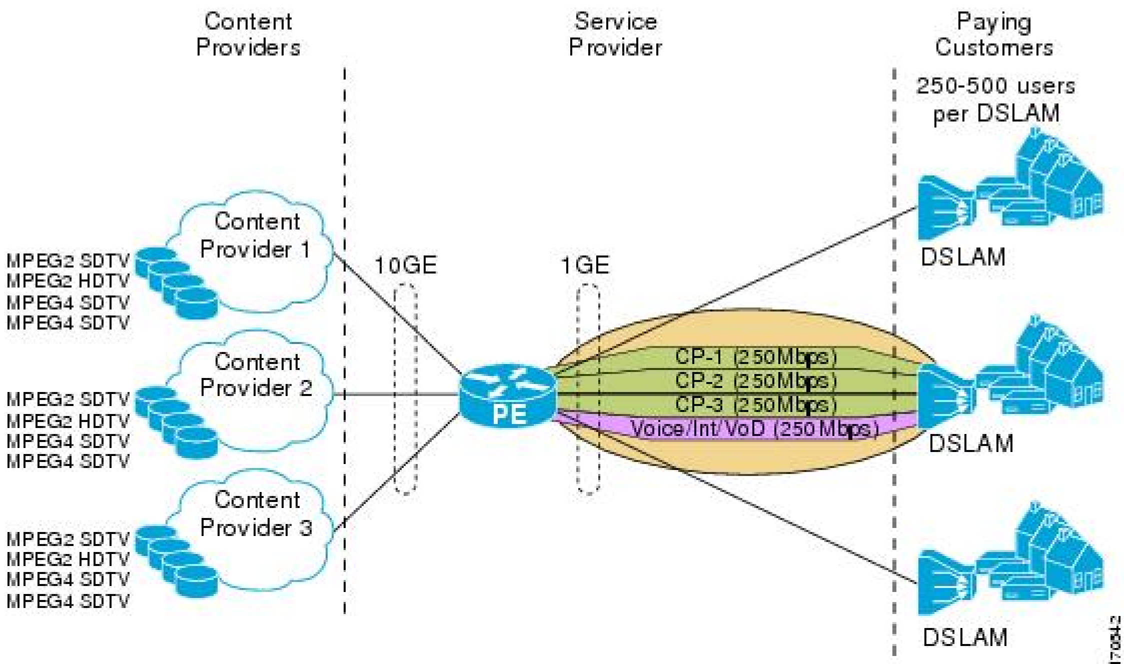
```
.
.
ip multicast limit out acl-basic 75
ip multicast limit out acl-premium 25
ip multicast limit out acl-gold 25
```

Example: Configuring Bandwidth-Based Multicast CAC Policies

The following example shows how to configure bandwidth-based multicast CAC policies to provide multicast CAC in a network environment where the multicast flows utilize different amounts of bandwidth.

This example uses the topology illustrated in the figure.

Figure 73: Bandwidth-Based CAC for IP Multicast Example Topology



In this example, three content providers are providing TV services across a service provider core. The content providers are broadcasting TV channels that utilize different amounts of bandwidth:

- MPEG-2 SDTV channels--4 Mbps per channel.
- MPEG-2 HDTV channels--18 Mbps per channel.
- MPEG-4 SDTV channels--1.6 Mbps per channel.
- MPEG-4 HDTV channels--6 Mbps per channel.

The service provider needs to provision the fair sharing of bandwidth between these three content providers to its subscribers across Gigabit Ethernet interfaces. The service provider, thus, determines that it needs to provision each Gigabit Ethernet interface on the PE device connected to the DSLAMs as follows:

- 250 Mbps per content provider.
- 250 Mbps for Internet, voice, and VoD services.

The service provider then configures three ACLs:

- acl-CP1-channels--The ACL that defines the channels being offered by the content provider CP1.
- acl-CP2-channels--The ACL that defines the channels being offered by the content provider CP2.
- acl-CP3-channels--The ACL that defines the channels being offered by the content provider CP3.

Because the content providers are broadcasting TV channels that utilize different amounts of bandwidth, the service provider needs to determine the values that need to be configured for the per interface mroute state limiters and bandwidth-based multicast CAC policies to provide the fair sharing of bandwidth required between the content providers.

Prior to the introduction of the Bandwidth-Based CAC for IP Multicast feature, per interface mroute state limiters were based strictly on the number of flows. The introduction of cost multipliers by the Bandwidth-Based CAC for IP Multicast feature expands how per interface mroute state limiters can be defined. Instead of defining the per interface mroute state limiters based on the number of multicast flows, the service provider looks for a common unit of measure and decides to represent the per interface mroute state limiters in kilobits per second (Kbps). The service provider then configures three per interface mroute state limiters, one per content provider. Because the link is a Gigabit, the service provider sets each limit to 250000 (because 250000 Kbps equals 250 Mbps, the number of bits that service provider needs to provision per content provider).

The service provider needs to further provision the fair sharing of bandwidth between the content providers, which can be achieved by configuring bandwidth-based multicast CAC policies. The service provider decides to create four bandwidth-based CAC policies, one policy per channel based on bandwidth. For these policies, the service provider configures the following ACLs:

- acl-MP2SD-channels--Defines all the MPEG-2 SD channels offered by the three content providers.
- acl-MP2HD-channels--Defines all the MPEG-2 HD channels offered by the three content providers.
- acl-MP4SD-channels--Defines all the MPEG-4 SD channels offered by the three content providers.
- acl-MP4HD-channels--Defines all the MPEG-4 HD channels offered by the three content providers.

For each policy, a cost multiplier (represented in Kbps) is defined for each ACL that is based on the bandwidth of the channels defined in the ACL:

- 4000--Represents the 4 Mbps MPEG-2 SD channels.
- 18000--Represents the 18 Mbps MPEG-2 HD channels.
- 1600--Represents the 1.6 Mbps MPEG-4 SD channels.
- 6000--Represents the 6 Mbps MPEG-4 HD channels.

The following configuration example shows how the service provider used per interface mroute state limiters with bandwidth-based multicast CAC policies to provision Gigabit Ethernet interface 0/0 for the fair sharing of bandwidth required between the three content providers:

```
!
ip multicast limit cost acl-MP2SD-channels 4000
ip multicast limit cost acl-MP2HD-channels 18000
ip multicast limit cost acl-MP4SD-channels 1600
ip multicast limit cost acl-MP4HD-channels 6000
!
.
.
```

```
.
!
interface GigabitEthernet0/0
 ip multicast limit out acl-CP1-channels 250000
 ip multicast limit out acl-CP2-channels 250000
 ip multicast limit out acl-CP3-channels 250000
!
```

Additional References

The following sections provide references related to configuring multicast admission control.

Related Documents

Related Topic	Document Title
Overview of the IP multicast technology area	" IP Multicast Technology Overview " module
Concepts, tasks, and examples for configuring an IP multicast network using PIM	" Configuring a Basic IP Multicast Network " module
Concepts, tasks, and examples for using MSDP to interconnection multiple PIM-SM domains	" Using MSDP to Interconnect Multiple PIM-SM Domains " module
Multicast commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring Multicast Admission Control

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 39: Feature Information for Configuring Multicast Admission Control

Feature Name	Releases	Feature Information
Bandwidth-Based CAC for IP Multicast	12.2(33)SRB 12.4(15)T 12.2(33)SXI	The Bandwidth-Based CAC for IP Multicast feature enhances the Per Interface Mroute State Limit feature by implementing a way to count per interface mroute state limiters using cost multipliers. This feature can be used to provide bandwidth-based CAC on a per interface basis in network environments where the multicast flows utilize different amounts of bandwidth. The following command was introduced by this feature: **ip multicast limit cost**.
IGMP State Limit	12.2(14)S 12.2(15)T 15.0(1)S	The IGMP State Limit feature introduces the capability to limit the number of mroute states resulting from IGMP membership states on a per interface or global basis. Membership reports exceeding the configured limits are not entered into the IGMP cache. This feature can be used to prevent DoS attacks or to provide a multicast CAC mechanism in network environments where all the multicast flows roughly utilize the same amount of bandwidth. The following commands were introduced or modified by this feature: **ip igmp limit**(global), **ip igmp limit**(interface), **show ip igmp interface**.

Feature Name	Releases	Feature Information
Per Interface Mroute State Limit	12.3(14)T 12.2(33)SRB 12.2(33)SXI	The Per Interface Mroute State Limit feature provides the capability to limit the number of mroute states on an interface for different ACL-classified sets of multicast traffic. This feature can be used to prevent DoS attacks, or to provide a multicast CAC mechanism in network environments where all the multicast flows roughly utilize the same amount of bandwidth. The following commands were introduced or modified by this feature: **clear ip multicast limit**, **debug ip mrouting limits**, **ip multicast limit**, **show ip multicast limit**.

CHAPTER 49

Per Interface Mroute State Limit

The Per Interface Mroute State Limit feature provides the capability to limit the number of mroute states on an interface for different ACL-classified sets of multicast traffic. This feature can be used to prevent DoS attacks or to provide a multicast CAC mechanism when all the multicast flows roughly utilize the same amount of bandwidth.

The Per Interface Mroute State Limit feature essentially is a complete superset of the IGMP State Limit feature (with the exception that it does not support a global limit). The Per Interface Mroute State Limit feature, moreover, is more flexible and powerful (albeit more complex) than the IGMP State Limit feature but is not intended to be a replacement for it because there are applications that suit both features.

The main differences between the Per Interface Mroute State Limit feature and the IGMP State Limit feature are as follows:

- The Per Interface Mroute State Limit feature allows multiple limits to be configured on an interface, whereas the IGMP State Limit feature allows only one limit to be configured on an interface. The Per Interface Mroute State Limit feature, thus, is more flexible than the IGMP State Limit feature in that it allows multiple limits to be configured for different sets of multicast traffic on an interface.
- The Per Interface Mroute State Limit feature can be used to limit both IGMP and PIM joins, whereas the IGMP State Limit feature can only be used to limit IGMP joins. The IGMP State Limit feature, thus, is more limited in application in that it is best suited to be configured on an edge router to limit the number of groups that receivers can join on an outgoing interface. The Per Interface Mroute State Limit feature has a wider application in that it can be configured to limit IGMP joins on an outgoing interface, to limit PIM joins (for Any Source Multicast [ASM] groups or Source Specific Multicast [SSM] channels) on an outgoing interface connected to other routers, to limit sources behind an incoming interface from sending multicast traffic, or to limit sources directly connected to an incoming interface from sending multicast traffic.

Note Although the PIM Interface Mroute State Limit feature allows you to limit both IGMP and PIM joins, it does not provide the capability to limit PIM or IGMP joins separately because it does not take into account whether the state is created as a result of an IGMP or PIM join. As such, the IGMP State Limit feature is more specific in application because it specifically limits IGMP joins.

- The Per Interface Mroute State Limit feature allows you to specify limits according to the direction of traffic; that is, it allows you to specify limits for outgoing interfaces, incoming interfaces, and for incoming interfaces having directly connected multicast sources. The IGMP State Limit feature, however, only can be used to limit outgoing interfaces. The Per Interface State Mroute State Limit feature, thus, is wider in scope in that it can be used to limit mroute states for both incoming and outgoing interfaces from both sources and receivers, whereas the IGMP State Limit feature is more narrow in scope in that it can only be used to limit mroute states for receivers on an LAN by limiting the number of IGMP joins on an outgoing interface.

Both the IGMP State Limit and Per Interface Mroute State Limit features provide a rudimentary multicast CAC mechanism that can be used to provision bandwidth utilization on an interface when all multicast flows roughly utilize the same amount of bandwidth. The Bandwidth-Based CAC for IP Multicast feature, however, offers a more flexible and powerful alternative for providing multicast CAC in network environments where IP multicast flows utilize different amounts of bandwidth.

Note For more information about the Bandwidth-Based CAC for IP Multicast feature, see the Bandwidth-Based CAC for IP Multicast, on page 641.

Prerequisites for Per Interface Mroute State Limit

IP multicast is enabled and the Protocol Independent Multicast (PIM) interfaces are configured using the tasks described in the "Configuring Basic IP Multicast" module of the *IP Multicast: PIM Configuration Guide*.

Information about Per Interface Mroute State Limit

Mechanics of Per Interface Mroute State Limiters

The mechanics of per interface mroute state limiters are as follows:

- Each time the state for an mroute is created or deleted and each time an olist member is added or removed, the software searches for a corresponding per interface mroute state limiter that matches the mroute.
- When an mroute is created or deleted, the software searches for a per interface mroute state limiter configured on the incoming (RPF) interface that matches the mroute to be created or deleted. When an olist member is added or removed, the software searches for a per interface mroute state limiter configured on the outgoing interface that matches the mroute to be added or removed.
- A top-down search is performed using the list of configured per interface mroute state limiters. Only per interface mroute state limiters that match the direction of traffic are considered. The first per interface mroute state limiter that matches is used for limiting (sometimes referred to as accounting). A match is found when the ACL permits the mroute state.

- When a match is found, the counter of the per interface mroute state limiter is updated (increased or decreased). If no per interface mroute state limiter is found that matches an mroute, no accounting is performed for the mroute (because there is no counter to update).
- The amount with which to update the counter is called the cost (sometimes referred to as the cost multiplier). The default cost is 1.

Note A per interface mroute state limiter always allows the deletion of an mroute or the removal of an interface from the olist. In those cases, the respective per interface mroute state limiter decreases the counter by the value of the cost multiplier. In addition, RPF changes to an existing mroute are always allowed (in order to not affect existing traffic). However, a per interface mroute state limiter only allows the creation of an mroute or the addition of an mroute olist member if adding the cost does not exceed the maximum number of mroutes permitted.

Tips for Configuring Per Interface Mroute State Limiters

- To ensure that all mroutes are accounted, you can configure a per interface mroute state limiter whose ACL contains a permit-any statement and set the value of zero (0) for maximum entries. Configuring an mroute state limiter in this manner effectively denies all fall through states, which may be a way to prevent a multicast DoS attack in and out of the interface.
- When creating an ACL, remember that, by default, the end of the ACL contains an implicit deny-any statement for everything if it did not find a match before reaching the end.
- An explicit deny statement for a specific mroute in an ACL can be used to specify the state that will not match the ACLwhich will prevent the ACL from being accounted. If an mroute matches a deny statement, the search immediately continues to the next configured mroute state limiter. Configuring an explicit deny statement in an ACL can be more efficient than forcing the mroute to fall through an ACL by using an implicit deny-any statement at the end of the ACL.

How to Configure Per Interface Mroute State Limit

Configuring Per Interface Mroute State Limiters

Perform this task to prevent DoS attacks or to provide a multicast CAC mechanism for controling bandwidth when all multicast flows utilize approximately the same amount of bandwidth.

Before you begin

All ACLs to be applied to per interface mroute state limiters must be configured prior to beginning this configuration task; otherwise, the limiters are ignored. For information about how to configure ACLs, see the “ Creating an IP Access List and Applying It to an Interface ” module of the *Security Configuration Guide: Access Control Lists* guide.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip multicast limit** [**connected** | **out** | **rpf**] *access-list max-entries*
5. Repeat Step 4 to configure additional per interface mroute state limiters on this interface.
6. Repeat Steps 3 and Step 4 to configure per interface mroute state limiters on additional interfaces.
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0`	Enters interface configuration mode for the specified interface type and number.
Step 4	**ip multicast limit** [**connected** | **out** | **rpf**] *access-list max-entries* **Example:** `Device(config-if)# ip multicast limit 15 100`	Configures per interface mroute state limiters.
Step 5	Repeat Step 4 to configure additional per interface mroute state limiters on this interface.	--
Step 6	Repeat Steps 3 and Step 4 to configure per interface mroute state limiters on additional interfaces.	--
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Monitoring Per Interface Mroute State Limiters and Bandwidth-Based Multicast CAC Policies

Perform this optional task to monitor per interface mroute state limiters and bandwidth-based multicast CAC policies.

SUMMARY STEPS

1. **enable**
2. **debug ip mrouting limits** [*group-address*]
3. **show ip multicast limit** *type number*
4. **clear ip multicast limit** [*type number*]

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **debug ip mrouting limits** [*group-address*]

Displays debugging information about configured per interface mroute state limiters and bandwidth-based multicast CAC policies.

The following output is from the **debug ip mrouting limits** command. The output displays the following events:

- An mroute state being created and the corresponding per interface mroute state limiter counter being increased by the default cost of 1 on incoming Ethernet interface 1/0.
- An mroute olist member being removed from the olist and the corresponding per interface mroute limiter being decreased by the default cost of 1 on outgoing Ethernet interface 1/0.
- An mroute being denied by the per interface mroute state limiter because the maximum number of mroute states has been reached.
- An mroute state being created and the corresponding per interface mroute state limiter counter being increased by the cost of 2 on incoming Ethernet interface 1/0.
- An mroute olist member being removed from the olist and the corresponding per interface mroute limiter being decreased by a cost of 2 on outgoing Ethernet interface 1/0.

Example:

```
device# debug ip mrouting limits

MRL(0): incr-ed acl 'rpf-list' to (13 < max 32), [n:0,p:0], (main) GigabitEthernet0/0, (10.41.0.41,
 225.30.200.60)
MRL(0): decr-ed acl 'out-list' to (10 < max 32), [n:0,p:0], (main) GigabitEthernet0/0, (*,
```

```
225.40.202.60)
MRL(0): Add mroute (10.43.0.43, 225.30.200.60) denied for GigabitEthernet0/2, acl std-list, (16 =
max 16)
MRL(0): incr-ed limit-acl `rpf-list' to (12 < max 32), cost-acl 'cost-list' cost 2, [n:0,p:0], (main)
 GigabitEthernet0/0, (10.41.0.41, 225.30.200.60)
MRL(0): decr-ed limit-acl `out-list' to (8 < max 32), cost-acl 'cost-list'' cost 2, [n:0,p:0], (main)
 GigabitEthernet0/0, (*, 225.40.202.60)
```

Step 3 **show ip multicast limit** *type* *number*

Displays counters related to mroute state limiters configured on the interfaces on the router.

For each per interface mroute state limiter shown in the output, the following information is displayed:

- The direction of traffic that the per mroute state limiter is limiting.
- The ACL referenced by the per interface mroute state limiter that defines the IP multicast traffic being limited.
- Statistics, enclosed in parenthesis, which track the current number of mroutes being limited less the configured limit. Each time the state for an mroute is created or deleted and each time an outgoing interface list (olist) member is added or removed, the counters for matching per interface mroute state limiters are increased or decreased accordingly.
- The exceeded counter, which tracks the total number of times that the limit configured for the per interface mroute state limiter has been exceeded. Each time an mroute is denied due to the configured limit being reached, the exceeded counter is increased by a value of 1.

The following is sample output from the **show ip multicast limit** command with the *type number* arguments. In this example, information about mroute state limiters configured on Gigabit Ethernet interface 0/0 is displayed.

Example:

```
Device# show ip multicast limit GigabitEthernet 0/0

Interface GigabitEthernet 0/0
 Multicast Access Limits
out acl out-list (1 < max 32) exceeded 0
rpf acl rpf-list (6 < max 32) exceeded 0
con acl conn-list (0 < max 32) exceeded 0
```

Step 4 **clear ip multicast limit** [*type number*]

Resets the exceeded counter for per interface mroute state limiters.

The following example shows how to reset exceeded counters for per interface mroute state limiters configured on Gigabit Ethernet interface 0/0:

Example:

```
Device# clear ip multicast limit interface GigabitEthernet 0/0
```

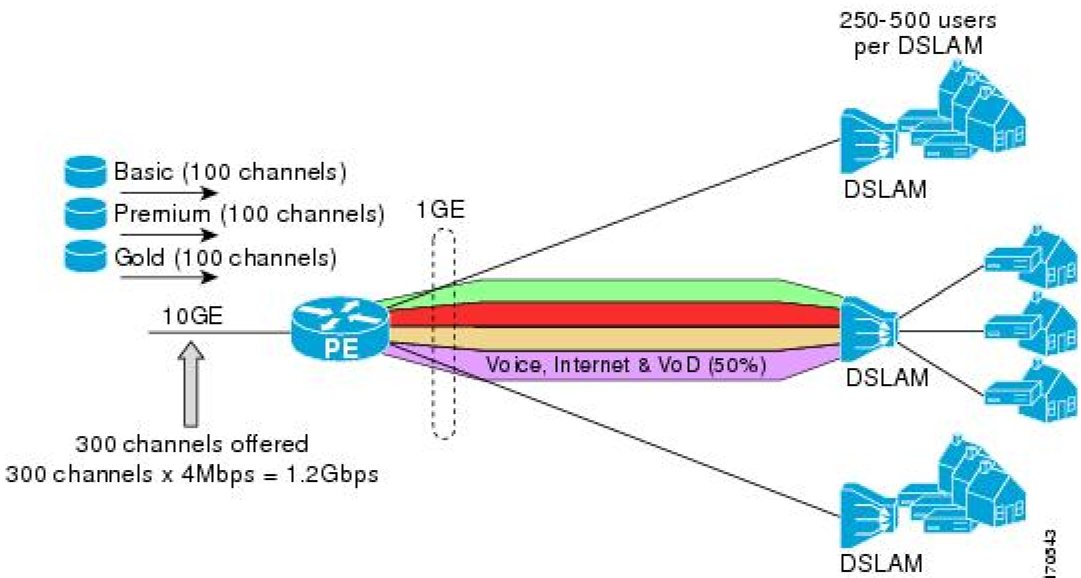
Configuration Examples for Per Interface Mroute State Limit

Example Configuring Per Interface Mroute State Limiters

The following example shows how to configure per interface mroute state limiters to provide multicast CAC in a network environment where all the multicast flows roughly utilize the same amount of bandwidth.

This example uses the topology illustrated in the figure.

Figure 74: Per Interface Mroute State Limit Example Topology



In this example, a service provider is offering 300 SD TV channels. The SD channels are being offered to customers in three service bundles (Basic, Premium, and Gold), which are available to customers on a subscription basis. Each bundle offers 100 channels to subscribers, and each channel utilizes approximately 4 Mbps of bandwidth.

The service provider must provision the Gigabit Ethernet interfaces on the PE device connected to DSLAMs as follows: 50% of the link's bandwidth (500 Mbps) must be available to subscribers of their Internet, voice, and VoD service offerings while the remaining 50% (500 Mbps) of the link's bandwidth must be available to subscribers of their SD channel bundle service offerings.

For the 500 Mbps of the link's bandwidth that must always be available to (but must never be exceeded by) the subscribers of the SD channel bundles, the interface must be further provisioned as follows:

- 60% of the bandwidth must be available to subscribers of the basic service (300 Mbps).
- 20% of the bandwidth must be available to subscribers of the premium service (100 Mbps).
- 20% of the bandwidth must be available to subscribers of the gold service (100 Mbps).

Because each SD channel utilizes the same amount of bandwidth (4 Mbps), per interface mroute state limiters can be used to provide the necessary CAC to provision the services being offered by the service provider. To determine the required CAC needed per interface, the number of channels for each bundle is divided by 4 (because each channel utilizes 4 Mbps of bandwidth). The required CAC needed per interface, therefore, is as follows:

- Basic Services: 300 / 4 = 75
- Premium Services: 100 / 4 = 25
- Gold Services: 100 / 4 = 25

Once the required CAC required per SD channel bundle is determined, the service provider uses the results to configure the mroute state limiters required to provision the Gigabit Ethernet interfaces on the PE device for the services being offered to subscribers behind the DSLAMs:

- For the Basic Services bundle, the service provider must limit the number of Basic Service SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 75. Configuring an mroute state limit of 75 for the SD channels offered in the Basic Service bundle provisions the interface for 300 Mbps of bandwidth (the 60% of the link's bandwidth that must always be available to [but never exceeded by] the subscribers of the Basic Services bundle).
- For the Premium Services bundle, the service provider must limit the number of Premium Service SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 25. Configuring an mroute state limit of 25 for the SD channels offered in the Premium Service bundle provisions the interface for 100 Mbps of bandwidth (the 20% of the link's bandwidth that must always be available to [but never exceeded by] the subscribers of the Premium Service bundle).
- For the Gold Services bundle, the service provider must limit the number of Gold Service SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 25. Configuring an mroute state limit of 25 for the SD channels offered in the Gold Service bundle provisions the interface for 100 Mbps of bandwidth (the 20% of the link's bandwidth that must always be available to [but never exceeded by] the subscribers of the Gold Service bundle).

The service provider then configures three ACLs to be applied to per interface mroute state limiters. Each ACL defines the SD channels for each SD channel bundle to be limited on an interface:

- acl-basic--The ACL that defines the SD channels offered in the basic service.
- acl-premium--The ACL that defines the SD channels offered in the premium service.
- acl-gold--The ACL that defines the SD channels offered in the gold service.

These ACLs are then applied to per interface mroute state limiters configured on the PE device's Gigabit Ethernet interfaces.

For this example, three per interface mroute state limiters are configured on Gigabit Ethernet interface 0/0 to provide the multicast CAC needed to provision the interface for the SD channel bundles being offered to subscribers:

- An mroute state limit of 75 for the SD channels that match acl-basic.
- An mroute state limit of 25 for the SD channels that match acl-premium.
- An mroute state limit of 25 for the SD channels that match acl-gold.

The following configuration shows how the service provider uses per interface mroute state limiters to provision Gigabit Ethernet interface 0/0 for the SD channel bundles and Internet, Voice, and VoD services being offered to subscribers:

```
interface GigabitEthernet0/0
description --- Interface towards the DSLAM ---
.
.
.
ip multicast limit out acl-basic 75
ip multicast limit out acl-premium 25
ip multicast limit out acl-gold 25
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP multicast commands	Cisco IOS IP Multicast Command Reference

Standards and RFCs

Standard/RFC	Title
No new or modified standards or RFCs are supported by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Per Interface Mroute State Limit

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 40: Feature Information for Per Interface Mroute State Limit

Feature Name	Releases	Feature Information
Per Interface Mroute State Limit	Cisco IOS XE Release 2.1 12.3(14)T 12.2(33)SRB 12.2(33)SXI 15.1(1)SG Cisco IOX XE Release 3.3.0SG	The Per Interface Mroute State Limit feature provides the capability to limit the number of mroute states on an interface for different ACL-classified sets of multicast traffic. This feature can be used to prevent DoS attacks, or to provide a multicast CAC mechanism in network environments where all the multicast flows roughly utilize the same amount of bandwidth. The following commands were introduced or modified: **clear ip multicast limit**, **debug ip mrouting limits**, **ip multicast limit**, **show ip multicast limit**.

CHAPTER 50

SSM Channel Based Filtering for Multicast Boundaries

The SSM Channel Based Filtering for Multicast Boundaries feature enables the user to apply filtering policies based on Source Specific Multicast (SSM) channels for Source and Group (S,G) addresses, which is a combination of source and destination IP addresses.

Prerequisites for SSM Channel Based Filtering for Multicast Boundaries

IP multicast is enabled on the device using the tasks described in the "Configuring Basic IP Multicast" module of the *IP Multicast: PIM Configuration Guide*.

Information About the SSM Channel Based Filtering for Multicast Boundaries Feature

Rules for Multicast Boundaries

The SSM Channel Based Filtering for Multicast Boundaries feature expands the **ip multicast boundary** command for control plane filtering support. More than one **ip multicast boundary** command can be applied to an interface.

The following rules govern the **ip multicast boundary** command:

- One instance of the **in** and **out** keywords can be configured on an interface.

- The **in** and **out** keywords can be used for standard or extended access lists.
- Only standard access lists are permitted with the use of the **filter-autorp** keyword or no keyword.
- A maximum of three instances of a command will be allowed on an interface: one instance of **in**, one instance of **out**, and one instance of **filter-autorp** or no keyword.
- When multiple instances of the command are used, the filtering will be cumulative. If a boundary statement with no keyword exists with a boundary statement with the **in**keyword, both access lists will be applied on the in direction and a match on either one will be sufficient.
- All instances of the command apply to both control and data plane traffic.
- Protocol information on the extended access list is parsed to allow reuse and filtering for consistency. An (S,G) operation will be filtered by an extended access list under all conditions stated above for keywords if the access list filters (S,G) traffic for all protocols.

Benefits of SSM Channel Based Filtering for Multicast Boundaries

- This feature allows input on the source interface.
- The access control capabilities are the same for SSM and Any Source Multicast (ASM).

How to Configure SSM Channel Based Filtering for Multicast Boundaries

Configuring Multicast Boundaries

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard**| **extended**} *access-list-name*
4. **permit** *protocol* **host** *address* **host** *address*
5. **deny** *protocol* **host** *address* **host** *address*
6. Repeat Step 4 or Step 5 as needed.
7. **interface** *type* **interface-number** *port -number*
8. **ip multicast boundary** *access-list-name* [**in**| **out** | **filter-autorp**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list** {**standard**| **extended**} *access-list-name* **Example:** `Device(config)# ip access-list 101`	Configures the standard or extended access list.
Step 4	**permit** *protocol* **host** *address* **host** *address* **Example:** `Device(config-ext-nacl)# permit ip host 181.1.2.201 host 232.1.1.11`	Permits specified ip host traffic.
Step 5	**deny** *protocol* **host** *address* **host** *address* **Example:** `Device(config-acl-nacl)# deny ip host 181.1.2.203 host 232.1.1.1`	Denies specified multicast ip group and source traffic.
Step 6	Repeat Step 4 or Step 5 as needed.	Permits and denies specified host and source traffic.
Step 7	**interface** *type* **interface-number** *port -number* **Example:** `Device(config)# interface gigabitethernet 2/3/0`	Enables interface configuration mode.
Step 8	**ip multicast boundary** *access-list-name* [**in**| **out** | **filter-autorp**] **Example:** `Device(config-if)# ip multicast boundary acc_grp1 out`	Configures the multicast boundary. **Note** The **filter-autorp** keyword does not support extended access lists.

Configuration Examples for SSM Channel Based Filtering for Multicast Boundaries

Configuring the Multicast Boundaries Permitting and Denying Traffic Example

The following example permits outgoing traffic for (181.1.2.201, 232.1.1.1) and (181.1.2.202, 232.1.1.1) and denies all other (S,G)s.

```
configure terminal
 ip access-list extended acc_grp1
 permit ip host 0.0.0.0 232.1.1.1 0.0.0.255
 permit ip host 181.1.2.201 host 232.1.1.1
 permit udp host 181.1.2.202 host 232.1.1.1
 permit ip host 181.1.2.202 host 232.1.1.1
 deny igmp host 181.2.3.303 host 232.1.1.1
 interface gigabitethernet 2/3/0
 ip multicast boundary acc_grp1 out
```

Configuring the Multicast Boundaries Permitting Traffic Example

The following example permits outgoing traffic for (192.168.2.201, 232.1.1.5) and 192.168.2.202, 232.1.1.5).

```
configure terminal
ip access-list extended acc_grp6
 permit ip host 0.0.0.0 232.1.1.1 5.0.0.255
 deny udp host 192.168.2.201 host 232.1.1.5
 permit ip host 192.168.2.201 host 232.1.1.5
 deny pim host 192.168.2.201 host 232.1.1.5
 permit ip host 192.168.2.202 host 232.1.1.5
 deny igmp host 192.2.3.303 host 232.1.1.1
interface gigabitethernet 2/3/0
 ip multicast boundary acc_grp6 out
```

Configuring the Multicast Boundaries Denying Traffic Example

The following example denies a group-range that is announced by the candidate RP. Because the group range is denied, no pim auto-rp mappings are created.

```
configure terminal
ip access-list standard acc_grp10
 deny 225.0.0.0 0.255.255.255
 permit any
 access-list extended acc_grp12
 permit pim host 181.1.2.201 host 232.1.1.8
 deny udp host 181.1.2.201 host 232.1.1.8
 permit pim host 181.1.2.203 0.0.0.255 host 227.7.7.7
 permit ip host 0.0.0.0 host 227.7.7.7
 permit ip 181.1.2.203 0.0.0.255 host 227.7.7.7
 permit ip host 181.1.2.201 host 232.1.1.7
ip access-list extended acc_grp13
 deny ip host 181.1.2.201 host 232.1.1.8
 permit ip any any
interface gigabitethernet 2/3/0
 ip multicast boundary acc_grp10 filter-autorp
 ip multicast boundary acc_grp12 out
 ip multicast boundary acc_grp13 in
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS IP Multicast commands	Cisco IOS IP Multicast Command Reference

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing standards has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for SSM Channel Based Filtering for Multicast Boundaries

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 41: Feature Information for SSM Channel Based Filtering for Multicast Boundaries

Feature Name	Releases	Feature Information
SSM Channel Based Filtering for Multicast Boundaries	Cisco IOS XE Release 2.1	The SSM Channel Based Filtering for Multicast Boundaries feature enables the user to apply filtering policies based on Source Specific Multicast (SSM) channels for Source and Group (S,G) addresses, which is a combination of source and destination IP addresses. The following command was introduced or modified: • **ip multicast boundary**

CHAPTER 51

IPv6 Multicast: Bandwidth-Based Call Admission Control

Information About IPv6 Multicast: Bandwidth-Based Call Admission Control

Bandwidth-Based CAC for IPv6 Multicast

The bandwidth-based call admission control (CAC) for IPv6 multicast feature implements a way to count per-interface mroute state limiters using cost multipliers. This feature can be used to provide bandwidth-based CAC on a per-interface basis in network environments where the multicast flows use different amounts of bandwidth.

This feature limits and accounts for IPv6 multicast state in detail. When this feature is configured, interfaces can be limited to the number of times they may be used as incoming or outgoing interfaces in the IPv6 multicast PIM topology.

With this feature, device administrators can configure global limit cost commands for state matching access lists and specify which cost multiplier to use when accounting such state against the interface limits. This feature provides the required flexibility to implement bandwidth-based local CAC policy by tuning appropriate cost multipliers for different bandwidth requirements.

Threshold Notification for mCAC Limit

The threshold notification for mCAC limit feature notifies the user when actual simultaneous multicast channel numbers exceeds or fall below a specified threshold percentage. For example, if the mCAC rate limit is set to 50,000,000 and the configured threshold percentage is 80 percent, then the user is notified if the limit exceeds 10,000,000.

How to Implement IPv6 Multicast Bandwidth-Based Call Admission Control

Configuring the Global Limit for Bandwidth-Based CAC in IPv6

Device administrators can configure global limit cost commands for state matching access lists.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast** [**vrf** *vrf-name*] **limit cost** *access-list cost-multiplier*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 multicast** [**vrf** *vrf-name*] **limit cost** *access-list cost-multiplier* **Example:** `Device(config)# ipv6 multicast limit cost costlist1 2`	Applies a cost to mroutes that match per-interface mroute state limiters in IPv6.

Configuring an Access List for Bandwidth-Based CAC in IPv6

In bandwidth-based CAC for IPv6, device administrators can configure global limit cost commands for state matching access lists. Perform this task to configure an access list to configure a state matching access list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. Use one of the following:
   - **permit**
   - **deny**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list costlist1`	Defines an IPv6 access list and places the device in IPv6 access list configuration mode.

Step 4	Use one of the following:
	• **permit**
	• **deny**
	Example:
	`Device(config)# permit any ff03::1/64`
	`Device(config_# deny any ff03::1/64`
		Sets conditions for an IPv6 access list.

Configuring the Interface Limit for Bandwidth-Based CAC in IPv6

Bandwidth-based CAC for IPv6 counts per-interface IPv6 mroute states using cost multipliers. With this feature, device administrators can specify which cost multiplier to use when accounting such state against the interface limits.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **configure terminal**
4. **interface** *type number*
5. **ipv6 address** [*ipv6-address* / *prefix-length* | *prefix-name sub-bits* / *prefix-length*]
6. **ipv6 multicast limit** [**connected** / **rpf** | **out**] *limit-acl max*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/3`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 5	**ipv6 address** [*ipv6-address* / *prefix-length* | *prefix-name sub-bits* / *prefix-length*] **Example:** `Device(config-if)# ipv6 address FE80::40:1:3 link-local`	Configures an IPv6 address based on an IPv6 general prefix.
Step 6	**ipv6 multicast limit** [**connected** / **rpf** | **out**] *limit-acl max* **Example:** `Device(config-if)# ipv6 multicast limit out acl1 10`	Configures per-interface mroute state limiters in IPv6.

Configuring the Threshold Notification for the mCAC Limit in IPv6

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast limit rate** *rate-value*
4. **interface** *type number*
5. **ipv6 multicast limit** [**connected** | **rpf** | **out**] *limit-acl max* [**threshold** *threshold-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ipv6 multicast limit rate** *rate-value* **Example:** `Device(config)# ipv6 multicast limit rate 2`	Configures the maximum allowed state on the source device.
Step 4	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/3/1`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 5	**ipv6 multicast limit** [**connected** | **rpf** | **out**] *limit-acl max* [**threshold** *threshold-value*] **Example:** `Device (config-if)# ipv6 multicast limit out acl1 10 threshold 20`	Configures per-interface mroute state limiters in IPv6.

Configuration Examples for IPv6 Multicast Bandwidth-Based Call Admission Control

Example: Configuring the Global Limit for Bandwidth-Based CAC

The following example configures the global limit on the source device.

```
ipv6 multicast limit cost cost-list 2
```

Example: Configuring an Access List for Bandwidth-Based CAC in IPv6

The following example shows how to configure an access list to use for bandwidth-based CAC:

```
ipv6 access-list cost-list
   permit any ff03::1/64
```

Example: Configuring the Interface Limit for Bandwidth-Based CAC in IPv6

The following example configures the interface limit on the source device's outgoing interface Ethernet 1/3.

```
interface Ethernet1/3
ipv6 address FE80::40:1:3 link-local
```

```
ipv6 address 2001:DB8:1:1:3/64
ipv6 multicast limit out acl1 10
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Multicast: Bandwidth-Based Call Admission Control

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 42: Feature Information for IPv6 Multicast: Bandwidth-Based Call Admission Control

Feature Name	Releases	Feature Information
IPv6 Multicast: Bandwidth-Based Call Admission Control	12.2(40)SG 3.2.0SG 15.0(2)SG 12.2(33)SRE Cisco IOS XE Release 2.6 15.0(1)S	This feature can be used to provide bandwidth-based CAC on a per-interface basis in network environments where the multicast flows use different amounts of bandwidth. The following commands were introduced or modified: **ipv6 access-list**, **ipv6 address**, **ipv6 multicast limit**, **ipv6 multicast limit cost**.
mCAC enhancement: configurable treshold notification for mCAC limits	Cisco IOS XE Release 2.6	This feature enables system notifications when actual simultaneous multicast channel numbers exceeds or fall below some percentage (called threshold percantage). The following command were introduced or modified by this feature: **ipv6 multicast limit**, **ipv6 multicast limit rate**.

CHAPTER 52

PIM Dense Mode State Refresh

This feature module describes the Protocol Independent Multicast (PIM) Dense Mode (DM) State Refresh feature, which is an extension to the dense operational mode of the PIM Version 2 multicast routing architecture.

Prerequisites for PIM Dense Mode State Refresh

- You must have PIM dense mode enabled on an interface before configuring the PIM Dense Mode State Refresh feature.
- The origination of state refresh control messages is disabled by default. Enable the origination of state refresh control messages so that these control messages can be processed and forwarded as part of the PIM Dense Mode State Refresh feature.

Restrictions on PIM Dense Mode State Refresh

- All devices in a PIM dense mode network must run a software release that supports the PIM Dense Mode State Refresh feature to process and forward state refresh control messages.
- The origination interval for the state refresh control message must be the same for all PIM devices on the same LAN. Specifically, the same origination interval must be configured on each device interface that is directly connected to the LAN.

Information About PIM Dense Mode State Refresh

PIM Dense Mode State Refresh Overview

The PIM Dense Mode State Refresh feature is an extension of the PIM Version 2 multicast routing architecture.

PIM dense mode builds source-based multicast distribution trees that operate on a flood and prune principle. Multicast packets from a source are flooded to all areas of a PIM dense mode network. PIM routers that receive multicast packets and have no directly connected multicast group members or PIM neighbors send a prune message back up the source-based distribution tree toward the source of the packets. As a result, subsequent multicast packets are not flooded to pruned branches of the distribution tree. However, the pruned state in PIM dense mode times out approximately every 3 minutes and the entire PIM dense mode network is reflooded with multicast packets and prune messages. This reflooding of unwanted traffic throughout the PIM dense mode network consumes network bandwidth.

The PIM Dense Mode State Refresh feature keeps the pruned state in PIM dense mode from timing out by periodically forwarding a control message down the source-based distribution tree. The control message refreshes the prune state on the outgoing interfaces of each router in the distribution tree.

Benefits of PIM Dense Mode State Refresh

The PIM Dense Mode State Refresh feature keeps the pruned state in PIM dense mode from timing out, which saves network bandwidth by greatly reducing the reflooding of unwanted multicast traffic to pruned branches of the PIM dense mode network. This feature also enables PIM devices in a PIM dense mode multicast network to recognize topology changes (sources joining or leaving a multicast group) before the default 3-minute state refresh timeout period.

How to Configure PIM Dense Mode State Refresh

Configuring PIM Dense Mode State Refresh

There are no configuration tasks for enabling the PIM Dense Mode State Refresh feature. By default, all PIM devices that are running a Cisco software release that supports the PIM Dense Mode State Refresh feature automatically process and forward state refresh control messages.

To disable the processing and forwarding of state refresh control messages on a PIM device, use the **ip pim state-refresh disable**global configuration command. To enable state refresh again if it has been disabled, use the **no ip pim state-refresh disable**global configuration command.

The origination of state refresh control messages is disabled by default. In PIM dense mode, enable the control message origination on the interface of the first-hop device to trigger a state-refresh. This will help prune the multicast table with accurate state entries for all devices that are running in PIM dense mode. To configure the origination of the control messages on a device running PIM, use the following commands beginning in global configuration mode:

Command	Purpose
`Device(config)# interface type number`	Specifies an interface and places the device in interface configuration mode.
`Device(config-if)# ip pim state-refresh origination-interval [interval]`	Configures the origination of the PIM Dense Mode State Refresh control message. Optionally, you can configure the number of seconds between control messages by using the *interval* argument. The default interval is 60 seconds. The interval range is 1 second to 100 seconds.

Verifying PIM Dense Mode State Refresh Configuration

Use the **show ip pim interface** [*type number*] **detail** and the **show ip pim neighbor** [*interface*] commands to verify that the PIM Dense Mode State Refresh feature is configured correctly. The following ample output indicates that processing, forwarding, and origination of state refresh control messages is enabled.

```
Device# show ip pim interface fastethernet 0/1/0 detail
FastEthernet0/1/0 is up, line protocol is up
  Internet address is 172.16.8.1/24
  Multicast switching:process
  Multicast packets in/out:0/0
  Multicast boundary:not set
  Multicast TTL threshold:0
  PIM:enabled
    PIM version:2, mode:dense
    PIM DR:172.16.8.1 (this system)
    PIM neighbor count:0
    PIM Hello/Query interval:30 seconds
 PIM State-Refresh processing:enabled
 PIM State-Refresh origination:enabled, interval:60 seconds
    PIM NBMA mode:disabled
    PIM ATM multipoint signalling:disabled
    PIM domain border:disabled
  Multicast Tagswitching:disabled
```

The S in the Mode field of the following **show ip pim neighbor** [*interface*] command output indicates that the neighbor has the PIM Dense Mode State Refresh feature configured.

```
Device# show ip pim neighbor
PIM Neighbor Table
Neighbor          Interface          Uptime/Expires    Ver   DR
Address                                                      Priority/Mode
172.16.5.1        Ethernet1/1        00:09:03/00:01:41 v2    1 / B S
```

Monitoring and Maintaining PIM DM State Refresh

Following are the PIM Dense Mode State Refresh control messages that are sent and received by a PIM router after the **debug ip pim** privileged EXEC command is configured for multicast group 239.0.0.1:

```
Router# debug ip pim 239.0.0.1
*Mar  1 00:25:10.416:PIM:Originating refresh message for
(172.16.8.3,239.0.0.1)
```

```
*Mar  1 00:25:10.416:PIM:Send SR on GigabitEthernet1/1/0 for (172.16.8.3,239.0.0.1)
TTL=9
```

The following output from the **show ip mroute** command displays are the resulting prune timer changes for GigabitEthernet interface1/0/0 and multicast group 239.0.0.1. (The following output assumes that the **debug ip pim** privileged EXEC command has already been configured on the router.) In the first output from the **show ip mroute** command, the prune timer reads 00:02:06. The debug messages indicate that a PIM Dense Mode State Refresh control message is received and sent on Ethernet interface 1/0, and that other PIM Dense Mode State Refresh routers were discovered. In the second output from the **show ip mroute** command, the prune timer has been reset to 00:02:55.

```
Router# show ip mroute 239.0.0.1
(172.16.8.3, 239.0.0.1), 00:09:50/00:02:06, flags:PT
  Incoming interface:GigabitEthernet1/1/0, RPF nbr 172.16.5.2
  Outgoing interface list:
 GigabitEthernet1/0/0, Prune/Dense, 00:09:43/00:02:06
Router#
*Mar  1 00:32:06.657:PIM:SR on iif from 172.16.5.2 orig 172.16.8.1 for
(172.16.8.3,239.0.0.1)
*Mar  1 00:32:06.661:     flags:prune-indicator
*Mar  1 00:32:06.661:PIM:Cached metric is [0/0]
*Mar  1 00:32:06.661:PIM:Keep RPF nbr 172.16.5.2
*Mar  1 00:32:06.661:PIM:Send SR on Ethernet1/0 for (172.16.8.3,239.0.0.1)
TTL=8
*Mar  1 00:32:06.661:     flags:prune-indicator
Router# show ip mroute 239.0.0.1
(172.16.8.3, 239.0.0.1), 00:10:01/00:02:55, flags:PT
  Incoming interface:GigabitEthernet1/1/0, RPF nbr 172.16.5.2
  Outgoing interface list:
 GigabitEthernet1/0/0, Prune/Dense, 00:09:55/00:02:55
```

Configuration Examples for PIM Dense Mode State Refresh

Originating Processing and Forwarding PIM Dense Mode State Refresh Control Messages Example

The following example is for a PIM router that is originating, processing, and forwarding PIM Dense Mode State Refresh control messages on Fast Ethernet interface 0/1/0 every 60 seconds:

```
ip multicast-routing distributed
interface FastEthernet0/1/0
 ip address 172.16.8.1 255.255.255.0
 ip pim state-refresh origination-interval 60
 ip pim dense-mode
```

Example: Processing and Forwarding PIM Dense Mode State Refresh Control Messages

The following example is for a PIM device that is just processing and forwarding PIM Dense Mode State Refresh control messages on Fast Ethernet interface 1/1/0:

```
ip multicast-routing
```

```
interface FastEthernet1/1/0
 ip address 172.16.7.3 255.255.255.0
 ip pim dense-mode
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS IP Multicast commands	Cisco IOS IP Multicast Command Reference

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing standards has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for PIM Dense Mode State Refresh

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 43: Feature Information for PIM Dense Mode State Refresh

Feature Name	Releases	Feature Information
PIM Dense Mode State Refresh	12.2(4)T 12.2(27)SBB Cisco IOS XE Release 2.1	PIM Dense Mode State Refresh is an extension to the dense operational mode of the PIM Version 2 multicast routing architecture. The following commands are introduced or modified in the feature documented in this module: **ip pim state-refresh disable** , **ip pim state-refresh origination-interval** , **show ip pim interface** .