IP Multicast Routing Configuration Guide, Cisco IOS XE Fuji 16.9.x (Catalyst 3650 Switches)

First Published: 2018-07-18

CONTENTS

CHAPTER 5 **Configuring VRF-lite** **39**

CHAPTER 13   **Configuring the Service Discovery Gateway** **271**

IP Multicast Routing Technology Overview

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.

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.

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 (and multilayer devices) on that LAN to track the multicast groups of which hosts are members. To participate in IP multicasting, multicast hosts, routers, and multilayer devices must have the Internet Group Management Protocol (IGMP) operating.

- 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.

- IGMP Snooping is used for multicasting in a Layer 2 switching environment. It helps reduce the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices.

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

Figure 1: IP Multicast Routing Protocols

According to IPv4 multicast standards, the MAC destination multicast address begins with 0100:5e and is appended by the last 23 bits of the IP address. For example, if the IP destination address is 239.1.1.39, the MAC destination address is 0100:5e01:0127.

A multicast packet is unmatched when the destination IPv4 address does not match the destination MAC address. The device forwards the unmatched packet in hardware based upon the MAC address table. If the destination MAC address is not in the MAC address table, the device floods the packet to the all port in the same VLAN as the receiving port.

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 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 2: Address Scoping at Boundaries
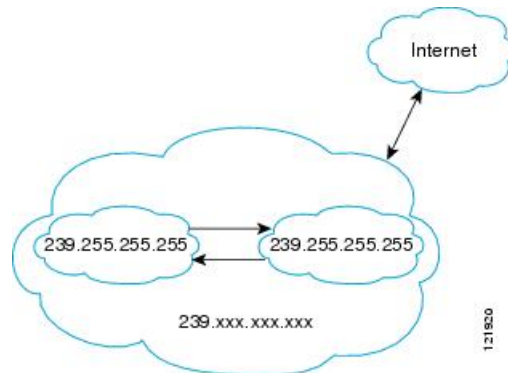
You can set up an administratively scoped boundary on an interface for multicast group addresses using the **ip multicast boundary** command with the *access-list* argument. A standard access list defines the range of addresses affected. When a boundary is set up, no multicast data packets are allowed 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.

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.

Note   All the packets with reserved link-local addresses are punted to CPU by default in the ASR 903 RSP2 Module.

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 6 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.

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.

Configuring Basic IP Multicast Routing

Prerequisites for Basic IP Multicast Routing

The following are the prerequisites for configuring basic IP multicast routing:

- To use this feature, the device must be running the IP services feature set. The IP Services image contains complete multicast routing.

- You must configure the PIM version and the PIM mode in order to perform IP multicast routing. The switch populates its multicast routing table and forwards multicast packets it receives from its directly connected LANs according to the mode setting. You can configure an interface to be in the PIM dense mode, sparse mode, or sparse-dense mode.

- Enabling PIM on an interface also enables IGMP operation on that interface. (To participate in IP multicasting, the multicast hosts, routers, and multilayer device must have IGMP operating.)

  If you enable PIM on multiple interfaces, when most of these interfaces are not on the outgoing interface list, and IGMP snooping is disabled, the outgoing interface might not be able to sustain line rate for multicast traffic because of the extra replication.

Restrictions for Basic IP Multicast Routing

The following are the restrictions for IP multicast routing:

- IP multicast routing is not supported on switches running the LAN base feature set.

- You cannot have a device stack containing a mix of Catalyst 3850 and Catalyst 3650 devices.

Information About Basic IP Multicast Routing

IP multicasting is an efficient way to use network resources, especially for bandwidth-intensive services such as audio and video. 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 devices forward incoming IP multicast packets out all interfaces that lead to 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 Forwarding Information Base Overview

The device uses the Multicast Forwarding Information Base (MFIB) architecture and the Multicast Routing Information Base (MRIB) for IP multicast.

The MFIB architecture provides both 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.

MFIB itself 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

The MRIB is the communication channel between MRIB clients. Examples of MRIB clients are PIM, IGMP, the multicast routing (mroute) table, and the MFIB.

Default IP Multicast Routing Configuration

This table displays the default IP multicast routing configuration.

Table 2: Default IP Multicast Routing Configuration

Feature	Default Setting
Multicast routing	Disabled on all interfaces.
PIM version	Version 2.
PIM mode	No mode is defined.
PIM stub routing	None configured.
PIM RP address	None configured.
PIM domain border	Disabled.

Feature	Default Setting
PIM multicast boundary	None.
Candidate BSRs	Disabled.
Candidate RPs	Disabled.
Shortest-path tree threshold rate	0 kb/s.
PIM router query message interval	30 seconds.

How to Configure Basic IP Multicast Routing

Configuring Basic IP Multicast Routing

By default, multicast routing is disabled, and there is no default mode setting.

This procedure is required.

Before you begin

You must configure the PIM version and the PIM mode. The switch populates its multicast routing table and forwards multicast packets it receives from its directly connected LANs according to the mode setting.

In populating the multicast routing table, dense-mode interfaces are always added to the table. Sparse-mode interfaces are added to the table only when periodic join messages are received from downstream devices or when there is a directly connected member on the interface. When forwarding from a LAN, sparse-mode operation occurs if there is an RP 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. The multicast source address must be on the directly connected incoming interface (that is part of the same subnet) of the first-hop router for both PIM dense mode and PIM any-source multicast mode. If the multicast traffic from a specific source is sufficient, the receiver's first-hop router might send join messages toward the source to build a source-based distribution tree.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing**
4. **interface** *interface-id*
5. **ip pim** {**dense-mode** | **sparse-mode** | **sparse-dense-mode**}
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable**	Enables privileged EXEC mode.
	Example:	Enter your password, if prompted.
	Device> **enable**	
Step 2	**configure terminal**	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	**ip multicast-routing**	Enables IP multicast routing.
	Example:	IP multicast routing is supported with Multicast Forwarding Information Base (MFIB) and Multicast Routing Information Base (MRIB).
	Device(config)# **ip multicast-routing**	
Step 4	**interface** *interface-id*	Specifies the Layer 3 interface on which you want to enable multicast routing, and enters interface configuration mode.
	Example:	The specified interface must be one of the following:
	Device(config)# **interface gigabitethernet 1/0/1**	• A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. You will also need to enable IP PIM sparse-dense-mode on the interface, and join the interface as a statically connected member to an IGMP static group.
		• An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. You will also need to enable IP PIM sparse-dense-mode on the VLAN, join the VLAN as a statically connected member to an IGMP static group, and then enable IGMP snooping on the VLAN, the IGMP static group, and physical interface.
		These interfaces must have IP addresses assigned to them.
Step 5	**ip pim** {**dense-mode** | **sparse-mode** | **sparse-dense-mode**}	Enables a PIM mode on the interface.
	Example:	By default, no mode is configured.
		The keywords have these meanings:
	Device(config-if)# **ip pim sparse-dense-mode**	• **dense-mode**—Enables dense mode of operation.
		• **sparse-mode**—Enables sparse mode of operation. If you configure sparse mode, you must also configure an RP.

	Command or Action	Purpose
		• **sparse-dense-mode**—Causes the interface to be treated in the mode in which the group belongs. Sparse-dense mode is the recommended setting. **Note** To disable PIM on an interface, use the **no ip pim** interface configuration command.
Step 6	**end** **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring IP Multicast Forwarding

You can use the following procedure to configure IPv4 Multicast Forwarding Information Base (MFIB) interrupt-level IP multicast forwarding of incoming packets or outgoing packets on the device.

Note After you have enabled IP multicast routing by using the **ip multicast-routing** command, IPv4 multicast forwarding is enabled. Because IPv4 multicast forwarding is enabled by default, you can use the **no** form of the **ip mfib** command to disable IPv4 multicast forwarding.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip mfib**
4. **exit**
5. **show running-config**
6. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip mfib** **Example:** Device(config)# **ip mfib**	Enables IP multicast forwarding.
Step 4	**exit** **Example:** Device(config)# **exit**	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring a Static Multicast Route (mroute)

You can use the following procedure to configure static mroutes. Static mroutes are similar to unicast static routes but differ in the following ways:

- Static mroutes are used to calculate RPF information, not to forward traffic.

- Static mroutes cannot be redistributed.

Static mroutes are strictly local to the device on which they are defined. Because Protocol Independent Multicast (PIM) does not have its own routing protocol, there is no mechanism to distribute static mroutes throughout

the network. Consequently, the administration of static mroutes tends to be more complicated than the administration of unicast static routes.

When static mroutes are configured, they are stored on the device in a separate table referred to as the static mroute table. When configured, the **ip mroute** command enters a static mroute into the static mroute table for the source address or source address range specified for the source-address and mask arguments. Sources that match the source address or that fall in the source address range specified for the source-address argument will RPF to either the interface associated with the IP address specified for the *rpf-address* argument or the local interface on the device specified for the *interface-type* and *interface-number* arguments. If an IP address is specified for the *rpf-address* argument, a recursive lookup is done from the unicast routing table on this address to find the directly connected neighbor.

If there are multiple static mroutes configured, the device performs a longest-match lookup of the mroute table. When the mroute with the longest match (of the source-address) is found, the search terminates and the information in the matching static mroute is used. The order in which the static mroutes are configured is not important.

The administrative distance of an mroute may be specified for the optional distance argument. If a value is not specified for the distance argument, the distance of the mroute defaults to zero. If the static mroute has the same distance as another RPF source, the static mroute will take precedence. There are only two exceptions to this rule: directly connected routes and the default unicast route.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip mroute** [**vrf** *vrf-name*] *source-address mask* { **fallback-lookup** {**global** | **vrf** *vrf-name* }[*protocol*] {*rpf-address* | *interface-type interface-number*}} [**distance**]
4. **exit**
5. **show running-config**
6. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip mroute** [**vrf** *vrf-name*] *source-address mask* { **fallback-lookup** {**global**	**vrf** *vrf-name* }[*protocol*] {*rpf-address*	*interface-type interface-number*}} [**distance**] **Example:**	The source IP address 10.1.1.1 is configured to be reachable through the interface associated with IP address 10.2.2.2.

	Command or Action	Purpose
	Device(configure)# **ip mroute 10.1.1.1 255.255.255.255 10.2.2.2**	
Step 4	**exit** **Example:** Device(config)# **exit**	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** Device# **show running-config**	(Optional) Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring Optional IP Multicast Routing Features

Defining the IP Multicast Boundary

You define a multicast boundary to prevent Auto-RP messages from entering the PIM domain. You create an access list to deny packets destined for 224.0.1.39 and 224.0.1.40, which carry Auto-RP information.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **deny** *source* [*source-wildcard*]
4. **interface** *interface-id*
5. **ip multicast boundary** *access-list-number*
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

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	**configure terminal** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **deny** *source* [*source-wildcard*] **Example:** Device(config)# **access-list 12 deny 224.0.1.39** **access-list 12 deny 224.0.1.40**	Creates a standard access list, repeating the command as many times as necessary. • For *access-list-number*, the range is 1 to 99. • The **deny** keyword denies access if the conditions are matched. • For *source*, enter multicast addresses 224.0.1.39 and 224.0.1.40, which carry Auto-RP information. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. The access list is always terminated by an implicit deny statement for everything.
Step 4	**interface** *interface-id* **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specifies the interface to be configured, and enters interface configuration mode. The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. These interfaces must have IP addresses assigned to them.
Step 5	**ip multicast boundary** *access-list-number* **Example:** Device(config-if)# **ip multicast boundary 12**	Configures the boundary, specifying the access list you created in Step 2.
Step 6	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config)# **end**	
Step 7	show running-config **Example:** Device# **show running-config**	Verifies your entries.
Step 8	copy running-config startup-config **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring sdr Listener Support

Enabling sdr Listener Support

By default, the device does not listen to session directory advertisements.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip sap listen**
5. **end**
6. **show running-config**
7. **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 **Example:** Device# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *interface-id* **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specifies the interface to be enabled for sdr, and enters interface configuration mode. The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. You will also need to enable IP PIM sparse-dense-mode on the interface, and join the interface as a statically connected member to an IGMP static group. For a configuration example, see Example: Interface Configuration as a Routed Port, on page 132 • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. You will also need to enable IP PIM sparse-dense-mode on the VLAN, join the VLAN as a statically connected member to an IGMP static group, and then enable IGMP snooping on the VLAN, the IGMP static group, and physical interface. For a configuration example, see Example: Interface Configuration as an SVI, on page 132 These interfaces must have IP addresses assigned to them.
Step 4	**ip sap listen** **Example:** Device(config-if)# **ip sap listen**	Enables the device software to listen to session directory announcements.
Step 5	**end** **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Limiting How Long an sdr Cache Entry Exists

By default, entries are never deleted from the sdr cache. You can limit how long the entry remains active so that if a source stops advertising SAP information, old advertisements are not unnecessarily kept.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sap cache-timeout** *minutes*
4. **end**
5. **show running-config**
6. **show ip sap**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable**	Enables privileged EXEC mode.
	Example:	Enter your password, if prompted.
	Device> **enable**	
Step 2	**configure terminal**	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	**ip sap cache-timeout** *minutes*	Limits how long a Session Announcement Protocol (SAP)
	Example:	cache entry stays active in the cache.
		By default, entries are never deleted from the cache.
	Device(config)# **ip sap cache-timeout 30**	For *minutes*, the range is 1 to 1440 minutes (24 hours).
Step 4	**end**	Returns to privileged EXEC mode.
	Example:	
	Device(config)# **end**	
Step 5	**show running-config**	Verifies your entries.
	Example:	
	Device# **show running-config**	

	Command or Action	Purpose
Step 6	**show ip sap** **Example:** Device# **show ip sap**	Displays the SAP cache.
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Monitoring and Maintaining Basic IP Multicast Routing

Clearing Caches, Tables, and Databases

You can remove all contents of a particular cache, table, or database. Clearing a cache, table, or database might be necessary when the contents of the particular structure are or suspected to be invalid.

You can use any of the privileged EXEC commands in the following table to clear IP multicast caches, tables, and databases.

Table 3: Commands for Clearing Caches, Tables, and Databases

Command	Purpose
clear ip igmp group {**group** [*hostname* | *IP address*] | **vrf** *name* **group** [*hostname* | *IP address*] }	Deletes entries from the IGMP cache.
clear ip mfib { **counters** [*group* | *source*] | **global counters** [*group* | *source*] | **vrf** * }	Clears all active IPv4 Multicast Forwarding Information Base (MFIB) traffic counters.
clear ip mrm {**status-report** [*source*] }	IP multicast routing clear commands.
clear ip mroute { * | [*hostname* | *IP address*] | **vrf** *name* **group** [*hostname* | *IP address*] }	Deletes entries from the IP multicast routing table.
clear ip msdp { **peer** | **sa-cache** | **statistics** | **vrf** }	Clears the Multicast Source Discovery Protocol (MSDP) cache.
clear ip multicast { **limit** | **redundancy statistics** }	Clears the IP multicast information.
clear ip pim { **df** [**int** | **rp** *rp address*] | **interface** | **rp-mapping** [*rp address*] | **vrf** *vpn name* { **df** | **interface** | **rp-mapping** }	Clears the PIM cache.
clear ip sap [*group-address* | **"***session-name***"**]	Deletes the Session Directory Protocol Version 2 cache or an sdr cache entry.

Displaying System and Network Statistics

You can display specific statistics, such as the contents of IP routing tables, caches, and databases.

Note This release does not support per-route statistics.

You can display information to learn resource usage and solve network problems. You can also display information about node reachability and discover the routing path that packets of your device are taking through the network.

You can use any of the privileged EXEC commands in the following table to display various routing statistics.

Table 4: Commands for Displaying System and Network Statistics

Command	Purpose
ping [*group-name*	*group-address*]	Sends an ICMP Echo Request to a multicast group address.
show ip igmp filter	Displays IGMP filter information.
show ip igmp groups [*group-name*|*group-address*|*type-number*]	Displays the multicast groups that are directly connected to the device and that were learned through IGMP.
show ip igmp interface [*type number*]	Displays multicast-related information about an interface.
show ip igmp profile [profile_number]	Displays IGMP profile information.
show ip igmp ssm-mapping [*hostname/IP address*]	Displays IGMP SSM mapping information.
show ip igmp static-group {**class-map** [**interface** [type]]	Displays static group information.
show ip igmp membership [*name/group address*	**all**	**tracked**]	Displays IGMP membership information for forwarding.
show ip igmp vrf	Displays the selected VPN Routing/Forwarding instance by name.
show ip mfib [*type number*]	Displays the IP multicast forwarding information base.
show ip mrib { **client**	**route**	**vrf** }	Displays the multicast routing information base.
show ip mrm { **interface**	**manager**	**status-report** }	Displays the IP multicast routing monitor information.
show ip mroute [*group-name*	*group-address*] [*source*] [**count**	**interface**	**proxy**	**pruned**	**summary**	**verbose**]	Displays the contents of the IP multicast routing table.

Command	Purpose
show ip msdp { **count** | **peer** | **rpf-peer** | **sa-cache** | **summary** | **vrf** }	Displays the Multicast Source Discovery Protocol (MSDP) information.
show ip multicast [**interface** | **limit** | **mpls** | **redundancy** | **vrf**]	Displays global multicast information.
show ip pim all-vrfs { **tunnel** }	Display all VRFs.
show ip pim autorp	Display global auto-RP information.
show ip pim boundary [*type number*]	Displays boundary information.
show ip pim bsr-router	Display bootstrap router information (version 2).
show ip pim interface [*type number*] [**count** | **detail** | **df** | **stats**]	Displays information about interfaces configured for PIM. This command is available in all software images.
show ip pim neighbor [*type number*]	Lists the PIM neighbors discovered by the device. This command is available in all software images.
show ip pim mdt [**bgp**]	Displays multicast tunnel information.
show ip pim rp [*group-name* | *group-address*]	Displays the RP routers associated with a sparse-mode multicast group. This command is available in all software images.
show ip pim rp-hash [*group-name* | *group-address*]	Displays the RP to be chosen based upon the group selected.
show ip pim tunnel [*tunnel* | *verbose*]	Displays the registered tunnels.
show ip pim vrf *name*	Displays VPN routing and forwarding instances.
show ip rpf {*source-address* | *name*}	Displays how the device is doing Reverse-Path Forwarding (that is, from the unicast routing table, DVMRP routing table, or static mroutes). Command parameters include: • *Host name* or *IP address*—IP name or group address. • **Select**—Group-based VRF select information. • **vrf**—Selects VPN Routing/Forwarding instance.

Command	Purpose
show ip sap [*group* | **"***session-name***"** | **detail**]	Displays the Session Announcement Protocol (SAP) Version 2 cache. Command parameters include: • *A.B.C.D*—IP group address. • *WORD*—Session name (in double quotes). • **detail**—Session details.

Displaying Multicast Peers, Packet Rates and Loss Information, and Path Tracing

You can use the privileged EXEC commands in the following table to monitor IP multicast routers, packets, and paths.

Table 5: Commands for Displaying Multicast Peers, Packet Rates and Loss Information, and Path Tracing

Command	Purpose
mrinfo { [*hostname* | *address*] | **vrf** }	Queries a multicast router or multilayer device about which neighboring multicast devices are peering with it.
mstat { [*hostname* | *address*] | **vrf** }	Displays IP multicast packet rate and information loss.

Configuration Examples for IP Multicast Routing

Example: Configuring an IP Multicast Boundary

This example shows how to set up a boundary for all administratively-scoped addresses:

```
Device(config)# access-list 1 deny 239.0.0.0 0.255.255.255
Device(config)# access-list 1 permit 224.0.0.0 15.255.255.255
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip multicast boundary 1
```

Example: Responding to mrinfo Requests

The software answers mrinfo requests sent by mrouted systems and Cisco routers and multilayer devices. The software returns information about neighbors through DVMRP tunnels and all the routed interfaces. This information includes the metric (always set to 1), the configured TTL threshold, the status of the interface, and various flags. You can also use the **mrinfo** privileged EXEC command to query the router or device itself, as in this example:

```
Device# mrinfo
  171.69.214.27 (mm1-7kd.cisco.com) [version cisco 11.1] [flags: PMS]:
```

```
171.69.214.27 -> 171.69.214.26 (mm1-r7kb.cisco.com) [1/0/pim/querier]
171.69.214.27 -> 171.69.214.25 (mm1-45a.cisco.com) [1/0/pim/querier]
171.69.214.33 -> 171.69.214.34 (mm1-45c.cisco.com) [1/0/pim]
171.69.214.137 -> 0.0.0.0 [1/0/pim/querier/down/leaf]
171.69.214.203 -> 0.0.0.0 [1/0/pim/querier/down/leaf]
171.69.214.18 -> 171.69.214.20 (mm1-45e.cisco.com) [1/0/pim]
171.69.214.18 -> 171.69.214.19 (mm1-45c.cisco.com) [1/0/pim]
171.69.214.18 -> 171.69.214.17 (mm1-45a.cisco.com) [1/0/pim]
```

Additional References for IP Multicast

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	

Standards and RFCs

Standard/RFC	Title
RFC 1112	*Host Extensions for IP Multicasting*
RFC 2236	*Internet Group Management Protocol, Version 2*
RFC 4601	*Protocol-Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	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 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 History and Information for IP Multicast

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.

Constraining IP Multicast in Switched Ethernet

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.
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.

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 for IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	

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 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 History and Information for Constraining IP Multicast in a Switched Ethernet Network

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.

CHAPTER 4

Configuring Multicast Routing over GRE Tunnel

Prerequisites for Configuring Multicast Routing over GRE Tunnel

Before configuring multicast routing over GRE, you should be familiar with the concepts of IP Multicast Routing Technology and GRE Tunneling.

Restrictions for Configuring Multicast Routing over GRE Tunnel

The following are the restrictions for configuring multicast routing over GRE tunnel:

- IPv6 multicast over GRE tunnel is not supported.

- The total number of supported multicast routes (mroutes) is 32000, across all tunnels.

  Use the formula 8000/(((Number of tunnels)/4) + 1) to derive the number of mroutes.

- Bidirectional PIM is not supported.

- Multicast routing should be configured on the first hop router (FHR), the rendezvous point (RP) and the last hop router (LHR) to support multicast over the GRE tunnel.

- On Catalyst 3650 series switches, the tunnel source can be a loopback, physical, or L3 EtherChannel interface.

- No feature interactions such as IPSec, ACL, Tunnel counters, Crypto support, Fragmentation, Cisco Discovery Protocol (CDP), QoS, GRE keepalive, Multipoint GRE, etc. are supported on the GRE Tunnel.

Information About Multicast Routing over GRE Tunnel

This chapter 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. Multicast Routing over GRE Tunnel supports sparse mode and pim-ssm mode; and supports static RP and auto-RP. See Rendevous Point and Auto-RP for information on configuring static RP and auto-RP.

Note   Beginning in Cisco IOS XE Denali 16.3.1, multicast routing and NHRP are supported with GRE Tunneling. NHRP can optionally be configured along with the multicast configuration on the tunnel interface to facilitate dynamic discovery of tunnel end points. Please see NHRP for configuring NHRP on a tunnel interface.

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.

How to Configure Multicast Routing over GRE Tunnel

Configuring a GRE Tunnel to Connect Non-IP Multicast Areas

You can configure a GRE tunnel to transport IP multicast packets between a source and destination that are connected by a medium that does not support multicast routing.

SUMMARY STEPS

1.   **enable**
2.   **configure terminal**
3.   **ip multicast-routing**
4.   **interface tunnel** *number*
5.   **ip address ip_address** *subnet_mask*
6.   **ip pim sparse-mode**
7.   **tunnel source** { *ip-address* | *interface-name* }
8.   **tunnel destination** { *hostname* | *ip-address* }
9.   **end**
10. **show interface** *type 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	**ip multicast-routing** **Example:** Device(config)# ip multicast-routing	Enables IP multicast routing.
Step 4	**interface tunnel** *number* **Example:** Device(config)# interface tunnel 0	Enters tunnel interface configuration mode.
Step 5	**ip address ip_address** *subnet_mask* **Example:** Device(config-if)# ip address 192.168.24.1 255.255.255.252	Configures IP address and IP subnet.
Step 6	**ip pim sparse-mode** **Example:** Device(config-if)# ip pim sparse-mode	Enables sparse mode of operation of Protocol Independent Multicast (PIM) on the tunnel interface with one of the following mode of operation:
Step 7	**tunnel source** { *ip-address* | *interface-name* } **Example:** Device(config-if)# tunnel source 100.1.1.1	Configures the tunnel source.
Step 8	**tunnel destination** { *hostname* | *ip-address* } **Example:** Device(config-if)# tunnel destination 100.1.5.3	Configures the tunnel destination.
Step 9	**end** **Example:** Device(config-if)# end	Ends the current configuration session and returns to privileged EXEC mode.
Step 10	**show interface** *type number* **Example:** Device# show interface tunnel 0	Displays tunnel interface information.

Tunneling to Connect Non-IP Multicast Areas Example

The following example shows multicast-routing between a Catalyst 3650/3850 switch through a GRE tunnel.

Figure 3: Tunnel Connecting Non-IP Multicast Areas
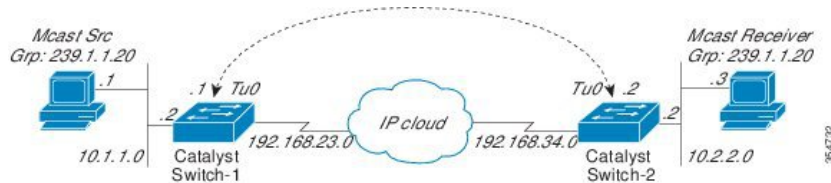
In the figure above, the multicast source (10.1.1.1) is connected to Catalyst 3650/3850 Switch-1 and is configured for multicast group 239.1.1.20. The multicast receiver (10.2.2.3) is connected to Catalyst 3650/3850 Switch-2 and is configured to receive multicast packets for group 239.1.1.20. Separating Switch-1 and Switch-2 is an IP cloud, which is not configured for multicast routing.

A GRE tunnel is configured between Switch-1 to Switch-2 sourced with their loopback interfaces. Multicast-routing is enabled on Switch-1 and Switch-2. The **ip pim sparse-mode** command is configured on tunnel interfaces to support PIM in the sparse mode. Sparse mode configuration on the tunnel interfaces allows sparse-mode packets to be forwarded over the tunnel depending on rendezvous point (RP) configuration for the group.

Switch-1 Configuration:

```
Device(config)# ip multicast-routing
Device(config)# interface Loopback0   //Tunnel source interface
Device(config-if)# ip address 2.2.2.2 255.255.255.255

Device(config)# interface Tunnel 10    //Tunnel interface configured for PIM
traffic
Device(config-if)# ip address 192.168.24.1 255.255.255.252
Device(config-if)# ip pim sparse-mode
Device(config-if)# ip nhrp map 192.168.24.3  4.4.4.4  //NHRP may optionally be
configured to dynamically discover tunnel end points.
Device(config-if)# ip nhrp map multicast 4.4.4.4
Device(config-if)# ip nhrp network-id 1
Device(config-if)# ip nhrp nhs 192.168.24.3
Device(config-if)# tunnel source Loopback0
Device(config-if)# tunnel destination 4.4.4.4

Device(config)# interface GigabitEthernet 0/0/0      //Source interface
Device(config-if)# ip address 10.1.1.2 255.255.255.0
Device(config-if)# ip pim sparse-mode
```

Switch-2 Configuration:

```
Device(config)# ip multicast-routing
Device(config)# interface Loopback0    //Tunnel source interface
Device(config-if)# ip address 4.4.4.4 255.255.255.255

Device(config)# interface Tunnel 10     //Tunnel interface configured for PIM
traffic
Device(config-if)# ip address 192.168.24.2 255.255.255.252
```

```
Device(config-if)# ip nhrp map 192.168.24.4 2.2.2.2  //NHRP may optionally be
configured to dynamically discover tunnel end points.
Device(config-if)# ip nhrp map multicast 2.2.2.2
Device(config-if)# ip nhrp network-id 1
Device(config-if)# ip nhrp nhs 192.168.24.4
Device(config-if)# ip pim sparse-mode
Device(config-if)# tunnel source Loopback0
Device(config-if)# tunnel destination 2.2.2.2

Device(config)# interface GigabitEthernet 0/0/0     //Receiver interface
Device(config-if)# ip address 10.2.2.2 255.255.255.0
Device(config-if)# ip pim sparse-mode
```

Additional References

The following sections provide references related to customizing IGMP.

Standards and RFCs

Standard/RFC	Title
None	N/A

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

Configuring Multicast Routing over GRE Tunnel

█ **Feature History and Information for IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment**

Feature History and Information for IP Multicast Optimization: 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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

Feature Name	Releases	Feature Information
IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment	Cisco IOS XE 3.3SE	This feature was implemented on the following platforms:

CHAPTER 5

Configuring VRF-lite

Information About VRF-lite

VRF-lite is a feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs. VRF-lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be either physical, such as Ethernet ports, or logical, such as VLAN SVIs, but a Layer 3 interface cannot belong to more than one VRF at any time.

Note VRF-lite interfaces must be Layer 3 interfaces.

VRF-lite includes these devices:

- Customer edge (CE) devices provide customer access to the service provider network over a data link to one or more provider edge routers. The CE device advertises the site's local routes to the provider edge router and learns the remote VPN routes from it. A Cisco Catalyst Switch can be a CE.

- Provider edge (PE) routers exchange routing information with CE devices by using static routing or a routing protocol such as BGP, RIPv1, or RIPv2.

  The PE is only required to maintain VPN routes for those VPNs to which it is directly attached, eliminating the need for the PE to maintain all of the service provider VPN routes. Each PE router maintains a VRF for each of its directly connected sites. Multiple interfaces on a PE router can be associated with a single VRF if all of these sites participate in the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CEs, a PE router exchanges VPN routing information with other PE routers by using internal BGP (iBGP).

• Provider routers (or core routers) are any routers in the service provider network that do not attach to CE devices.

With VRF-lite, multiple customers can share one CE, and only one physical link is used between the CE and the PE. The shared CE maintains separate VRF tables for each customer and switches or routes packets for each customer based on its own routing table. VRF-lite extends limited PE functionality to a CE device, giving it the ability to maintain separate VRF tables to extend the privacy and security of a VPN to the branch office.

The following figure displays a configuration where each Cisco Catalyst switch acts as multiple virtual CEs. Because VRF-lite is a Layer 3 feature, each interface in a VRF must be a Layer 3 interface.

Figure 4: Cisco Catalyst Switches Acting as Multiple Virtual CEs
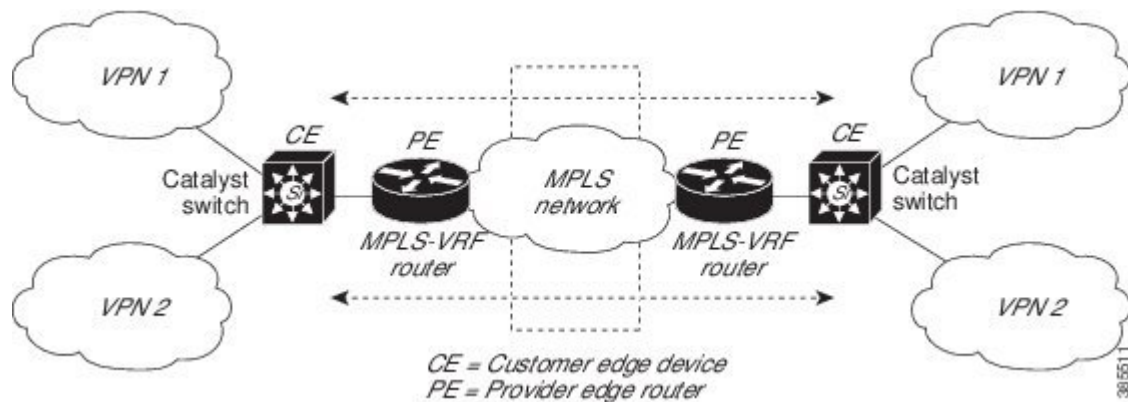
This figure illustrates the packet-forwarding process in a VRF-lite CE-enabled network.

• When the CE receives a packet from a VPN, it looks up the routing table based on the input interface. When a route is found, the CE forwards the packet to the PE.

• When the ingress PE receives a packet from the CE, it performs a VRF lookup. When a route is found, the router adds a corresponding MPLS label to the packet and sends it to the MPLS network.

• When an egress PE receives a packet from the network, it strips the label and uses the label to identify the correct VPN routing table. The egress PE then performs the normal route lookup. When a route is found, it forwards the packet to the correct adjacency.

• When a CE receives a packet from an egress PE, it uses the input interface to look up the correct VPN routing table. If a route is found, the CE forwards the packet within the VPN.

To configure VRF, create a VRF table and specify the Layer 3 interface associated with the VRF. You then configure the routing protocols in the VPN and between the CE and the PE. BGP is the preferred routing protocol used to distribute VPN routing information across the providers' backbone. The VRF-lite network has three major components:

• VPN route target communities—Lists all other members of a VPN community. You need to configure VPN route targets for each VPN community member.

• Multiprotocol BGP peering of VPN community PE routers—Propagates VRF reachability information to all members of a VPN community. You need to configure BGP peering in all PE routers within a VPN community.

• VPN forwarding—Transports all traffic between all VPN community members across a VPN service-provider network.

Guidelines for Configuring VRF-lite

IPv4 and IPv6

- A switch with VRF-lite is shared by multiple customers, and all customers have their own routing tables.

- Because customers use different VRF tables, you can reuse the same IP addresses. Overlapped IP addresses are allowed in different VPNs.

- VRF-lite lets multiple customers share the same physical link between the PE and the CE. Trunk ports with multiple VLANs separate packets among customers. All customers have their own VLANs.

- For the PE router, there is no difference between using VRF-lite or using multiple CEs. In #unique_69, multiple virtual Layer 3 interfaces are connected to the VRF-lite device.

- The Cisco Catalyst switch supports configuring VRF by using physical ports, VLAN SVIs, or a combination of both. You can connect SVIs through an access port or a trunk port.

- A customer can use multiple VLANs as long because they do not overlap with those of other customers. A customer's VLANs are mapped to a specific routing table ID that is used to identify the appropriate routing tables stored on the switch.

- The Layer 3 TCAM resource is shared between all VRFs. To ensure that any one VRF has sufficient CAM space, use the **maximum routes** command.

- A Cisco Catalyst switch using VRF can support one global network and multiple VRFs. The total number of routes supported is limited by the size of the TCAM.

- A single VRF can be configured for both IPv4 and IPv6.

- If an incoming packet's destination address is not found in the vrf table, the packet is dropped. Also, if insufficient TCAM space exists for a VRF route, hardware switching for that VRF is disabled and the corresponding data packets are sent to software for processing.

IPv4 Specific

- You can use most routing protocols (BGP, OSPF, EIGRP, RIP and static routing) between the CE and the PE. However, we recommend using external BGP (EBGP) for these reasons:

    - BGP does not require multiple algorithms to communicate with multiple CEs.

    - BGP is designed for passing routing information between systems run by different administrations.

    - BGP makes simplifies passing attributes of the routes to the CE.

- The Cisco Catalyst switch supports PIM-SM and PIM-SSM protocols.

- The **capability vrf-lite** subcommand under **router ospf** should be used when configuring OSPF as the routing protocol between the PE and the CE.

IPv6 specific

- VRF-aware OSPFv3, BGPv6, EIGRPv6, and IPv6 static routing are supported.

• VRF-aware IPv6 route applications include: ping, telnet, ssh, tftp, ftp and traceroute. (This list does not include the Mgt interface, which is handled differently even though you can configure both IPv4 or IPv6 VRF under it.)

Topic 2.1

How to Configure VRF-lite

Configuring VRF-lite for IPv4

Configuring VRF-Aware Services

IP services can be configured on global interfaces and within the global routing instance. IP services are enhanced to run on multiple routing instances; they are VRF-aware. Any configured VRF in the system can be specified for a VRF-aware service.

VRF-aware services are implemented in platform-independent modules. VRF provides multiple routing instances in Cisco IOS. Each platform has its own limit on the number of VRFs it supports.

VRF-aware services have the following characteristics:

• The user can ping a host in a user-specified VRF.

• ARP entries are learned in separate VRFs. The user can display Address Resolution Protocol (ARP) entries for specific VRFs.

Configuring the User Interface for ARP

Procedure

	Command or Action	Purpose
Step 1	**show ip arp vrf** *vrf-name* **Example:** `Switch# show ip arp vrf vrf-name`	Displays the ARP table (static and dynamic entries) in the specified VRF.

Step 2	**arp vrf** *vrf-name ip-address mac-address ARPA*
	
	Example:
	`Switch(config)# arp vrf vrf-name ip-address`
	`mac-address ARPA`
		Creates a static ARP entry in the specified VRF.

Configuring Per-VRF for TACACS+ Servers

The per-VRF for TACACS+ servers feature enables you to configure per-virtual route forwarding (per-VRF) authentication, authorization, and accounting (AAA) on TACACS+ servers.

You can create the VRF routing table (shown in Steps 3 and 4) and configure the interface (Steps 6, 7, and 8). The actual configuration of per-VRF on a TACACS+ server is done in Steps 10 through 13.

Before you begin

Before configuring per-VRF on a TACACS+ server, you must have configured AAA and a server group.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** Switch# configure terminal	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** Switch(config)# ip vrf vrf-name	Configures a VRF table and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** Switch (config-vrf)# rd route-distinguisher	Creates routing and forwarding tables for a VRF instance.
Step 5	**exit** **Example:** Switch (config-vrf)# exit	Exits VRF configuration mode.
Step 6	**interface** *interface-name* **Example:** Switch (config)# interface interface-name	Configures an interface and enters interface configuration mode.
Step 7	**vrf forwarding** *vrf-name* **Example:** Switch (config-if)# vrf forwarding vrf-name	Configures a VRF for the interface.
Step 8	**ip address** *ip-address mask* [**secondary**] **Example:** Switch (config-if)# ip address ip-address mask [secondary]	Sets a primary or secondary IP address for an interface.
Step 9	**exit** **Example:** Switch (config-vrf)# exit	Exits interface configuration mode.
Step 10	**aaa group server tacacs+** *group-name* **Example:** Switch (config)# aaa group server tacacs+ tacacs1	Groups different TACACS+ server hosts into distinct lists and distinct methods and enters server-group configuration mode.

	Command or Action	Purpose
Step 11	**server-private** {*ip-address*	*name*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0**	**7**] *string*] **Example:** `Switch (config-sg-tacacs+)# server-private 10.1.1.1 port 19 key cisco`	Configures the IP address of the private TACACS+ server for the group server.
Step 12	**vrf forwarding** *vrf-name* **Example:** `Switch (config-sg-tacacs+)# vrf forwarding vrf-name`	Configures the VRF reference of a AAA TACACS+ server group.
Step 13	**ip tacacs source-interface** *subinterface-name* **Example:** `Switch (config-sg-tacacs+)# ip tacacs source-interface subinterface-name`	Uses the IP address of a specified interface for all outgoing TACACS+ packets.
Step 14	**exit** **Example:** `Switch (config-sg-tacacs)# exit`	Exits server-group configuration mode.

Example

The following example lists all the steps to configure per-VRF TACACS+:

```
Switch> enable
Switch# configure terminal
Switch (config)# ip vrf cisco
Switch (config-vrf)# rd 100:1
Switch (config-vrf)# exit
Switch (config)# interface Loopback0
Switch (config-if)# vrf forwarding cisco
Switch (config-if)# ip address 10.0.0.2 255.0.0.0
Switch (config-if)# exit
Switch (config-sg-tacacs+)# vrf forwarding cisco
Switch (config-sg-tacacs+)# ip tacacs source-interface Loopback0
Switch (config-sg-tacacs)# exit
```

Configuring Multicast VRFs

SUMMARY STEPS

1. **configure terminal**
2. **ip routing**
3. **ip vrf** *vrf-name*
4. **ip multicast-routing vrf** *vrf-name*
5. **rd** *route-distinguisher*
6. **route-target** {**export** | **import** | **both**} *route-target-ext-community*

7. **import map** *route-map*

8. **interface** *interface-id*

9. **vrf forwarding** *vrf-name*

10. **ip address** *ip-address* **mask**

11. **ip pim sparse-mode**

12. **end**

13. show ip vrf [brief | detail | interfaces] [vrf-name]

14. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Switch# configure terminal	Enters global configuration mode.
Step 2	**ip routing** **Example:** Switch(config)# ip routing	Enables IP routing.
Step 3	**ip vrf** *vrf-name* **Example:** Switch(config)# ip vrf vrf-name	Configures a VRF table and enters VRF configuration mode.
Step 4	**ip multicast-routing vrf** *vrf-name* **Example:** Switch(config-vrf)# ip multicast-routing vrf vrf-name	(Optional) Enables global multicast routing for VRF table.
Step 5	**rd** *route-distinguisher* **Example:** Switch (config-vrf)# rd route-distinguisher	Creates a VRF table by specifying a route distinguisher. Enter either an AS number and an arbitrary number (xxx:y) or an IP address and arbitrary number (A.B.C.D:y).
Step 6	**route-target** {**export** | **import** | **both**} *route-target-ext-community* **Example:** Switch(config-vrf)# route-target {export | import | both} route-target-ext-community	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). The route-target-ext-community value should be the same as the route-distinguisher value entered in Step 4.
Step 7	**import map** *route-map* **Example:** Switch(config-vrf)# import map route-map	(Optional) Associates a route map with the VRF.

	Command or Action	Purpose
Step 8	**interface** *interface-id* **Example:** `Switch (config)# interface interface-id`	Enters interface configuration mode and specifies the Layer 3 interface to be associated with the VRF. The interface can be a routed port or a SVI.

Step 9	**vrf forwarding** *vrf-name*
	
	Example:
	
	`Switch (config-sg-tacacs+)# vrf forwarding`
	`vrf-name`
		Associates the VRF with the Layer 3 interface.

Step 10	**ip address** *ip-address* **mask** **Example:** `Switch (config-if)# ip address ip-address mask`	Configures IP address for the Layer 3 interface.
Step 11	**ip pim sparse-mode** **Example:** `Switch(config-if)# ip pim sparse-mode`	Enables PIM on the VRF-associated Layer 3 interface.
Step 12	**end** **Example:** `Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 13	show ip vrf [brief | detail | interfaces] [vrf-name] **Example:** **show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*]	Verifies the configuration. Display information about the configured VRFs.
Step 14	**copy running-config startup-config** **Example:** `Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Example

The following example shows how to configure multicast within a VRF table:

```
Switch(config)# ip routing
Switch(config)# ip vrf multiVrfA
Switch(config-vrf)# ip multicast-routing vrf multiVrfA
Switch(config-vrf)# interface GigabitEthernet3/1/0
Switch(config-if)# vrf forwarding multiVrfA
Switch(config-if)# ip address 172.21.200.203 255.255.255.0
Switch(config-if)# ip pim sparse-mode
```

Configuring a VPN Routing Session

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Switch# configure terminal	Enters global configuration mode.
Step 2	**router ospf** *process-id* **vrf vrf-name** **Example:** Switch(config)# router ospf process-id vrf vrf-name	Enables OSPF routing, specifies a VPN forwarding table, and enters router configuration mode.
Step 3	**log-adjacency-changes** **Example:** Switch(config-router)# log-adjacency-changes	(Optional) Logs changes in the adjacency state (the default state).
Step 4	**redistribute bgp** *autonomous-system-number subnets* **Example:** Switch(config-router)# redistribute bgp autonomous-system-number subnets	Sets the switch to redistribute information from the BGP network to the OSPF network.
Step 5	**network** *network-number* **area** *area-id* **Example:** Switch(config-router)# network network-number area area-id	Defines a network address and mask on which OSPF runs and the area ID for that network address.
Step 6	**end** **Example:** Switch(config-router)# end	Returns to privileged EXEC mode.
Step 7	**show ip ospf** *process-id* **Example:** Switch# show ip ospf process-id	Verifies the configuration of the OSPF network.
Step 8	copy running-config startup-config **Example:** Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file. Use the **no router ospf** *process-id* **vrf** *vrf-name* global configuration command to disassociate the VPN forwarding table from the OSPF routing process.

Example

```
Switch(config)# ip vrf VRF-RED
Switch(config-vrf)# rd 1:1
Switch(config-vrf)# exit
Switch(config)# router eigrp virtual-name
Switch(config-router)# address-family ipv4 vrf VRF-RED autonomous-system 1
```

```
Switch(config-router-af)# network 10.0.0.0 0.0.0.255
Switch(config-router-af)# topology base
Switch(config-router-topology)# default-metric 10000 100 255 1 1500
Switch(config-router-topology)# exit-af-topology
Switch(config-router-af)# exit-address-family
```

Configuring BGP PE to CE Routing Sessions

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** Switch# configure terminal	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system-number* **Example:** Switch(config)# router bgp autonomous-system-number	Configures the BGP routing process with the AS number passed to other BGP routers and enters router configuration mode.
Step 3	**network** *network-number* **mask** *network-mask* **Example:** Switch(config-router)# network network-number mask network-mask	Specifies a network and mask to announce using BGP.
Step 4	**redistribute ospf** *process-id match internal* **Example:** Switch(config-router)# redistribute ospf process-id match internal	Sets the switch to redistribute OSPF internal routes.
Step 5	**network** *network-number* **area** *area-id* **Example:** Switch(config-router)# network network-number area area-id	Defines a network address and mask on which OSPF runs and the area ID for that network address.
Step 6	**address-family ipv4 vrf** *vrf-name* **Example:** Switch(config-router-af)# address-family ipv4 vrf vrf-name	Defines BGP parameters for PE to CE routing sessions and enters VRF address-family mode.
Step 7	**neighbor** *address* **remote-as** *as-number* **Example:** Switch(config-router-af)# neighbor address remote-as as-number	Defines a BGP session between PE and CE routers.
Step 8	**neighbor** *address* **activate** **Example:** Switch(config-router-af)# neighbor address activate	Activates the advertisement of the IPv4 address family.

	Command or Action	Purpose
Step 9	**end** **Example:** `Switch(config-router-af)# end`	Returns to privileged EXEC mode.
Step 10	**show ip bgp** [**ipv4**] [**neighbors**] **Example:** `Switch# show ip bgp [ipv4] [neighbors]`	Verifies BGP configuration. Use the **no router bgp** *autonomous-system-number* global configuration command to delete the BGP routing process. Use the command with keywords to delete routing characteristics.

Configuring IPv4 VRFs

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**ip routing** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Switch(config)# ip vrf vrf-name`	Names the VRF and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Switch(config-vrf)# rd route-distinguisher`	Creates a VRF table by specifying a route distinguisher. Enter either an Autonomous System number number and an arbitrary number (xxx:y) or an IP address and arbitrary number (A.B.C.D:y).
Step 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* **Example:** `Switch(config-vrf)# route-target {export | import | both} route-target-ext-community`	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). **Note** This command is effective only if BGP is running.
Step 6	**import map** *route-map* **Example:** `Switch(config-vrf)# import map route-map`	(Optional) Associates a route map with the VRF.

	Command or Action	Purpose
Step 7	**interface** *interface-id* **Example:** Switch(config-vrf)# interface interface-id	Enters interface configuration mode and specify the Layer 3 interface to be associated with the VRF. The interface can be a routed port or SVI.
Step 8	**vrf forwarding** *vrf-name* **Example:** Switch(config-if)# vrf forwarding vrf-name	Associates the VRF with the Layer 3 interface.
Step 9	**end** **Example:** Switch(config-if)# end	Returns to privileged EXEC mode.
Step 10	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] **Example:** Switch# show ip vrf [brief | detail | interfaces] [vrf-name]	Verifies the configuration. Displays information about the configured VRFs.
Step 11	**copy running-config startup-config** **Example:** Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file. **Note** For complete syntax and usage information for the following commands, see the switch command reference for this release and see the Cisco IOS Switching Services Command Reference. Use the **no ip vrf** *vrf-name* global configuration command to delete a VRF and to remove all interfaces from it. Use the **no vrf forwarding** interface configuration command to remove an interface from the VRF.

Configuring VRF-lite for IPv6

Configuring VRF-Aware Services

IPv6 services can be configured on global interfaces and within the global routing instance. IPv6 services are enhanced to run on multiple routing instances; they are VRF-aware. Any configured VRF in the system can be specified for a VRF-aware service.

VRF-aware services are implemented in platform-independent modules. VRF provides multiple routing instances in Cisco IOS. Each platform has its own limit on the number of VRFs it supports.

VRF-aware services have the following characteristics:

- The user can ping a host in a user-specified VRF.

- Neighbor Discovery entries are learned in separate VRFs. The user can display Neighbor Discovery (ND) entries for specific VRFs.

The following services are VRF-aware:

- Ping

- Unicast Reverse Path Forwarding (uRPF)

- Traceroute

- FTP and TFTP

- Telnet and SSH

- NTP

Configuring the User Interface for PING

Perform the following task to configure a VRF-aware ping:

Procedure

	Command or Action	Purpose
Step 1	**ping vrf** *vrf-name* **ipv6-host** **Example:** `Switch# ping vrf vrf-name ipv6-host`	Pings an IPv6 host or address in the specified VRF.

Configuring the User Interface for uRPF

You can configure uRPF on an interface assigned to a VRF. Source lookup is performed in the VRF table

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **no switchport**
4. **vrf forwarding** *vrf-name*
5. **ipv6 address** *ip-address* **subnet-mask**
6. **ipv6 verify unicast source reachable-via rx allow-default**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Switch (config)# interface interface-id`	Enters interface configuration mode and specifies the Layer 3 interface to configure.

	Command or Action	Purpose
Step 3	**no switchport** **Example:** Switch (config-if)# no switchport	Removes the interface from Layer 2 configuration mode if it is a physical interface.
Step 4	**vrf forwarding** *vrf-name* **Example:** Switch (config-if)# vrf forwarding vrf-name	Configures VRF on the interface.
Step 5	**ipv6 address** *ip-address* **subnet-mask** **Example:** Switch (config-if)# ip address ip-address mask	Enters the IPv6 address for the interface.
Step 6	**ipv6 verify unicast source reachable-via rx allow-default** **Example:** Switch(config-if)# ipv6 verify unicast source reachable-via rx allow-default	Enables uRPF on the interface.
Step 7	**end** **Example:** Switch(config-if)# end	Returns to privileged EXEC mode.

Configuring the User Interface for Traceroute

SUMMARY STEPS

      1. **traceroute vrf** *vrf-name ipv6address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**traceroute vrf** *vrf-name ipv6address* **Example:** Switch# traceroute vrf vrf-name ipv6address	Specifies the name of a VPN VRF in which to find the destination address.

Configuring the User Interface for Telnet and SSH

SUMMARY STEPS

      1. **telnet** *ipv6-address*/**vrf** *vrf-name*
      2. **ssh -l** *username* **-vrf** *vrf-name* **ipv6-host**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**telnet** *ipv6-address*/**vrf** *vrf-name* **Example:** Switch# telnet ipv6-address/vrf vrf-name	Connects through Telnet to an IPv6 host or address in the specified VRF.
Step 2	**ssh -l** *username* **-vrf** *vrf-name* **ipv6-host** **Example:** Switch# ssh -l username -vrf vrf-name ipv6-host	Connects through SSH to an IPv6 host or address in the specified VRF.

Configuring the User Interface for NTP

SUMMARY STEPS

1. **configure terminal**
2. **ntp server vrf** *vrf-name* **ipv6-host**
3. **ntp peer vrf** *vrf-name* **ipv6-host**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 2	**ntp server vrf** *vrf-name* **ipv6-host** **Example:** Device(config)# **ntp server vrf vrf-name ipv6-host**	Configure the NTP server in the specified VRF.
Step 3	**ntp peer vrf** *vrf-name* **ipv6-host** **Example:** Device(config)# ntp peer vrf vrf-name ipv6-host	Configure the NTP peer in the specified VRF.

Configuring IPv6 VRFs

SUMMARY STEPS

1. **configure terminal**
2. **vrf definition** *vrf-name*
3. **rd** *route-distinguisher*
4. **address-family** *ipv4* | *ipv6*
5. **route-target** {**export** | **import** | **both**} *route-target-ext-community*

6. **exit-address-family**
7. **vrf definition** *vrf-name*
8. **ipv6 multicast multitopology**
9. **address-family ipv6 multicast**
10. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** Switch# configure terminal	Enters global configuration mode.
Step 2	**vrf definition** *vrf-name* **Example:** Switch(config)# vrf definition vrf-name	Names the VRF and enters VRF configuration mode.
Step 3	**rd** *route-distinguisher* **Example:** Switch(config-vrf)# rd route-distinguisher	(Optional) Creates a VRF table by specifying a route distinguisher. Enter either an Autonomous System number and an arbitrary number (xxx:y) or an IP address and arbitrary number (A.B.C.D:y).
Step 4	**address-family** *ipv4* | *ipv6* **Example:** Switch(config-vrf)# address-family ipv4 | ipv6	(Optional) IPv4 by default. Configuration MUST for IPv6.
Step 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* **Example:** Switch(config-vrf)# route-target {export | import | both} route-target-ext-community	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). **Note** This command is effective only if BGP is running.
Step 6	**exit-address-family** **Example:** Switch(config-vrf)# exit-address-family	Exits VRF address-family configuration mode and return to VRF configuration mode.
Step 7	**vrf definition** *vrf-name* **Example:** Switch(config)# vrf definition vrf-name	Enters VRF configuration mode.
Step 8	**ipv6 multicast multitopology** **Example:** Switch(config-vrf-af)# ipv6 multicast multitopology	Enables multicast specific RPF topology.

	Command or Action	Purpose
Step 9	**address-family ipv6 multicast** **Example:** `Switch(config-vrf)# address-family ipv6 multicast`	Enter multicast IPv6 address-family.
Step 10	**end** **Example:** `Switch(config-vrf-af)# end`	Returns to privileged EXEC mode.

Example

This example shows how to configure VRFs:

```
Switch(config)#  vrf definition red
Switch(config-vrf)# rd 100:1
Switch(config-vrf)# address family ipv6
Switch(config-vrf-af)# route-target both 200:1
Switch(config-vrf)# exit-address-family
Switch(config-vrf)# vrf definition red
Switch(config-if)# ipv6 multicast multitopology
Switch(config-if)# address-family ipv6 multicast
Switch(config-vrf-af)# end
Switch#
```

Associating Interfaces to the Defined VRFs

SUMMARY STEPS

1. **interface** *interface-id*
2. **no switchport**
3. **vrf forwarding** *vrf-name*
4. **ipv6 enable**
5. **ipv6 address** *ip-address subnet-mask*
6. **show ipv6 vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*]
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *interface-id* **Example:** `Switch(config-vrf)# interface interface-id`	Enters interface configuration mode and specify the Layer 3 interface to be associated with the VRF. The interface can be a routed port or SVI.
Step 2	**no switchport** **Example:** `Switch(config-if)# no switchport`	Removes the interface from configuration mode if it is a physical interface.

	Command or Action	**Purpose**
Step 3	**vrf forwarding** *vrf-name* **Example:** `Switch(config-if)# vrf forwarding vrf-name`	Associates the VRF with the Layer 3 interface.
Step 4	**ipv6 enable** **Example:** `Switch(config-if)# ipv6 enable`	Enable IPv6 on the interface.
Step 5	**ipv6 address** *ip-address subnet-mask* **Example:** `Switch(config-if)# ipv6 address ip-address subnet-mask`	Enters the IPv6 address for the interface.
Step 6	**show ipv6 vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] **Example:** `Switch# show ipv6 vrf [brief	detail	interfaces] [vrf-name]`	Verifies the configuration. Displays information about the configured VRFs.
Step 7	**copy running-config startup-config** **Example:** `Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Example

This example shows how to associate an interface to VRFs:

```
Switch(config-vrf)# interface ethernet0/1
Switch(config-if)# vrf forwarding red
Switch(config-if)# ipv6 enable
Switch(config-if)# ipv6 address 5000::72B/64
```

Populate VRF with Routes via Routing Protocols

Configuring VRF Static Routes

SUMMARY STEPS

1. **configure terminal**
2. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix*/*prefix-length* {*ipv6-address* \| **interface-type** *interface-number* [*ipv6-address*]}

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure terminal**	Enters global configuration mode.
	Example:	
	Switch# configure terminal	
Step 2	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix*/*prefix-length* {*ipv6-address* | **interface-type** *interface-number* [*ipv6-address*]}	To configure static routes specific to VRF.
	Example:	
	Switch(config)# ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address | interface-type interface-number [ipv6-address]}	

Example

```
Switch(config)# ipv6 route vrf v6a 7000::/64 TenGigabitEthernet32 4000::2
```

Configuring OSPFv3 Router Process

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3** *process-id*
3. **area** *area-ID* [**default-cot** \| **nssa** \| **stub**]
4. **router-id** *router-id*
5. **address-family ipv6 unicast vrf** *vrf-name*
6. **redistribute source-protocol** [*process-id*] **options**
7. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure terminal**	Enters global configuration mode.
	Example:	
	Switch# configure terminal	
Step 2	**router ospfv3** *process-id*	Enables OSPFv3 router configuration mode for the IPv6 address family.
	Example:	
	Switch(config)# router ospfv3 process-id	
Step 3	**area** *area-ID* [**default-cot** | **nssa** | **stub**]	Configures the OSPFv3 area.
	Example:	
	Switch(config-router)# area area-ID [default-cot | nssa | stub]	

	Command or Action	**Purpose**
Step 4	**router-id** *router-id* **Example:** `Switch(config-router)# router-id router-id`	Use a fixed router ID.

Step 5	**address-family ipv6 unicast vrf** *vrf-name*
	
	Example:
	
	`Switch(config-router)# address-family ipv6 unicast`
	`vrf vrf-name`
		Enters IPv6 address family configuration mode for OSPFv3 in VRF vrf-name

Step 6	**redistribute source-protocol** [*process-id*] **options**
	
	Example:
	
	`Switch(config-router)# redistribute source-protocol`
	`[process-id] options`
		Redistributes IPv6 routes from one routing domain into another routing domain.

Step 7	**end** **Example:** `Switch(config-router)# end`	Returns to privileged EXEC mode.

Example

This example shows how configure the OSPFv3 router process:

```
Switch(config-router)# router ospfv3 1
Switch(config-router)# router-id 1.1.1.1
Switch(config-router)# address-family ipv6 unicast
Switch(config-router-af)# exit-address-family
```

Enabling OSPFv3 on an Interface

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type-number*
3. **ospfv3** *process-id* **area** *area-ID* **ipv6** [**instance** *instance-id*]
4. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type-number* **Example:**	Specifies an interface type and number, and places the switch in interface configuration mode.

	Command or Action	Purpose
	`Switch(config-vrf)# interface type-number`	

Step 3	**ospfv3** *process-id* **area** *area-ID* **ipv6** [**instance** *instance-id*]
	
	Example:
	
	`Switch(config-if)# ospfv3 process-id area area-ID`
	`ipv6 [instance instance-id]`
		Enables OSPFv3 on an interface with IPv6 AF.

Step 4	**end** **Example:** `Switch(config-if)# end`	Returns to privileged EXEC mode.

Example

This example show how to enable OSPFv3 on an interface:

```
Switch(config)# interface GigabitEthernet2/1
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 4000::2/64
Switch(config-if)# ipv6 enable
Switch(config-if)# ipv6 ospf 1 area 0
Switch(config-if)# end
```

Configuring EIGRPv6 Routing Process

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp** *virtual-instance-name*
3. **address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number*
4. **topology** {**base** | **topology-name tid number**
5. **exit-aftopology**
6. **eigrp router-id** *ip-address*
7. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**router eigrp** *virtual-instance-name* **Example:** `Switch(config)# router eigrp virtual-instance-name`	Configures the EIGRP routing process and enters router configuration mode.

	Command or Action	Purpose

Step 3	**address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number*
	
	Example:
	`Switch(config-router)# address-family ipv6 vrf`
	`vrf-name autonomous-system autonomous-system-number`
		Enables EIGRP IPv6 VRF-Lite and enters address family configuration mode.

Step 4	**topology** {**base** | **topology-name tid number**
	
	Example:
	`Switch(config-router-af)# topology {base |`
	`topology-name tid number`
		Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.

Step 5	**exit-aftopology** **Example:** `Switch(config-router-af-topology)# exit-aftopology`	Exits address family topology configuration mode.
Step 6	**eigrp router-id** *ip-address* **Example:** `Switch(config-router)# eigrp router-id ip-address`	Enables the use of a fixed router-id.
Step 7	end **Example:** `Switch(config-router)# end`	Exits router configuration mode.

Example

This example shows how to configure an EIGRP routing process:

```
Switch(config)# router eigrp test
Switch(config-router)# address-family ipv6 unicast vrf b1 autonomous-system 10
Switch(config-router-af)# topology base
Switch(config-router-af-topology)# exit-af-topology
Switch(config-router)# eigrp router-id 2.3.4.5
Switch(config-router)# exit-address-family
```

Configuring EBGPv6 Routing Process

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *peer-group-name* **peer-group**
4. **neighbor** {*ip-address* \| *ipv6-address*[%] \| *peer-group-name*}**remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number ...*]
5. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** \| **multicast** \| **vpnv6**]
6. **neighbor** *ipv6-address* **peer-group** *peer-group-name*
7. **neighbor** {*ip-address* \| *peer-group-name* \| *ipv6-address*[%]}**route-map** *map-name* {**in** \| **out**}

8. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Switch# configure terminal	Enters global configuration mode.
Step 2	**router bgp** *as-number* **Example:** Switch(config)# router bgp as-number	Enters router configuration mode for the specified routing process.
Step 3	**neighbor** *peer-group-name peer-group* **Example:** Switch(config-router)# neighbor peer-group-name peer-group	Creates a multiprotocol BGP peer group.
Step 4	**neighbor** {*ip-address* | *ipv6-address*[%] | *peer-group-name*}**remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...] **Example:** Switch(config-router)# neighbor {ip-address | ipv6-address[%] | peer-group-name}remote-as autonomous-system-number [alternate-as autonomous-system-number ...]	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router.
Step 5	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**] **Example:** Switch(config-router)# address-family ipv6 [vrf vrf-name] [unicast | multicast | vpnv6]	Specifies the IPv6 address family, and enters address family configuration mode. • The unicast keyword specifies the IPv6 unicast address family. By default, the switch is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6command. • The multicast keyword specifies IPv6 multicast address prefixes.
Step 6	**neighbor** *ipv6-address* **peer-group** *peer-group-name* **Example:** Switch(config-router-af)# neighbor ipv6-address peer-group peer-group-name	Assigns the IPv6 address of a BGP neighbor to a peer group.
Step 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*[%]}**route-map** *map-name* {**in** | **out**} **Example:** Switch(config-router-af)# neighbor {ip-address | peer-group-name | ipv6-address[%]}route-map map-name {in | out}	Applies a route map to incoming or outgoing routes. Changes to the route map will not take effect for existing peers until the peering is reset or a soft reset is performed. Using the clear bgp ipv6 command with the soft and in keywords will perform a soft reset.

	Command or Action	**Purpose**
Step 8	exit **Example:** `Switch(config-router-af)# exit`	Exits address family configuration mode, and returns the router to router configuration mode.

Example

This example shows how to configure EBRPv6:

```
Switch(config)# router bgp 2
Switch(config-router)# bgp router-id 2.2.2.2
Switch(config-router)# bgp log-neighbor-changes
Switch(config-router)# no bgp default ipv4-unicast
Switch(config-router)# neighbor 2500::1 remote-as 1
Switch(config-router)# neighbor 4000::2 remote-as 3
Switch(config-router)# address-family ipv6 vrf b1
Switch(config-router-af)# network 2500::/64
Switch(config-router-af)# network 4000::/64
Switch(config-router-af)# neighbor 2500::1 remote-as 1
Switch(config-router-af)# neighbor 2500::1 activate
Switch(config-router-af)# neighbor 4000::2 remote-as 3
Switch(config-router-af)# neighbor 4000::2 activate
Switch(config-router-af)# exit-address-family
```

Additional Information for VRF-lite

VPN Co-existence Between IPv4 and IPv6

Backward compatibility between the "older" CLI for configuring IPv4 and the "new" CLI for IPv6 exists. This means that a configuration might contain both CLI. The IPv4 CLI retains the ability to have on the same interface, an IP address defined within a VRF as well as an IPv6 address defined in the global routing table.

For example:

```
vrf definition red
 rd 100:1
 address family ipv6
 route-target both 200:1
 exit-address-family
!
ip vrf blue
 rd 200:1
 route-target both 200:1
!
interface Ethernet0/0
 vrf forwarding red
 ip address 50.1.1.2 255.255.255.0
 ipv6 address 4000::72B/64
!
interface Ethernet0/1
 vrf forwarding blue
 ip address 60.1.1.2 255.255.255.0
 ipv6 address 5000::72B/64
```

In this example, all addresses (v4 and v6) defined for Ethernet0/0 refer to VRF red whereas for Ethernet0/1, the IP address refers to VRF blue but the ipv6 address refers to the global IPv6 routing table.

Verifying VRF-lite Configuration

Displaying IPv4 VRF-lite Status

To display information about VRF-lite configuration and status, perform one of the following tasks:

Command	Purpose
Switch# **show ip protocols vrf** *vrf-name*	Displays routing protocol information associated with a VRF.
Switch# **show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*]] [**list**] [**mobile**] [**odr**] [**profile**] [**static**] [**summary**][**supernets-only**]	Displays IP routing table information associated with a VRF.
Switch# **show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*]	Displays information about the defined VRF instances.
Switch# **bidir vrf** *instance-name a.b.c.d* | **active** | **bidriectional**| **count** | **interface** | **proxy** | **pruned** | **sparse** | **ssm** | **static** | **summary**	Displays information about the defined VRF instances.

This example shows how to display multicast route table information within a VRF instance:

```
Switch# show ip mroute 226.0.0.2
IP Multicast Routing Table
Flags: 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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 226.0.0.2), 00:01:17/stopped, RP 1.11.1.1, flags: SJCF
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan100, Forward/Sparse, 00:01:17/00:02:36

(5.0.0.11, 226.0.0.2), 00:01:17/00:01:42, flags: FT
  Incoming interface: Vlan5, RPF nbr 0.0.0.0
```

```
        Outgoing interface list:
          Vlan100, Forward/Sparse, 00:01:17/00:02:36
```

Displaying IPv6 VRF-lite Status

To display information about VRF-lite configuration and status, perform one of the following tasks:

Command	Purpose
Switch# **show ipv6 mroute vrf** *instance-name* [*X:X:X:X::X/<0-128>*] [**bgp**] [**connected**] [**eigrp**] [**interface**] [**isis**] [**local**] [**nd**] [**nsf**] [**ospf**] [**repair**] [**rip**] [**shortcut**] [**static**] [**summary**] [**tag**] [**updated**] [**watch**]	Displays routing protocol information associated with a VRF.
Switch# **show ipv6 mfib vrf** *instance-name* *a.b.c.d* | **active** | **all** | **count** | **linkscope**| **route** | **summary** | **update-sets** | **verbose**	Displays information about the defined VRF instances.

This example shows how to display multicast route table information within a VRF instance:

```
show ipv6 mroute vrf vrf1 FF05:ABCD:0:0:0:0:0:1
Multicast Routing Table
Flags: 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, Y - Joined MDT-data group,
y - Sending to MDT-data group

g - BGP signal originated, G - BGP Signal received,
N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
q - BGP Src-Active originated, Q - BGP Src-Active received
E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF05:ABCD::1), 00:06:22/never, RP 1010:ABCD::10, flags: SCJ
Incoming interface: Port-channel33
RPF nbr: FE80::2E31:24FF:FE06:134A
Immediate Outgoing interface list:
TenGigabitEthernet4/0/18, Forward, 00:06:22/never

(3232:ABCD::2, FF05:ABCD::1), 00:04:54/00:02:16, flags: SJT
Incoming interface: Port-channel33
RPF nbr: FE80::2E31:24FF:FE06:134A
Inherited Outgoing interface list:
TenGigabitEthernet4/0/18, Forward, 00:06:22/never
```

The following example displays the output of **show ipv6 mfib** command.

```
Switch# show ipv6 mfib vrf vrf1 FF05:ABCD:0:0:0:0:0:1
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.
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:   FS Pkt Count/PS Pkt Count
VRF testvrf1
 (*,FF05:ABCD::1) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   295/0/512/0, Other: 0/0/0
   Port-channel33 Flags: A NS
   TenGigabitEthernet4/0/18 Flags: F NS
     Pkts: 0/0
 (3232:ABCD::2,FF05:ABCD::1) Flags: HW
   SW Forwarding: 50/0/512/0, Other: 111/0/111
   HW Forwarding:   4387686/14849/512/59398, Other: 0/0/0
   Port-channel33 Flags: A
   TenGigabitEthernet4/0/18 Flags: F NS
     Pkts: 0/50

Switch#
```

Configuration Examples for VRF-lite

Configuration Example for IPv4 VRF-lite

OSPF is the protocol used in VPN1, VPN2, and the global network. BGP is used in the CE to PE connections. The example commands show how to configure the CE switch S8 and include the VRF configuration for switches S20 and S11 and the PE router commands related to traffic with switch S8. Commands for configuring the other switches are not included but would be similar.

Figure 5: VRF-lite Configuration Example
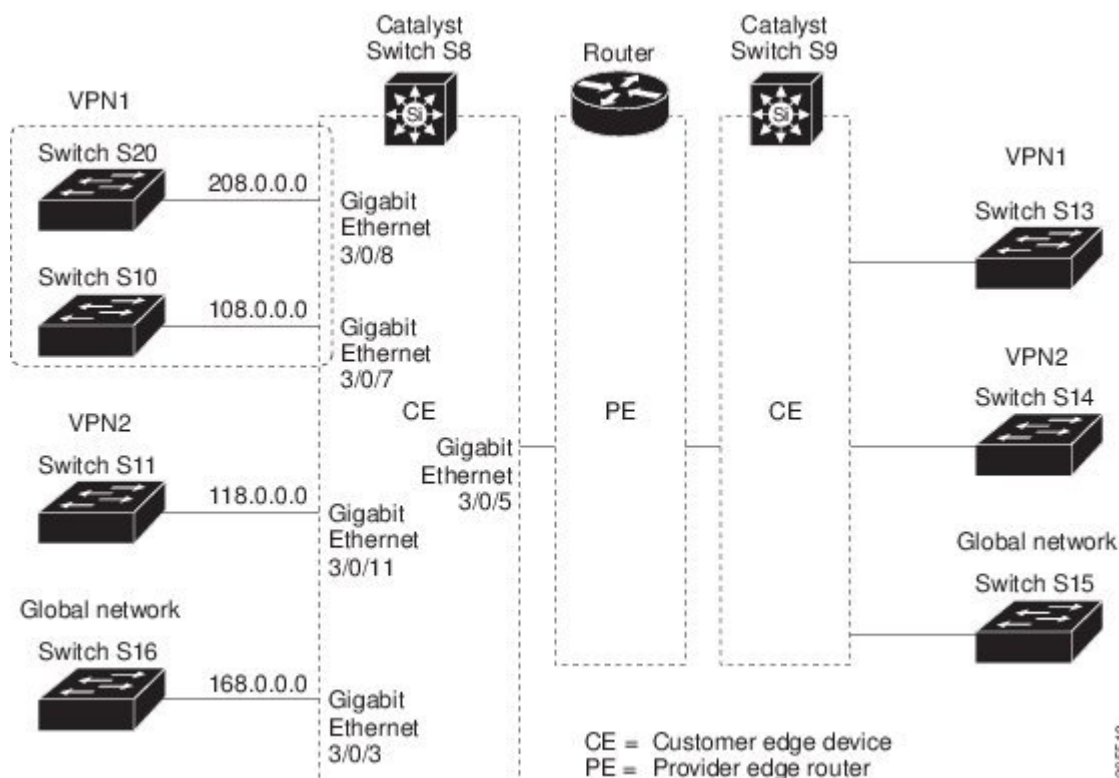
Configuring Switch S8

On switch S8, enable routing and configure VRF.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# ip vrf v11
Switch(config-vrf)# rd 800:1
Switch(config-vrf)# route-target export 800:1
Switch(config-vrf)# route-target import 800:1
Switch(config-vrf)# exit
Switch(config)# ip vrf v12
Switch(config-vrf)# rd 800:2
Switch(config-vrf)# route-target export 800:2
Switch(config-vrf)# route-target import 800:2
Switch(config-vrf)# exit
```

Configure the loopback and physical interfaces on switch S8. Fast Ethernet interface 3/5 is a trunk connection to the PE. Interfaces 3/7 and 3/11 connect to VPNs:

```
Switch(config)# interface loopback1
Switch(config-if)# vrf forwarding v11
Switch(config-if)# ip address 8.8.1.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface loopback2
Switch(config-if)# vrf forwarding v12
Switch(config-if)# ip address 8.8.2.8 255.255.255.0
Switch(config-if)# exit
```

```
Switch(config)# interface FastEthernet3/5
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface FastEthernet3/8
Switch(config-if)# switchport access vlan 208
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface FastEthernet3/11
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit
```

Configure the VLANs used on switch S8. VLAN 10 is used by VRF 11 between the CE and the PE. VLAN 20 is used by VRF 12 between the CE and the PE. VLANs 118 and 208 are used for VRF for the VPNs that include switch S11 and switch S20, respectively:

```
Switch(config)# interface Vlan10
Switch(config-if)# vrf forwarding v11
Switch(config-if)# ip address 38.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan20
Switch(config-if)# vrf forwarding v12
Switch(config-if)# ip address 83.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan118
Switch(config-if)# vrf forwarding v12
Switch(config-if)# ip address 118.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan208
Switch(config-if)# vrf forwarding v11
Switch(config-if)# ip address 208.0.0.8 255.255.255.0
Switch(config-if)# exit
```

Configure OSPF routing in VPN1 and VPN2:

```
Switch(config)# router ospf 1 vrf vl1
Switch(config-router)# redistribute bgp 800 subnets
Switch(config-router)# network 208.0.0.0 0.0.0.255 area 0
Switch(config-router)# exit
Switch(config)# router ospf 2 vrf vl2
Switch(config-router)# redistribute bgp 800 subnets
Switch(config-router)# network 118.0.0.0 0.0.0.255 area 0
Switch(config-router)# exit
```

Configure BGP for CE to PE routing:

```
Switch(config)# router bgp 800
Switch(config-router)# address-family ipv4 vrf vl2
Switch(config-router-af)# redistribute ospf 2 match internal
Switch(config-router-af)# neighbor 83.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 83.0.0.3 activate
Switch(config-router-af)# network 8.8.2.0 mask 255.255.255.0
Switch(config-router-af)# exit

Switch(config-router)# address-family ipv4 vrf vl1
Switch(config-router-af)# redistribute ospf 1 match internal
Switch(config-router-af)# neighbor 38.0.0.3 remote-as 100
```

```
Switch(config-router-af)# neighbor 38.0.0.3 activate
Switch(config-router-af)# network 8.8.1.0 mask 255.255.255.0
Switch(config-router-af)# end
```

Configuring Switch S20

Configure S20 to connect to CE:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface Fast Ethernet 0/7
Switch(config-if)# no switchport
Switch(config-if)# ip address 208.0.0.20 255.255.255.0
Switch(config-if)# exit

Switch(config)# router ospf 101
Switch(config-router)# network 208.0.0.0 0.0.0.255 area 0
Switch(config-router)# end
```

Configuring Switch S11

Configure S11 to connect to CE:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface Gigabit Ethernet 0/3
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface Vlan118
Switch(config-if)# ip address 118.0.0.11 255.255.255.0
Switch(config-if)# exit

Switch(config)# router ospf 101
Switch(config-router)# network 118.0.0.0 0.0.0.255 area 0
Switch(config-router)# end
```

Configuring the PE Switch S3

On switch S3 (the router), these commands configure only the connections to switch S8:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip vrf v1
Router(config-vrf)# rd 100:1
Router(config-vrf)# route-target export 100:1
Router(config-vrf)# route-target import 100:1
Router(config-vrf)# exit

Router(config)# ip vrf v2
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target export 100:2
Router(config-vrf)# route-target import 100:2
Router(config-vrf)# exit

Router(config)# ip cef
Router(config)# interface Loopback1
Router(config-if)# vrf forwarding v1
Router(config-if)# ip address 3.3.1.3 255.255.255.0
```

```
Router(config-if)# exit

Router(config)# interface Loopback2
Router(config-if)# vrf forwarding v2
Router(config-if)# ip address 3.3.2.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Fast Ethernet3/0.10
Router(config-if)# encapsulation dot1q 10
Router(config-if)# vrf forwarding v1
Router(config-if)# ip address 38.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Fast Ethernet3/0.20
Router(config-if)# encapsulation dot1q 20
Router(config-if)# vrf forwarding v2
Router(config-if)# ip address 83.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf v2
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.2.0 mask 255.255.255.0
Router(config-router-af)# exit
Router(config-router)# address-family ipv4 vrf vl
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.1.0 mask 255.255.255.0
Router(config-router-af)# end
```

Configuration Example for IPv6 VRF-lite

The following topology illustrates how to use OSPFv3 for CE-PE routing.

Figure 6: VRF-lite Configuration Example
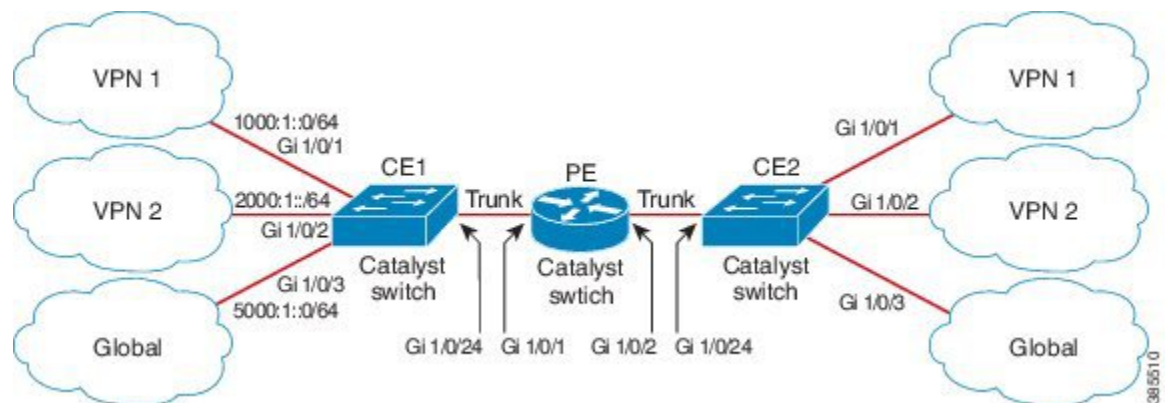
Configuring CE1 Switch

```
ipv6 unicast-routing
vrf definition v1
 rd 100:1
 !
address-family ipv6
 exit-address-family
```

```
!

vrf definition v2
 rd 200:1
 !
address-family ipv6
 exit-address-family
!

interface Vlan100
 vrf forwarding v1
 ipv6 address 1000:1::1/64
 ospfv3 100 ipv6 area 0
!

interface Vlan200
 vrf forwarding v2
 ipv6 address 2000:1::1/64
 ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
switchport access vlan 100
end

interface GigabitEthernet 1/0/2
switchport access vlan 200
end

interface GigabitEthernet 1/0/24
 switchport trunk encapsulation dot1q

switchport mode trunk
end

router ospfv3 100
 router-id 10.10.10.10
 !
 address-family ipv6 unicast vrf v1
  redistribute connected
  area 0 normal
 exit-address-family
!

router ospfv3 200
 router-id 20.20.20.20
 !
 address-family ipv6 unicast vrf v2
  redistribute connected
  area 0 normal
 exit-address-family
!
```

Configuring PE Switch

```
ipv6 unicast-routing

vrf definition v1
 rd 100:1
 !
address-family ipv6
 exit-address-family
!
```

```
vrf definition v2
 rd 200:1
 !
address-family ipv6
 exit-address-family
!
interface Vlan600
 vrf forwarding v1
 no ipv6 address
 ipv6 address 1000:1::2/64
 ospfv3 100 ipv6 area 0
!

interface Vlan700
 vrf forwarding v2
 no ipv6 address
 ipv6 address 2000:1::2/64
 ospfv3 200 ipv6 area 0
!

interface Vlan800
 vrf forwarding v1
 ipv6 address 3000:1::7/64
 ospfv3 100 ipv6 area 0
!
interface Vlan900
 vrf forwarding v2
 ipv6 address 4000:1::7/64
 ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
 switchport trunk encapsulation dot1q
 switchport mode trunk
 exit

interface GigabitEthernet 1/0/2
 switchport trunk encapsulation dot1q

switchport mode trunk
 exit

router ospfv3 100
 router-id 30.30.30.30
 !
 address-family ipv6 unicast vrf v1
  redistribute connected
  area 0 normal
 exit-address-family
 !
address-family ipv6 unicast vrf v2
  redistribute connected
  area 0 normal
 exit-address-family
 !
```

Configuring CE2 Switch

```
ipv6 unicast-routing

vrf definition v1
 rd 100:1
 !
address-family ipv6
```

```
 exit-address-family
!

vrf definition v2
 rd 200:1
 !
address-family ipv6
 exit-address-family
!

interface Vlan100
 vrf forwarding v1

ipv6 address 1000:1::3/64
 ospfv3 100 ipv6 area 0
!

interface Vlan200
 vrf forwarding v2
 ipv6 address 2000:1::3/64
 ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
switchport access vlan 100
end

interface GigabitEthernet 1/0/2
switchport access vlan 200
end

interface GigabitEthernet 1/0/24
 switchport trunk encapsulation dot1q
 switchport mode trunk
 end

router ospfv3 100
 router-id 40.40.40.40
 !
 address-family ipv6 unicast vrf v1
  redistribute connected
  area 0 normal
 exit-address-family
!

router ospfv3 200
 router-id 50.50.50.50
 !
 address-family ipv6 unicast vrf v2
  redistribute connected

area 0 normal
 exit-address-family
!
```

Feature History and Information for Multicast VRF-lite

Feature Name	Release	Feature Information
IPv6 Multicast support with VRF-Lite	Cisco IOS XE Everest 16.6.1	IPv6 VRF-Lite allows a service provider to support two or more VPNs with overlapping IP addresses using one interface. This feature was introduced.

Configuring IGMP Snooping

Prerequisites for IGMP and IGMP Snooping

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 Basic IP Multicast Routing" module.

Prerequisites for IGMP Snooping

Observe these guidelines when configuring the IGMP snooping querier:

- Configure the VLAN in global configuration mode.

- Configure an IP address on the VLAN interface. When enabled, the IGMP snooping querier uses the IP address as the query source address.

- If there is no IP address configured on the VLAN interface, the IGMP snooping querier tries to use the configured global IP address for the IGMP querier. If there is no global IP address specified, the IGMP querier tries to use the VLAN device virtual interface (SVI) IP address (if one exists). If there is no SVI IP address, the device uses the first available IP address configured on the device. The first IP address available appears in the output of the **show ip interface** privileged EXEC command. The IGMP snooping querier does not generate an IGMP general query if it cannot find an available IP address on the device.

• The IGMP snooping querier supports IGMP Versions 1 and 2.

• When administratively enabled, the IGMP snooping querier moves to the nonquerier state if it detects the presence of a multicast router in the network.

• When it is administratively enabled, the IGMP snooping querier moves to the operationally disabled state under these conditions:

  • IGMP snooping is disabled in the VLAN.

  • PIM is enabled on the SVI of the corresponding VLAN.

•

•

Restrictions for IGMP and IGMP Snooping

Restrictions for Configuring IGMP

The following are the restrictions for configuring IGMP:

• The device supports IGMP Versions 1, 2 , and 3.

> **Note**   For IGMP Version 3, only IGMP Version 3 BISS (Basic IGMPv3 Snooping Support) is supported.

• IGMP Version 3 uses new membership report messages that might not be correctly recognized by older IGMP snooping devices.

• IGMPv3 can operate with both ISM and SSM. In ISM, both exclude and include mode reports are applicable. In SSM, only include mode reports are accepted by the last-hop router. Exclude mode reports are ignored.

• IGMP filtering and throttling is not supported under the WLAN.

• You cannot have a device stack containing a mix of Catalyst 3850 and Catalyst 3650 devices.

Restrictions for IGMP Snooping

The following are the restrictions for IGMP snooping:

• The device supports IGMPv3 snooping based only on the destination multicast IP address. It does not support snooping based on a source IP address or proxy report.

• IGMPv3 join and leave messages are not supported on devices running IGMP filtering or Multicast VLAN registration (MVR).

• IGMP report suppression is supported only when the multicast query has IGMPv1 and IGMPv2 reports. This feature is not supported when the query includes IGMPv3 reports.

- The IGMP configurable leave time is only supported on hosts running IGMP Version 2. IGMP version 2 is the default version for the device.

  The actual leave latency in the network is usually the configured leave time. However, the leave time might vary around the configured time, depending on real-time CPU load conditions, network delays and the amount of traffic sent through the interface.

- The IGMP throttling action restriction can be applied only to Layer 2 ports. You can use **ip igmp max-groups action replace** interface configuration command on a logical EtherChannel interface but cannot use it on ports that belong to an EtherChannel port group.

  When the maximum group limitation is set to the default (no maximum), entering the **ip igmp max-groups action** {**deny** | **replace**} command has no effect.

  If you configure the throttling action and set the maximum group limitation after an interface has added multicast entries to the forwarding table, the forwarding-table entries are either aged out or removed, depending on the throttling action.

Information About 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 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.

IGMP Versions

The device supports IGMP version 1, IGMP version 2, and IGMP version 3. These versions are interoperable on the device. For example, if IGMP snooping is enabled and the querier's version is IGMPv2, and the device receives an IGMPv3 report from a host, then the device can forward the IGMPv3 report to the multicast router.

An IGMPv3 device can receive messages from and forward messages to a device running the Source Specific Multicast (SSM) feature.

IGMP Version 1

IGMP version 1 (IGMPv1) primarily uses a query-response model that enables the multicast router and multilayer device to find which multicast groups are active (have one or more hosts interested in a multicast group) on the local subnet. IGMPv1 has other processes that enable a host to join and leave a multicast group. For more information, see RFC 1112.

IGMP Version 2

IGMPv2 extends IGMP functionality by providing such features as the IGMP leave process to reduce leave latency, group-specific queries, and an explicit maximum query response time. IGMPv2 also adds the capability for routers to elect the IGMP querier without depending on the multicast protocol to perform this task. For more information, see RFC 2236.

Note IGMP version 2 is the default version for the device.

IGMP Version 3

The device supports IGMP version 3.

An IGMPv3 device supports Basic IGMPv3 Snooping Support (BISS), which includes support for the snooping features on IGMPv1 and IGMPv2 switches and for IGMPv3 membership report messages. BISS constrains the flooding of multicast traffic when your network includes IGMPv3 hosts. It constrains traffic to approximately the same set of ports as the IGMP snooping feature on IGMPv2 or IGMPv1 hosts.

An IGMPv3 device can receive messages from and forward messages to a device running the Source Specific Multicast (SSM) feature.

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.

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 7: 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 and Leave Process

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 Snooping

Layer 2 devices can use IGMP snooping to constrain the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. As the name implies, IGMP snooping requires the LAN device to snoop on the IGMP transmissions between the host and the router and to keep track of multicast groups and member ports. When the device receives an IGMP report from a host for a particular multicast group, the device adds the host port number to the forwarding table entry; when it receives an IGMP Leave Group message from a host, it removes the host port from the table entry. It also periodically deletes entries if it does not receive IGMP membership reports from the multicast clients.

Note For more information on IP multicast and IGMP, see RFC 1112 and RFC 2236.

The multicast router (which could be a device with the IP services feature) set on the active device sends out periodic general queries to all VLANs. All hosts interested in this multicast traffic send join requests and are added to the forwarding table entry. The device creates one entry per VLAN in the IGMP snooping IP multicast forwarding table for each group from which it receives an IGMP join request.

The device supports IP multicast group-based bridging, instead of MAC-addressed based groups. With multicast MAC address-based groups, if an IP address being configured translates (aliases) to a previously configured MAC address or to any reserved multicast MAC addresses (in the range 224.0.0.xxx), the command fails. Because the device uses IP multicast groups, there are no address aliasing issues.

The IP multicast groups learned through IGMP snooping are dynamic. However, you can statically configure multicast groups by using the **ip igmp snooping vlan** *vlan-id* **static** *ip_address* **interface** *interface-id* global configuration command. If you specify group membership for a multicast group address statically, your setting supersedes any automatic manipulation by IGMP snooping. Multicast group membership lists can consist of both user-defined and IGMP snooping-learned settings.

You can configure an IGMP snooping querier to support IGMP snooping in subnets without multicast interfaces because the multicast traffic does not need to be routed.

If a port spanning-tree, a port group, or a VLAN ID change occurs, the IGMP snooping-learned multicast groups from this port on the VLAN are deleted.

These sections describe IGMP snooping characteristics:

Joining a Multicast Group

Figure 7: Initial IGMP Join Message

When a host connected to the device wants to join an IP multicast group and it is an IGMP version 2 client, it sends an unsolicited IGMP join message, specifying the IP multicast group to join. Alternatively, when the device receives a general query from the router, it forwards the query to all ports in the VLAN. IGMP version 1 or version 2 hosts wanting to join the multicast group respond by sending a join message to the device. The device CPU creates a multicast forwarding-table entry for the group if it is not already present. The CPU also adds the interface where the join message was received to the forwarding-table entry. The host associated with that interface receives multicast traffic for that multicast group.
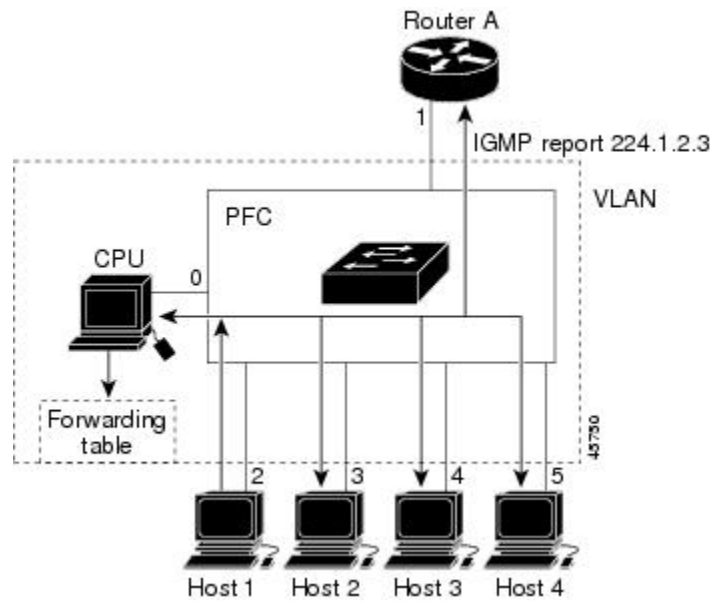
Router A sends a general query to the device, which forwards the query to ports 2 through 5, all of which are members of the same VLAN. Host 1 wants to join multicast group 224.1.2.3 and multicasts an IGMP membership report (IGMP join message) to the group. The device CPU uses the information in the IGMP report to set up a forwarding-table entry that includes the port numbers connected to Host 1 and to the router.

Table 8: IGMP Snooping Forwarding Table

Destination Address	Type of Packet	Ports
224.1.2.3	IGMP	1, 2

The device hardware can distinguish IGMP information packets from other packets for the multicast group. The information in the table tells the switching engine to send frames addressed to the 224.1.2.3 multicast IP address that are not IGMP packets to the router and to the host that has joined the group.

Figure 8: Second Host Joining a Multicast Group

If another host (for example, Host 4) sends an unsolicited IGMP join message for the same group, the CPU receives that message and adds the port number of Host 4 to the forwarding table. Because the forwarding table directs IGMP messages only to the CPU, the message is not flooded to other ports on the device. Any

known multicast traffic is forwarded to the group and not to the CPU.
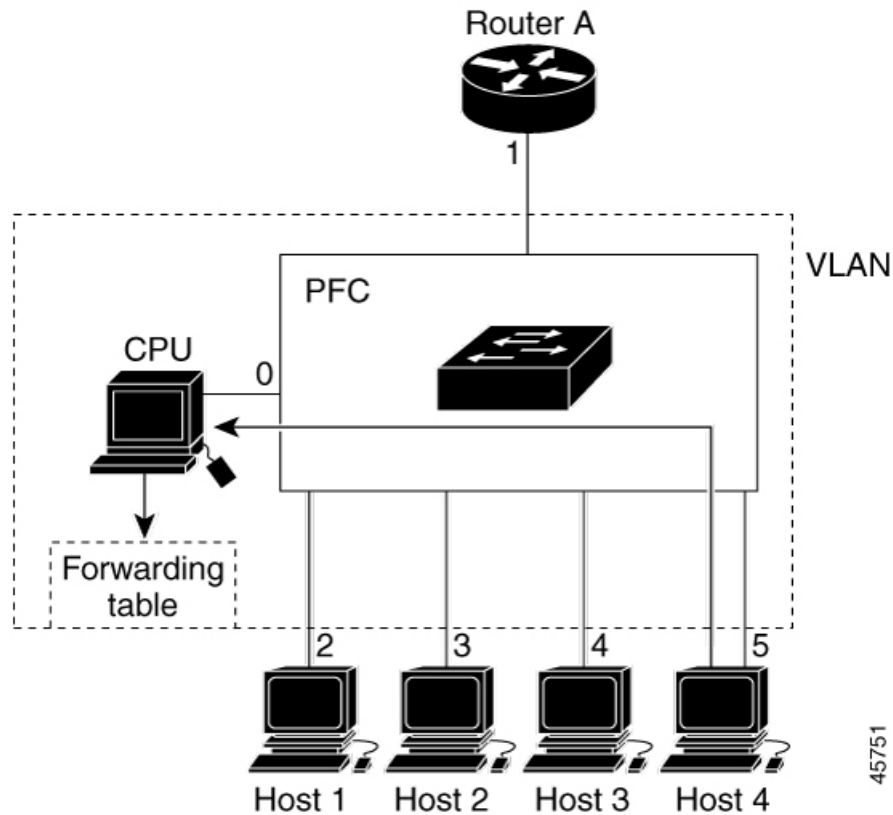
Table 9: Updated IGMP Snooping Forwarding Table

Destination Address	Type of Packet	Ports
224.1.2.3	IGMP	1, 2, 5

Leaving a Multicast Group

The router sends periodic multicast general queries, and the device forwards these queries through all ports in the VLAN. Interested hosts respond to the queries. If at least one host in the VLAN wants to receive multicast traffic, the router continues forwarding the multicast traffic to the VLAN. The device forwards multicast group traffic only to those hosts listed in the forwarding table for that IP multicast group maintained by IGMP snooping.

When hosts want to leave a multicast group, they can silently leave, or they can send a leave message. When the device receives a leave message from a host, it sends a group-specific query to learn if any other devices connected to that interface are interested in traffic for the specific multicast group. The device then updates the forwarding table for that MAC group so that only those hosts interested in receiving multicast traffic for the group are listed in the forwarding table. If the router receives no reports from a VLAN, it removes the group for the VLAN from its IGMP cache.

Immediate Leave

The device uses IGMP snooping Immediate Leave to remove from the forwarding table an interface that sends a leave message without the device sending group-specific queries to the interface. The VLAN interface is pruned from the multicast tree for the multicast group specified in the original leave message. Immediate Leave ensures optimal bandwidth management for all hosts on a switched network, even when multiple multicast groups are simultaneously in use.

Immediate Leave is only supported on IGMP version 2 hosts. IGMP version 2 is the default version for the device.

Note You should use the Immediate Leave feature only on VLANs where a single host is connected to each port. If Immediate Leave is enabled on VLANs where more than one host is connected to a port, some hosts may be dropped inadvertently.

IGMP Configurable-Leave Timer

You can configure the time that the device waits after sending a group-specific query to determine if hosts are still interested in a specific multicast group. The IGMP leave response time can be configured from 100 to 32767 milliseconds.

IGMP Report Suppression

Note IGMP report suppression is supported only when the multicast query has IGMPv1 and IGMPv2 reports. This feature is not supported when the query includes IGMPv3 reports.

The device uses IGMP report suppression to forward only one IGMP report per multicast router query to multicast devices. When IGMP report suppression is enabled (the default), the device sends the first IGMP report from all hosts for a group to all the multicast routers. The device does not send the remaining IGMP reports for the group to the multicast routers. This feature prevents duplicate reports from being sent to the multicast devices.

If the multicast router query includes requests only for IGMPv1 and IGMPv2 reports, the device forwards only the first IGMPv1 or IGMPv2 report from all hosts for a group to all the multicast routers.

If the multicast router query also includes requests for IGMPv3 reports, the device forwards all IGMPv1, IGMPv2, and IGMPv3 reports for a group to the multicast devices.

If you disable IGMP report suppression, all IGMP reports are forwarded to the multicast routers.

IGMP Filtering and Throttling

In some environments, for example, metropolitan or multiple-dwelling unit (MDU) installations, you might want to control the set of multicast groups to which a user on a device port can belong. You can control the distribution of multicast services, such as IP/TV, based on some type of subscription or service plan. You might also want to limit the number of multicast groups to which a user on a device port can belong.

With the IGMP filtering feature, you can filter multicast joins on a per-port basis by configuring IP multicast profiles and associating them with individual device ports. An IGMP profile can contain one or more multicast groups and specifies whether access to the group is permitted or denied. If an IGMP profile denying access

to a multicast group is applied to a device port, the IGMP join report requesting the stream of IP multicast traffic is dropped, and the port is not allowed to receive IP multicast traffic from that group. If the filtering action permits access to the multicast group, the IGMP report from the port is forwarded for normal processing. You can also set the maximum number of IGMP groups that a Layer 2 interface can join.

IGMP filtering controls only group-specific query and membership reports, including join and leave reports. It does not control general IGMP queries. IGMP filtering has no relationship with the function that directs the forwarding of IP multicast traffic. The filtering feature operates in the same manner whether CGMP or MVR is used to forward the multicast traffic.

IGMP filtering applies only to the dynamic learning of IP multicast group addresses, not static configuration.

With the IGMP throttling feature, you can set the maximum number of IGMP groups that a Layer 2 interface can join. If the maximum number of IGMP groups is set, the IGMP snooping forwarding table contains the maximum number of entries, and the interface receives an IGMP join report, you can configure an interface to drop the IGMP report or to replace the randomly selected multicast entry with the received IGMP report.

Note IGMPv3 join and leave messages are not supported on devices running IGMP filtering.

Default IGMP Configuration

This table displays the default IGMP configuration for the device.

Table 10: Default IGMP Configuration

Feature	Default Setting
Multilayer device as a member of a multicast group	No group memberships are defined.
Access to multicast groups	All groups are allowed on an interface.
IGMP version	Version 2 on all interfaces.
IGMP host-query message interval	60 seconds on all interfaces.
IGMP query timeout	60 seconds on all interfaces.
IGMP maximum query response time	10 seconds on all interfaces.
Multilayer device as a statically connected member	Disabled.

Default IGMP Snooping Configuration

This table displays the default IGMP snooping configuration for the device.

Table 11: Default IGMP Snooping Configuration

Feature	Default Setting
IGMP snooping	Enabled globally and per VLAN
Multicast routers	None configured

Feature	Default Setting
IGMP snooping Immediate Leave	Disabled
Static groups	None configured
TCN[1] flood query count	2
TCN query solicitation	Disabled
IGMP snooping querier	Disabled
IGMP report suppression	Enabled

[1] (1) TCN = Topology Change Notification

Default IGMP Filtering and Throttling Configuration

This table displays the default IGMP filtering and throttling configuration for the device.

Table 12: Default IGMP Filtering Configuration

Feature	Default Setting
IGMP filters	None applied.
IGMP maximum number of IGMP groups	No maximum set.
	Note When the maximum number of groups is in the forwarding table, the default IGMP throttling action is to deny the IGMP report.
IGMP profiles	None defined.
IGMP profile action	Deny the range addresses.

How to Configure IGMP

Configuring the Device as a Member of a Group

You can configure the device as a member of a multicast group and discover multicast reachability in a network. If all the multicast-capable routers and multilayer devices that you administer are members of a multicast group, pinging that group causes all of these devices to respond. The devices respond to ICMP echo-request packets addressed to a group of which they are members. Another example is the multicast trace-route tools provided in the software.

⚠️

Caution   Performing this procedure might impact the CPU performance because the CPU will receive all data traffic for the group address.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp join-group** *group-address*
5. **end**
6. **show ip igmp interface** [*interface-id*]
7. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specifies the Layer 3 interface on which you want to enable multicast routing, and enters interface configuration mode. The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. These interfaces must have IP addresses assigned to them.
Step 4	**ip igmp join-group** *group-address* **Example:** Device(config-if)# **ip igmp join-group 225.2.2.2**	Configures the device to join a multicast group. By default, no group memberships are defined. For *group-address*, specify the multicast IP address in dotted decimal notation.
Step 5	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config)# **end**	
Step 6	**show ip igmp interface** [*interface-id*] **Example:** Device# **show ip igmp interface**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Controlling Access to IP Multicast Group

The device sends IGMP host-query messages to find which multicast groups have members on attached local networks. The device then forwards to these group members all packets addressed to the multicast group. You can place a filter on each interface to restrict the multicast groups that hosts on the subnet serviced by the interface can join.

To limit the number of joins on the interface, configure the port for the filter which associates with the IGMP profile.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp profile**
4. **permit**
5. **exit**
6. **interface** *interface-id*
7. **ip igmp filter** *filter_number*
8. **end**
9. **show ip igmp interface** [*interface-id*]
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 profile** **Example:** Device(config)# **ip igmp profile 10** Device(config-igmp-profile)# **?**	Enters an IGMP filter profile number from 1 to 4294967295. For additional information about configuring IGMP filter profiles, see Configuring IGMP Profiles , on page 99.

Step 4	**permit**
	
	Example:
	
	Device(config-igmp-profile)#
	permit 229.9.9.0
		Enters an IGMP profile configuration action. The following IGMP profile configuration actions are supported:
		
		   • **deny**—Matching IP addresses are denied.
		
		   • **exit**—Exits from the IGMP profile configuration mode.
		
		   • **no**—Negates a command or set its defaults.
		
		   • **permit**—Matching addresses are permitted.
		
		   • **range**—Adds a range to the set.

Step 5	**exit** **Example:** Device(config-igmp-profile)# **exit**	Returns to global configuration mode.
Step 6	**interface** *interface-id* **Example:** Device(config)# **interface** **gigabitethernet 1/0/1**	Specifies the interface to be configured, and enters interface configuration mode.
Step 7	**ip igmp filter** *filter_number* **Example:** Device(config-if)# **ip igmp filter 10**	Specifies the IGMP filter profile number. For additional information about applying IGMP filter profiles, see Applying IGMP Profiles , on page 101.
Step 8	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config)# **end**	
Step 9	**show ip igmp interface** [*interface-id*]	Verifies your entries.
	Example:	
	Device# **show ip igmp interface**	
Step 10	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.
	Example:	
	Device# **copy running-config startup-config**	

Changing the IGMP Version

By default, the switch uses IGMP Version 2, which provides features such as the IGMP query timeout and the maximum query response time.

All systems on the subnet must support the same version. The switch does not automatically detect Version 1 systems and switch to Version 1. You can mix Version 1 and Version 2 hosts on the subnet because Version 2 routers or switches always work correctly with IGMPv1 hosts.

Configure the switch for Version 1 if your hosts do not support Version 2.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp version** {**1** | **2** | **3** }
5. **end**
6. **show ip igmp interface** [*interface-id*]
7. **copy running-config startup-config**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** Device# `configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** Device(config)# `interface gigabitethernet 1/0/1`	Specifies the interface to be configured, and enters the interface configuration mode.
Step 4	**ip igmp version** {**1** | **2** | **3** } **Example:** Device(config-if)# `ip igmp version 2`	Specifies the IGMP version that the switch uses. **Note** If you change to Version 1, you cannot configure the **ip igmp query-interval** or the **ip igmp query-max-response-time** interface configuration commands. To return to the default setting, use the **no ip igmp version** interface configuration command.
Step 5	**end** **Example:** Device(config)# `end`	Returns to privileged EXEC mode.
Step 6	**show ip igmp interface** [*interface-id*] **Example:** Device# `show ip igmp interface`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** Device# `copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Modifying the IGMP Host-Query Message Interval

The device periodically sends IGMP host-query messages to discover which multicast groups are present on attached networks. These messages are sent to the all-hosts multicast group (224.0.0.1) with a time-to-live (TTL) of 1. The device sends host-query messages to refresh its knowledge of memberships present on the network. If, after some number of queries, the software discovers that no local hosts are members of a multicast group, the software stops forwarding multicast packets to the local network from remote origins for that group and sends a prune message upstream toward the source.

The device elects a PIM designated router (DR) for the LAN (subnet). The designated router is responsible for sending IGMP host-query messages to all hosts on the LAN. In sparse mode, the designated router also sends PIM register and PIM join messages toward the RP router. With IGMPv2, the DR is the router or multilayer device with the highest IP address. With IGMPv1, the DR is elected according to the multicast routing protocol that runs on the LAN.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp query-interval** *seconds*
5. **end**
6. **show ip igmp interface** [*interface-id*]
7. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specifies the Layer 3 interface on which you want to enable multicast routing, and enters interface configuration mode. The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. These interfaces must have IP addresses assigned to them.
Step 4	**ip igmp query-interval** *seconds* **Example:**	Configures the frequency at which the designated router sends IGMP host-query messages.

	Command or Action	**Purpose**
	Device(config-if)# **ip igmp query-interval 75**	By default, the designated router sends IGMP host-query messages every 60 seconds to keep the IGMP overhead very low on hosts and networks. The range is 1 to 65535.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show ip igmp interface** [*interface-id*] **Example:** Device# **show ip igmp interface**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Changing the IGMP Query Timeout for IGMPv2

If you are using IGMPv2, you can specify the period of time before the device takes over as the querier for the interface. By default, the device waits twice the query interval period controlled by the **ip igmp query-interval** interface configuration command. After that time, if the device has received no queries, it becomes the querier.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp querier-timeout** *seconds*
5. **end**
6. **show ip igmp interface** [*interface-id*]
7. **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**	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	**interface** *interface-id*	Specifies the Layer 3 interface on which you want to enable multicast routing, and enters interface configuration mode.
	Example:	The specified interface must be one of the following:
	Device(config)# **interface gigabitethernet 1/0/1**	• A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command.
		• An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command.
		These interfaces must have IP addresses assigned to them.
Step 4	**ip igmp querier-timeout** *seconds*	Specifies the IGMP query timeout.
	Example:	The default is 60 seconds (twice the query interval). The range is 60 to 300.
	Device(config-if)# **ip igmp querier-timeout 120**	
Step 5	**end**	Returns to privileged EXEC mode.
	Example:	
	Device(config)# **end**	
Step 6	**show ip igmp interface** [*interface-id*]	Verifies your entries.
	Example:	
	Device# **show ip igmp interface**	
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.
	Example:	
	Device# **copy running-config startup-config**	

Changing the Maximum Query Response Time for IGMPv2

If you are using IGMPv2, you can change the maximum query response time advertised in IGMP queries. The maximum query response time enables the device to quickly detect that there are no more directly connected group members on a LAN. Decreasing the value enables the device to prune groups faster.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp query-max-response-time** *seconds*
5. **end**
6. **show ip igmp interface** [*interface-id*]
7. **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** Example: Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* Example: Device(config)# **interface gigabitethernet 1/0/1**	Specifies the Layer 3 interface on which you want to enable multicast routing, and enters interface configuration mode. The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. These interfaces must have IP addresses assigned to them.
Step 4	**ip igmp query-max-response-time** *seconds* Example: Device(config-if)# **ip igmp**	Changes the maximum query response time advertised in IGMP queries. The default is 10 seconds. The range is 1 to 25.

	Command or Action	Purpose
	`query-max-response-time 15`	
Step 5	**end** Example: Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show ip igmp interface** [*interface-id*] Example: Device# **show ip igmp interface**	Verifies your entries.
Step 7	**copy running-config startup-config** Example: Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring the Device as a Statically Connected Member

At various times, either there is not a group member on a network segment or a host that cannot report its group membership by using IGMP. However, you may want multicast traffic to be sent to that network segment. The following commands are used to pull multicast traffic down to a network segment:

- **ip igmp join-group**—The device accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the device from fast switching.

- **ip igmp static-group**—The device does not accept the packets itself, but only forwards them. This method enables 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.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp static-group** *group-address*
5. **end**
6. **show ip igmp interface** [*interface-id*]
7. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specifies the Layer 3 interface on which you want to enable multicast routing, and enters interface configuration mode. The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. These interfaces must have IP addresses assigned to them.
Step 4	**ip igmp static-group** *group-address* **Example:** Device(config-if)# **ip igmp static-group 239.100.100.101**	Configures the device as a statically connected member of a group. By default, this feature is disabled.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show ip igmp interface** [*interface-id*] **Example:** Device# **show ip igmp interface gigabitethernet 1/0/1**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

Command or Action	Purpose
Device# **copy running-config startup-config**	

Configuring IGMP Profiles

Follow these steps to create an IGMP profile:

This task is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp profile** *profile number*
4. **permit** | **deny**
5. **range** *ip multicast address*
6. **end**
7. **show ip igmp profile** *profile number*
8. **show running-config**
9. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip igmp profile** *profile number* **Example:** Device(config)# **ip igmp profile 3**	Assigns a number to the profile you are configuring, and enters IGMP profile configuration mode. The profile number range is 1 to 4294967295. When you are in IGMP profile configuration mode, you can create the profile by using these commands: • **deny**—Specifies that matching addresses are denied; this is the default. • **exit**—Exits from igmp-profile configuration mode. • **no**—Negates a command or returns to its defaults.

Command or Action	Purpose
	• **permit**—Specifies that matching addresses are permitted.
	• **range**—Specifies a range of IP addresses for the profile. You can enter a single IP address or a range with a start and an end address.
	The default is for the device to have no IGMP profiles configured.
	Note To delete a profile, use the **no ip igmp profile** *profile number* global configuration command.
Step 4 **permit** | **deny** **Example:** Device(config-igmp-profile)# **permit**	(Optional) Sets the action to permit or deny access to the IP multicast address. If no action is configured, the default for the profile is to deny access.
Step 5 **range** *ip multicast address* **Example:** Device(config-igmp-profile)# **range 229.9.9.0**	Enters the IP multicast address or range of IP multicast addresses to which access is being controlled. If entering a range, enter the low IP multicast address, a space, and the high IP multicast address. You can use the **range** command multiple times to enter multiple addresses or ranges of addresses. **Note** To delete an IP multicast address or range of IP multicast addresses, use the **no range** *ip multicast address* IGMP profile configuration command.
Step 6 **end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 7 **show ip igmp profile** *profile number* **Example:** Device# **show ip igmp profile 3**	Verifies the profile configuration.
Step 8 **show running-config** **Example:** Device# **show running-config**	Verifies your entries.

	Command or Action	**Purpose**
Step 9	**copy running-config startup-config** Example: Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Applying IGMP Profiles

To control access as defined in an IGMP profile, you have to apply the profile to the appropriate interfaces. You can apply IGMP profiles only to Layer 2 access ports; you cannot apply IGMP profiles to routed ports or SVIs. You cannot apply profiles to ports that belong to an EtherChannel port group. You can apply a profile to multiple interfaces, but each interface can have only one profile applied to it.

Follow these steps to apply an IGMP profile to a switch port:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp filter** *profile number*
5. **end**
6. **show running-config**
7. **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** Example: Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* Example: Device(config)# **interface gigabitethernet1/0/1**	Specifies the physical interface, and enters interface configuration mode. The interface must be a Layer 2 port that does not belong to an EtherChannel port group.

	Command or Action	**Purpose**
Step 4	**ip igmp filter** *profile number* **Example:** Device(config-if)# **ip igmp filter 321**	Applies the specified IGMP profile to the interface. The range is 1 to 4294967295. **Note** To remove a profile from an interface, use the **no ip igmp filter** *profile number* interface configuration command.
Step 5	**end** **Example:** Device(config-if)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Setting the Maximum Number of IGMP Groups

Follow these steps to set the maximum number of IGMP groups that a Layer 2 interface can join:

Before you begin

This restriction can be applied to Layer 2 ports only; you cannot set a maximum number of IGMP groups on routed ports or SVIs. You also can use this command on a logical EtherChannel interface but cannot use it on ports that belong to an EtherChannel port group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp max-groups** *number*
5. **end**
6. **show running-config interface** *interface-id*
7. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** Device(config)# **interface gigabitethernet1/0/2**	Specifies the interface to be configured, and enters interface configuration mode. The interface can be a Layer 2 port that does not belong to an EtherChannel group or a EtherChannel interface.
Step 4	**ip igmp max-groups** *number* **Example:** Device(config-if)# **ip igmp max-groups 20**	Sets the maximum number of IGMP groups that the interface can join. The range is 0 to 4294967294. The default is to have no maximum set. **Note** The device supports a maximum number of 4096 Layer 2 IGMP groups and 2048 Layer 3 IGMP groups.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config interface** *interface-id* **Example:** Device# **interface gigabitethernet1/0/1**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring the IGMP Throttling Action

After you set the maximum number of IGMP groups that a Layer 2 interface can join, you can configure an interface to replace the existing group with the new group for which the IGMP report was received.

Follow these steps to configure the throttling action when the maximum number of entries is in the forwarding table:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp max-groups action** {**deny** | **replace**}
5. **end**
6. **show running-config interface** *interface-id*
7. **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** Example: Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* Example: Device(config)# **interface gigabitethernet1/0/1**	Specifies the physical interface to be configured, and enters interface configuration mode. The interface can be a Layer 2 port that does not belong to an EtherChannel group or an EtherChannel interface. The interface cannot be a trunk port.
Step 4	**ip igmp max-groups action** {**deny**	**replace**} Example: Device(config-if)# **ip igmp max-groups action replace**	When an interface receives an IGMP report and the maximum number of entries is in the forwarding table, specifies the action that the interface takes: • **deny**—Drops the report. If you configure this throttling action, the entries that were previously in the forwarding table are not removed but are aged out. After these entries are aged out and the maximum number of entries is in the forwarding table, the device drops the next IGMP report received on the interface.

	Command or Action	Purpose
		• **replace**—Replaces the existing group with the new group for which the IGMP report was received. If you configure this throttling action, the entries that were previously in the forwarding table are removed. When the maximum number of entries is in the forwarding table, the device replaces a randomly selected entry with the received IGMP report.
		To prevent the device from removing the forwarding-table entries, you can configure the IGMP throttling action before an interface adds entries to the forwarding table.
		Note To return to the default action of dropping the report, use the **no ip igmp max-groups action** interface configuration command.
Step 5	**end** Example: Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config interface** *interface-id* Example: Device# **show running-config interface gigabitethernet1/0/1**	Verifies your entries.
Step 7	**copy running-config startup-config** Example: Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

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* **Example:** `device(config)# interface gigabitethernet 1`	Enters interface configuration mode. • 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.

	Command or Action	Purpose
		• 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. • 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:**	Applies the specified access list to IGMP reports.

	Command or Action	Purpose
	Device(config-if)# ip igmp access-group mygroup	
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:** Device(config-if)# end	Returns to privileged EXEC mode.

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:**	Enters global configuration mode.

	Command or Action	Purpose
	Device# configure terminal	
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.

Enabling or Disabling IGMP Snooping on a VLAN Interface

Follow these steps to enable IGMP snooping on a VLAN interface:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping vlan** *vlan-id*
4. **end**
5. **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** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	Device# **configure terminal**	
Step 3	**ip igmp snooping vlan** *vlan-id* **Example:** Device(config)# **ip igmp snooping vlan 7**	Enables IGMP snooping on the VLAN interface. The VLAN ID range is 1 to 1001 and 1006 to 4094. IGMP snooping must be globally enabled before you can enable VLAN snooping. **Note** To disable IGMP snooping on a VLAN interface, use the **no ip igmp snooping vlan** *vlan-id* global configuration command for the specified VLAN number.
Step 4	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Setting the Snooping Method

Multicast-capable router ports are added to the forwarding table for every Layer 2 multicast entry. The device learns of the ports through one of these methods:

- Snooping on IGMP queries, Protocol-Independent Multicast (PIM) packets

- Statically connecting to a multicast router port using the **ip igmp snooping mrouter** global configuration command

Beginning in privileged EXEC mode, follow these steps to alter the method in which a VLAN interface accesses a multicast router:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping vlan** *vlan-id* **mrouter interface** {**GigabitEthernet** | **Port-Channel** | **TenGigabitEthernet**}
4. **end**
5. **show ip igmp snooping**
6. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip igmp snooping vlan** *vlan-id* **mrouter interface** {**GigabitEthernet**	**Port-Channel**	**TenGigabitEthernet**} **Example:** Device(config)# **ip igmp snooping** **vlan 1 mrouter interface GigabitEthernet1/0/3**	Enables IGMP snooping on a VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.
Step 4	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 5	**show ip igmp snooping** **Example:** Device# **show ip igmp snooping**	Verifies the configuration.
Step 6	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring a Multicast Router Port

Perform these steps to add a multicast router port (enable a static connection to a multicast router) on the device.

Note Static connections to multicast routers are supported only on device ports.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping vlan** *vlan-id* **mrouter interface** *interface-id*
4. **end**
5. **show ip igmp snooping mrouter** [**vlan** *vlan-id*]
6. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip igmp snooping vlan** *vlan-id* **mrouter interface** *interface-id* **Example:** Device(config)# **ip igmp snooping vlan 5 mrouter interface gigabitethernet1/0/1**	Specifies the multicast router VLAN ID and the interface to the multicast router. • The VLAN ID range is 1 to 1001 and 1006 to 4094. • The interface can be a physical interface or a port channel. The port-channel range is 1 to 128. **Note** To remove a multicast router port from the VLAN, use the **no ip igmp snooping vlan** *vlan-id* **mrouter interface** *interface-id* global configuration command.
Step 4	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 5	**show ip igmp snooping mrouter** [**vlan** *vlan-id*] **Example:** Device# **show ip igmp snooping mrouter vlan 5**	Verifies that IGMP snooping is enabled on the VLAN interface.

	Command or Action	**Purpose**
Step 6	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring a Host Statically to Join a Group

Hosts or Layer 2 ports normally join multicast groups dynamically, but you can also statically configure a host on an interface.

Follow these steps to add a Layer 2 port as a member of a multicast group:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping vlan** *vlan-id* **static** *ip_address* **interface** *interface-id*
4. **end**
5. **show ip igmp snooping groups**
6. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip igmp snooping vlan** *vlan-id* **static** *ip_address* **interface** *interface-id* **Example:** Device(config)# **ip igmp snooping vlan 105 static 230.0.0.1 interface gigabitethernet1/0/1**	Statically configures a Layer 2 port as a member of a multicast group: • *vlan-id* is the multicast group VLAN ID. The range is 1 to 1001 and 1006 to 4094. • *ip-address* is the group IP address. • *interface-id* is the member port. It can be a physical interface or a port channel (1 to 128).

	Command or Action	Purpose
		Note To remove the Layer 2 port from the multicast group, use the **no ip igmp snooping vlan** *vlan-id* **static** *mac-address* **interface** *interface-id* global configuration command.
Step 4	**end** Example: Device(config)# **end**	Returns to privileged EXEC mode.
Step 5	**show ip igmp snooping groups** Example: Device# **show ip igmp snooping groups**	Verifies the member port and the IP address.
Step 6	**copy running-config startup-config** Example: Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Enabling IGMP Immediate Leave

When you enable IGMP Immediate Leave, the device immediately removes a port when it detects an IGMP Version 2 leave message on that port. You should use the Immediate-Leave feature only when there is a single receiver present on every port in the VLAN.

Note Immediate Leave is supported only on IGMP Version 2 hosts. IGMP Version 2 is the default version for the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping vlan** *vlan-id* **immediate-leave**
4. **end**
5. **show ip igmp snooping vlan** *vlan-id*
6. **end**

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	**configure terminal** Example: Device# `configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping vlan** *vlan-id* **immediate-leave** Example: Device(config)# `ip igmp snooping vlan 21` `immediate-leave`	Enables IGMP Immediate Leave on the VLAN interface. **Note** To disable IGMP Immediate Leave on a VLAN, use the **no ip igmp snooping vlan** *vlan-id* **immediate-leave** global configuration command.
Step 4	**end** Example: Device(config)# `end`	Returns to privileged EXEC mode.
Step 5	**show ip igmp snooping vlan** *vlan-id* Example: Device# `show ip igmp snooping vlan 21`	Verifies that Immediate Leave is enabled on the VLAN interface.
Step 6	**end** Example: Device(config)# `end`	Returns to privileged EXEC mode.

Configuring the IGMP Leave Timer

You can configure the leave time globally or on a per-VLAN basis. Follow these steps to enable the IGMP configurable-leave timer:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping last-member-query-interval** *time*
4. **ip igmp snooping vlan** *vlan-id* **last-member-query-interval** *time*
5. **end**
6. **show ip igmp snooping**

7. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip igmp snooping last-member-query-interval** *time* **Example:** Device(config)# **ip igmp snooping last-member-query-interval 1000**	Configures the IGMP leave timer globally. The range is 100 to 32767 milliseconds. The default leave time is 1000 milliseconds. **Note** To globally reset the IGMP leave timer to the default setting, use the **no ip igmp snooping last-member-query-interval** global configuration command.
Step 4	**ip igmp snooping vlan** *vlan-id* **last-member-query-interval** *time* **Example:** Device(config)# **ip igmp snooping vlan 210 last-member-query-interval 1000**	(Optional) Configures the IGMP leave time on the VLAN interface. The range is 100 to 32767 milliseconds. **Note** Configuring the leave time on a VLAN overrides the globally configured timer. **Note** To remove the configured IGMP leave-time setting from the specified VLAN, use the **no ip igmp snooping vlan** *vlan-id* **last-member-query-interval** global configuration command.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show ip igmp snooping** **Example:** Device# **show ip igmp snooping**	(Optional) Displays the configured IGMP leave time.

	Command or Action	**Purpose**
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring the IGMP Robustness-Variable

Use the following procedure to configure the IGMP robustness variable on the device.

The robustness variable is the integer used by IGMP snooping during calculations for IGMP messages. The robustness variable provides fine tuning to allow for expected packet loss.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping robustness-variable** *count*
4. **ip igmp snooping vlan** *vlan-id* **robustness-variable** *count*
5. **end**
6. **show ip igmp snooping**
7. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip igmp snooping robustness-variable** *count* **Example:** Device(config)# **ip igmp snooping robustness-variable 3**	Configures the IGMP robustness variable. The range is 1 to 3 times. The recommended value for the robustness variable is 2. Use this command to change the value of the robustness variable for IGMP snooping from the default (2) to a specified value.

	Command or Action	Purpose
Step 4	**ip igmp snooping vlan** *vlan-id* **robustness-variable** *count* **Example:** Device(config)#**ip igmp snooping vlan 100 robustness-variable 3**	(Optional) Configures the IGMP robustness variable on the VLAN interface. The range is 1 to 3 times. The recommended value for the robustness variable is 2. **Note** Configuring the robustness variable count on a VLAN overrides the globally configured value.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show ip igmp snooping** **Example:** Device# **show ip igmp snooping**	(Optional) Displays the configured IGMP robustness variable count.
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring the IGMP Last Member Query Count

To configure the number of times the device sends IGMP group-specific or group-source-specific (with IGMP version 3) query messages in response to receiving a group-specific or group-source-specific leave message, use this command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping last-member-query-count** *count*
4. **ip igmp snooping vlan** *vlan-id* **last-member-query-count** *count*
5. **end**
6. **show ip igmp snooping**
7. **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**	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	**ip igmp snooping last-member-query-count** *count*	Configures the IGMP last member query count. The range is 1 to 7 messages. The default is 2 messages.
	Example:	
	Device(config)# **ip igmp snooping last-member-query-count 3**	
Step 4	**ip igmp snooping vlan** *vlan-id* **last-member-query-count** *count*	(Optional) Configures the IGMP last member query count on the VLAN interface. The range is 1 to 7 messages.
	Example:	**Note** Configuring the last member query count on a VLAN overrides the globally configured timer.
	Device(config)#**ip igmp snooping vlan 100 last-member-query-count 3**	
Step 5	**end**	Returns to privileged EXEC mode.
	Example:	
	Device(config)# **end**	
Step 6	**show ip igmp snooping**	(Optional) Displays the configured IGMP last member query count.
	Example:	
	Device# **show ip igmp snooping**	
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.
	Example:	
	Device# **copy running-config startup-config**	

Configuring TCN-Related Commands

Controlling the Multicast Flooding Time After a TCN Event

You can configure the number of general queries by which multicast data traffic is flooded after a topology change notification (TCN) event. If you set the TCN flood query count to 1 the flooding stops after receiving

1 general query. If you set the count to 7, the flooding continues until 7 general queries are received. Groups are relearned based on the general queries received during the TCN event.

Some examples of TCN events are when the client location is changed and the receiver is on same port that was blocked but is now forwarding, and when a port goes down without sending a leave message.

Follow these steps to configure the TCN flood query count:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping tcn flood query count** *count*
4. **end**
5. **show ip igmp snooping**
6. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip igmp snooping tcn flood query count** *count* **Example:** Device(config)# **ip igmp snooping tcn flood query count 3**	Specifies the number of IGMP general queries for which the multicast traffic is flooded. The range is 1 to 10. The default, the flooding query count is 2. **Note** To return to the default flooding query count, use the **no ip igmp snooping tcn flood query count** global configuration command.
Step 4	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 5	**show ip igmp snooping** **Example:**	Verifies the TCN settings.

	Command or Action	**Purpose**
	Device# **show ip igmp snooping**	
Step 6	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Recovering from Flood Mode

When a topology change occurs, the spanning-tree root sends a special IGMP leave message (also known as global leave) with the group multicast address 0.0.0.0. However, you can enable the device to send the global leave message whether it is the spanning-tree root or not. When the router receives this special leave, it immediately sends general queries, which expedite the process of recovering from the flood mode during the TCN event. Leaves are always sent if the device is the spanning-tree root regardless of this configuration.

Follow these steps to enable sending of leave messages:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping tcn query solicit**
4. **end**
5. **show ip igmp snooping**
6. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip igmp snooping tcn query solicit** **Example:** Device(config)# **ip igmp snooping tcn query solicit**	Sends an IGMP leave message (global leave) to speed the process of recovering from the flood mode caused during a TCN event. By default, query solicitation is disabled.

	Command or Action	Purpose
		Note To return to the default query solicitation, use the **no ip igmp snooping tcn query solicit** global configuration command.
Step 4	**end** Example: Device(config)# **end**	Returns to privileged EXEC mode.
Step 5	**show ip igmp snooping** Example: Device# **show ip igmp snooping**	Verifies the TCN settings.
Step 6	**copy running-config startup-config** Example: Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Disabling Multicast Flooding During a TCN Event

When the device receives a TCN, multicast traffic is flooded to all the ports until 2 general queries are received. If the device has many ports with attached hosts that are subscribed to different multicast groups, this flooding might exceed the capacity of the link and cause packet loss. Follow these steps to control TCN flooding:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **no ip igmp snooping tcn flood**
5. **end**
6. **show ip igmp snooping**
7. **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.

	Command or Action	**Purpose**
Step 2	**configure terminal** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specifies the interface to be configured, and enters interface configuration mode.
Step 4	**no ip igmp snooping tcn flood** **Example:** Device(config-if)# **no ip igmp snooping tcn flood**	Disables the flooding of multicast traffic during a spanning-tree TCN event. By default, multicast flooding is enabled on an interface. **Note** To re-enable multicast flooding on an interface, use the **ip igmp snooping tcn flood** interface configuration command.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show ip igmp snooping** **Example:** Device# **show ip igmp snooping**	Verifies the TCN settings.
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring the IGMP Snooping Querier

Follow these steps to enable the IGMP snooping querier feature in a VLAN:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping querier**
4. **ip igmp snooping querier address** *ip_address*

5. **ip igmp snooping querier query-interval** *interval-count*
6. **ip igmp snooping querier tcn query** [**count** *count* | **interval** *interval*]
7. **ip igmp snooping querier timer expiry** *timeout*
8. **ip igmp snooping querier version** *version*
9. **end**
10. **show ip igmp snooping vlan** *vlan-id*
11. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip igmp snooping querier** **Example:** Device(config)# **ip igmp snooping querier**	Enables the IGMP snooping querier.
Step 4	**ip igmp snooping querier address** *ip_address* **Example:** Device(config)# **ip igmp snooping querier address 172.16.24.1**	(Optional) Specifies an IP address for the IGMP snooping querier. If you do not specify an IP address, the querier tries to use the global IP address configured for the IGMP querier. **Note** The IGMP snooping querier does not generate an IGMP general query if it cannot find an IP address on the device.
Step 5	**ip igmp snooping querier query-interval** *interval-count* **Example:** Device(config)# **ip igmp snooping querier query-interval 30**	(Optional) Sets the interval between IGMP queriers. The range is 1 to 18000 seconds.
Step 6	**ip igmp snooping querier tcn query** [**count** *count*	**interval** *interval*] **Example:**	(Optional) Sets the time between Topology Change Notification (TCN) queries. The count range is 1 to 10. The interval range is 1 to 255 seconds.

	Command or Action	Purpose
	Device(config)# **ip igmp snooping querier tcn query interval 20**	
Step 7	**ip igmp snooping querier timer expiry** *timeout* **Example:** Device(config)# **ip igmp snooping querier timer expiry 180**	(Optional) Sets the length of time until the IGMP querier expires. The range is 60 to 300 seconds.
Step 8	**ip igmp snooping querier version** *version* **Example:** Device(config)# **ip igmp snooping querier version 2**	(Optional) Selects the IGMP version number that the querier feature uses. Select 1 or 2.
Step 9	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 10	**show ip igmp snooping vlan** *vlan-id* **Example:** Device# **show ip igmp snooping vlan 30**	(Optional) Verifies that the IGMP snooping querier is enabled on the VLAN interface. The VLAN ID range is 1 to 1001 and 1006 to 4094.
Step 11	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Disabling IGMP Report Suppression

Follow these steps to disable IGMP report suppression:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip igmp snooping report-suppression**
4. **end**
5. **show ip igmp snooping**
6. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**no ip igmp snooping report-suppression** **Example:** Device(config)# **no ip igmp snooping report-suppression**	Disables IGMP report suppression. When report suppression is disabled, all IGMP reports are forwarded to the multicast routers. IGMP report suppression is enabled by default. When IGMP report supression is enabled, the device forwards only one IGMP report per multicast router query. **Note** To re-enable IGMP report suppression, use the **ip igmp snooping report-suppression** global configuration command.
Step 4	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 5	**show ip igmp snooping** **Example:** Device# **show ip igmp snooping**	Verifies that IGMP report suppression is disabled.
Step 6	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Monitoring IGMP

You can display specific statistics, such as the contents of IP routing tables, caches, and databases.

Note This release does not support per-route statistics.

You can display information to learn resource usage and solve network problems. You can also display information about node reachability and discover the routing path that packets of your device are taking through the network.

You can use any of the privileged EXEC commands in the following table to display various routing statistics.

Table 13: Commands for Displaying System and Network Statistics

Command	Purpose
ping [*group-name* | *group-address*]	Sends an ICMP Echo Request to a multicast group address.
show ip igmp filter	Displays IGMP filter information.
show ip igmp groups [*type-number* | *detail*]	Displays the multicast groups that are directly connected to the device and that were learned through IGMP.
show ip igmp interface [*type number*]	Displays multicast-related information about an interface.
show ip igmp membership [*name/group address* | **all** | **tracked**]	Displays IGMP membership information for forwarding.
show ip igmp profile [*profile_number*]	Displays IGMP profile information.
show ip igmp ssm-mapping [*hostname/IP address*]	Displays IGMP SSM mapping information.
show ip igmp static-group {**class-map** [**interface** [*type*]]	Displays static group information.
show ip igmp vrf	Displays the selected VPN routing/forwarding instance by name.

Monitoring IGMP Snooping Information

You can display IGMP snooping information for dynamically learned and statically configured router ports and VLAN interfaces. You can also display MAC address multicast entries for a VLAN configured for IGMP snooping.

Table 14: Commands for Displaying IGMP Snooping Information

Command	Purpose
show ip igmp snooping detail	Displays the operational state information.

Command	Purpose
show ip igmp snooping groups [**count**	[**vlan** *vlan-id* [*A.B.C.D*	**count**]]	Displays multicast table information for the device or about a specific parameter: • **count**—Displays the total number of groups. • **vlan**—Displays group information by VLAN ID.
show ip igmp snooping mrouter [**vlan** *vlan-id*]	Displays information on dynamically learned and manually configured multicast router interfaces. **Note** When you enable IGMP snooping, the device automatically learns the interface to which a multicast router is connected. These are dynamically learned interfaces. (Optional) Enter **vlan** *vlan-id* to display information for a single VLAN.
show ip igmp snooping querier [**detail**	**vlan** *vlan-id*]	Displays information about the IP address and receiving port for the most-recently received IGMP query messages in the VLAN. (Optional) Enter **detail** to display the detailed IGMP querier information in a VLAN. (Optional) Enter **vlan** *vlan-id* to display information for a single VLAN.
show ip igmp snooping [**vlan** *vlan-id* [**detail**]]	Displays the snooping configuration information for all VLANs on the device or for a specified VLAN. (Optional) Enter **vlan** *vlan-id* to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.

Monitoring IGMP Filtering and Throttling Configuration

You can display IGMP profile characteristics, and you can display the IGMP profile and maximum group configuration for all interfaces on the device or for a specified interface. You can also display the IGMP throttling configuration for all interfaces on the device or for a specified interface.

Table 15: Commands for Displaying IGMP Filtering and Throttling Configuration

Command	Purpose
show ip igmp profile [*profile number*]	Displays the specified IGMP profile or all the IGMP profiles defined on the device.

Command	Purpose
show running-config [**interface** *interface-id*]	Displays the configuration of the specified interface or the configuration of all interfaces on the device, including (if configured) the maximum number of IGMP groups to which an interface can belong and the IGMP profile applied to the interface.

Configuration Examples for IGMP

Example: Configuring the Device as a Member of a Multicast Group

This example shows how to enable the device to join multicast group 255.2.2.2:

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip igmp join-group 255.2.2.2
Device(config-if)#
```

Example: Controlling Access to Multicast Groups

To limit the number of joins on the interface, configure the port for filter which associates with the IGMP profile.

```
Device# configure terminal
Device(config)# ip igmp profile 10
Device(config-igmp-profile)# ?

IGMP profile configuration commands:
deny matching addresses are denied
exit Exit from igmp profile configuration mode
no Negate a command or set its defaults
permit matching addresses are permitted
range add a range to the set

Device(config-igmp-profile)# range 172.16.5.1
Device(config-igmp-profile)# exit
Device(config)#
Device(config)# interface gigabitEthernet 2/0/10
Device(config-if)# ip igmp filter 10
```

Examples: Configuring IGMP Snooping

This example shows how to enable a static connection to a multicast router:

```
Device# configure terminal
Device(config)# ip igmp snooping vlan 200 mrouter interface gigabitethernet1/0/2
Device(config)# end
```

This example shows how to statically configure a host on a port:

```
Device# configure terminal
Device(config)# ip igmp snooping vlan 105 static 224.2.4.12 interface gigabitethernet1/0/1
Device(config)# end
```

This example shows how to enable IGMP Immediate Leave on VLAN 130:

```
Device# configure terminal
Device(config)# ip igmp snooping vlan 130 immediate-leave
Device(config)# end
```

This example shows how to set the IGMP snooping querier source address to 10.0.0.64:

```
Device# configure terminal
Device(config)# ip igmp snooping querier 10.0.0.64
Device(config)# end
```

This example shows how to set the IGMP snooping querier maximum response time to 25 seconds:

```
Device# configure terminal
Device(config)# ip igmp snooping querier query-interval 25
Device(config)# end
```

This example shows how to set the IGMP snooping querier timeout to 60 seconds:

```
Device# configure terminal
Device(config)# ip igmp snooping querier timer expiry 60
Device(config)# end
```

This example shows how to set the IGMP snooping querier feature to Version 2:

```
Device# configure terminal
Device(config)# no ip igmp snooping querier version 2
Device(config)# end
```

Example: Configuring IGMP Profiles

This example shows how to create IGMP profile 4 allowing access to the single IP multicast address and how to verify the configuration. If the action was to deny (the default), it would not appear in the **show ip igmp profile** output display.

```
Device(config)# ip igmp profile 4
Device(config-igmp-profile)# permit
Device(config-igmp-profile)# range 229.9.9.0
Device(config-igmp-profile)# end
Device# show ip igmp profile 4
IGMP Profile 4
    permit
    range 229.9.9.0 229.9.9.0
```

Example: Applying IGMP Profile

This example shows how to apply IGMP profile 4 to a port:

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip igmp filter 4
Device(config-if)# end
```

Example: Setting the Maximum Number of IGMP Groups

This example shows how to limit to 25 the number of IGMP groups that a port can join:

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip igmp max-groups 25
Device(config-if)# end
```

Example: Interface Configuration as a Routed Port

This example shows how to configure an interface on the device as a routed port. This configuration is required on the interface for several IP multicast routing configuration procedures that require running the **no switchport** command.

```
Device configure terminal
Device(config)#  interface GigabitEthernet1/0/9
Device(config-if)# description interface to be use as routed port
Device(config-if)# no switchport
Device(config-if)# ip address 10.20.20.1 255.255.255.0
Device(config-if)# ip pim sparse-mode
Device(config-if)# ip igmp join-group 224.1.2.3 source 15.15.15.2
Device(config-if)# end
Device# configure terminal
Device# show run interface gigabitEthernet 1/0/9

Current configuration : 166 bytes
!
interface GigabitEthernet1/0/9
 no switchport
 ip address 10.20.20.1 255.255.255.0
 ip pim sparse-mode
 ip igmp static-group 224.1.2.3 source 15.15.15.2
end
```

Example: Interface Configuration as an SVI

This example shows how to configure an interface on the device as an SVI. This configuration is required on the interface for several IP multicast routing configuration procedures that require running the **no switchport** command.

```
Device(config)# interface vlan 150
Device(config-if)# ip address 10.20.20.1 255.255.255.0
Device(config-if)# ip pim sparse-mode
Device(config-if)# ip igmp join-group 224.1.2.3 source 15.15.15.2
Device(config-if)# end
```

```
Device# configure terminal
Device(config)#  ip igmp snooping vlan 20 static 224.1.2.3
interface gigabitEthernet 1/0/9
Device# show run interface vlan 150

Current configuration : 137 bytes
!
interface Vlan150
 ip address 10.20.20.1 255.255.255.0
 ip pim sparse-mode
 ip igmp static-group 224.1.2.3 source 15.15.15.2
end
```

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
```

Additional References for IGMP

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	

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
All the supported MIBs for this release.	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 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 History and Information for IGMP

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.

Configuring IGMP Proxy

Prerequisites for IGMP Proxy

- All devices on the IGMP UDL have the same subnet address. If all devices on the UDL cannot have the same subnet address, the upstream device must be configured with secondary addresses to match all of the subnets to which the downstream devices are attached.

- IP multicast is enabled and the PIM interfaces are configured.

Note   Use the following guidelines when configuring PIM interfaces for IGMP proxy:

- 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.

Information about IGMP Proxy

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.

Note Although the following illustration and example uses routers in the configuration, any device (router or switch) can be used.
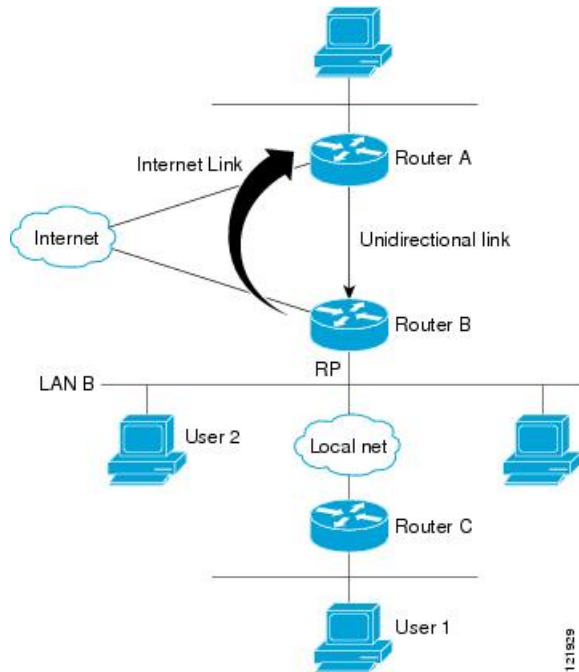
Scenario 1--Traditional UDLR Scenario (UDL Device with Directly Connected Receivers)

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 Configure IGMP Proxy

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: `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 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**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Device> enable	• 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:** Device(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	**Purpose**
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:** `Device# show ip igmp interface`	(Optional) Displays multicast-related information about an interface.

	Command or Action	Purpose
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 IGMP Proxy

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 for IGMP Proxy

The following sections provide references related to customizing IGMP.

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	
Overview of the IP multicast technology area	" IP Multicast Routing Technology Overview " module
Basic IP multicast concepts, configuration tasks, and examples	" Configuring Basic IP Multicast Routing" 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 History and Information for IGMP Proxy

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.

CHAPTER **8**

IGMP Explicit Tracking

This module describes the explicit tracking of hosts, groups, and channels for the Internet Group Management Protocol (IGMP).

- IGMP Explicit Tracking, on page 147

IGMP Explicit Tracking

This module describes the explicit tracking of hosts, groups, and channels for the Internet Group Management Protocol (IGMP).

Restrictions for IGMP Explicit Tracking

The following restrictions apply to this feature.

- If one or more hosts that supports only IGMP Version 1 or Version 2 are present on a network, the leave latencies for the multicast groups to which these hosts are joined will revert to the leave latencies of the IGMP version of the hosts—approximately 3 seconds for IGMP Version 2 and up to 180 seconds for IGMP Version 1. This condition affects only multicast groups to which these legacy hosts are actually joined at any given point in time. In addition, the membership reports for these multicast groups sent by IGMPv3 hosts may revert to IGMP Version 1 or Version 2 membership reports, thus disabling explicit tracking of those host memberships.

- Explicit tracking of IGMP Version 3 lite (IGMP v3lite) or URL Rendezvous Directory (URD) channel membership reports is not supported. Therefore, the leave latency for multicast groups sending traffic to hosts using IGMPv3 lite or URD will be determined by the leave latency of the version of IGMP configured on the hosts (for IGMPv3, the leave latency is typically 3 seconds when explicit tracking is not configured).

Information About IGMP Explicit Tracking

IGMP Explicit Tracking

The Internet Group Management Protocol (IGMP) is used by IP hosts to report their multicast group memberships to neighboring multicast devices. The IGMP Explicit Tracking feature enables a multicast device to explicitly track the membership of all multicast hosts in a particular multiaccess network. IGMP explicit tracking can be enabled globally and on Layer3 interfaces.

IP Multicast Routing Configuration Guide, Cisco IOS XE Fuji 16.9.x (Catalyst 3650 Switches)

147

The explicit tracking of hosts, groups, and channels enables the device to keep track of each individual host that is joined to a particular group or channel. The main benefits of this feature are that it provides minimal leave latencies, faster channel changing, and improved diagnostics capabilities for IGMP.

Minimal Leave Latencies

The main benefit of the explicit tracking of hosts, groups, and channels in IGMP is to allow minimal leave latencies when a host leaves a multicast group or channel. The length of time between a host wanting to leave and a device stopping traffic forwarding is called the IGMP leave latency. A device configured with IGMP Version 3 (IGMPv3) and explicit tracking can immediately stop forwarding traffic if the last host to request to receive traffic from the device indicates that it no longer wants to receive traffic. The leave latency is thus bound only by the packet transmission latencies in the multiaccess network and the processing time in the device.

In IGMP Version 2, when a device receives an IGMP leave message from a host, it must first send an IGMP group-specific query to learn if other hosts on the same multiaccess network are still requesting to receive traffic. If after a specific time (the default value is approximately 3 seconds) no host replies to the query, the device will then stop forwarding the traffic. This query process is required because, in IGMP Version 1 and 2, IGMP membership reports are suppressed if the same report is already sent by another host in the network. Therefore, it is impossible for the device to reliably know how many hosts on a multiaccess network are requesting to receive traffic.

Faster Channel Changing

In networks where bandwidth is constrained between multicast devices and hosts (like in xDSL deployments), the bandwidth between devices and hosts is typically large enough to only sustain, in general, N multicast streams to be received in parallel. In these deployments, each host will typically join to only one multicast stream and the overall number of allowed hosts will be limited to N. The effective leave latency in these environments defines the channel change time of the receiver application—a single host cannot receive the new multicast stream before forwarding of the old stream has stopped. If an application tries to change the channel faster than the leave latency, the application will overload the bandwidth of the access network, resulting in a temporary degradation of traffic flow for all hosts. The explicit tracking of hosts, groups, and channels in IGMP allows for minimal leave latencies, and thus allows for fast channel changing capabilities.

Improved Diagnostic Capabilities

The explicit tracking of hosts, groups, and channels in IGMP allows network administrators to easily determine which multicast hosts are joined to other multicast groups or channels.

How to Configure IGMP Explicit Tracking

Enabling Explicit Tracking Globally

You can enable explicit-tracking globally and on Layer 3 interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping vlan** *vlan-id* **explicit-tracking**
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 igmp snooping vlan** *vlan-id* **explicit-tracking** **Example:** Device(config)# ip igmp snooping vlan 1 explicit-tracking	Enables IGMP explicit host tracking.
Step 4	**exit** **Example:** Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Enabling Explicit Tracking on Layer 3 Interfaces

You can enable explicit-tracking globally and on Layer 3 interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip pim sparse-mode**
6. **ip igmp version 3**
7. **ip igmp explicit-tracking**
8. **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.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** Device(config)# interface vlan 77	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** Device(config-if)# ip address 10.1.1.1 255.255.255.254	Sets a primary or secondary IP address for an interface.
Step 5	**ip pim sparse-mode** **Example:** Device(config-if)# ip pim sparse-mode	Enables Protocol Independent Multicast (PIM) sparse mode on an interface.
Step 6	**ip igmp version 3** **Example:** Device(config-if)# ip igmp version 3	Configure Internet Group Management Protocol (IGMP) Version 3 (IGMPv3) on the device.
Step 7	**ip igmp explicit-tracking** **Example:** Device(config-if)# ip igmp explicit-tracking	Enables IGMP explicit host tracking.
Step 8	**exit** **Example:** Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for IGMP Explicit Tracking

Example: Enabling Explicit Tracking

The following example shows a basic configuration to enable IGMP explicit tracking globally:

```
Device# configure terminal
Device(config)# ip multicast routing
Device(config)# ip igmp snooping vlan 1 explicit-tracking
Device(config)# end
```

The following example shows a basic configuration to enable IGMP explicit tracking on Layer 3 interfaces:

```
Device# configure terminal
Device(config)# interface vlan 77
Device(config-if)# ip address 10.1.1.1 255.255.255.254
Device(config-if)# ip pim sparse-mode
Device(config-if)# ip igmp version 3
Device(config-if)# ip igmp explicit-tracking
```

```
Device(config-if)# end
```

Verifying IGMP Explicit Tracking

SUMMARY STEPS

1. **enable**
2. **show ip igmp snooping vlan** *vlan-ID*
3. **show ip igmp groups** *interface-type interface-number*
4. **show ip igmp membership tracked**
5. **show ip igmp snooping vlan** *vlan-ID*

DETAILED STEPS

Step 1   **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

• Enter your password if prompted.

Step 2   **show ip igmp snooping vlan** *vlan-ID*

Example:

```
Device# show ip igmp snooping vlan 77
```

Displays snooping information in a Catalyst VLAN.

```
Device# show  ip igmp  snooping vlan  77

Global IGMP Snooping configuration:
-------------------------------------
IGMP snooping              : Enabled
IGMPv3 snooping            : Enabled
Report suppression         : Enabled
TCN solicit query          : Disabled
TCN flood query count      : 2
Robustness variable        : 2
Last member query count    : 2
Last member query interval  : 1000

Vlan 77:
--------
IGMP snooping                  : Enabled
IGMPv2 immediate leave         : Disabled
Explicit host tracking         : Enabled
Multicast router learning mode : pim-dvmrp
CGMP interoperability mode     : IGMP_ONLY
Robustness variable            : 2
Last member query count        : 2
Last member query interval     : 1000
```

```
Device#
```

Step 3   **show ip igmp groups** *interface-type interface-number*

Example:

```
Device# show ip igmp groups GigabitEthernet 1/0/24
```

Displays the multicast groups that are directly connected to a device, and that are learned through IGMP.

```
show ip igmp groups GigabitEthernet 1/0/24

IGMP Connected Group Membership
Group Address    Interface            Uptime    Expires   Last Reporter   Group Accounted
203.0.113.245    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.244    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.247    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.246    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.241    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.240    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.243    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.242    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.253    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.252    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.221    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.254    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.249    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.248    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.251    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.250    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.228    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.229    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.230    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.231    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
203.0.113.224    GigabitEthernet1/0/24  00:00:35  stopped   10.34.34.2
```

Step 4   **show ip igmp membership tracked**

Example:

```
Device# show ip igmp membership tracked
```

Displays the multicast groups with the explicit tracking feature enabled.

```
Device# show ip igmp  membership tracked

Flags: A  - aggregate, T - tracked
       L  - Local, S - static, V - virtual, R - Reported through v3
       I - v3lite, U - Urd, M - SSM (S,G) channel
       1,2,3 - The version of IGMP, the group is in
Channel/Group-Flags:
       / - Filtering entry (Exclude mode (S,G), Include mode (G))
Reporter:
       <mac-or-ip-address> - last reporter if group is not explicitly tracked
       <n>/<m>      - <n> reporter in include mode, <m> reporter in exclude

 Channel/Group                Reporter         Uptime   Exp.  Flags  Interface
 *,203.0.113.10               1/0              00:20:46 stop  3AT    Gi1/0/24
 192.168.0.2,203.0.113.10     10.34.34.2       00:20:46 02:59 T      Gi1/0/24
 *,203.0.113.11               1/0              00:20:46 stop  3AT    Gi1/0/24
 192.168.0.2,203.0.113.11     10.34.34.2       00:20:46 02:59 T      Gi1/0/24
 *,203.0.113.14               1/0              00:20:46 stop  3AT    Gi1/0/24
 192.168.0.2,203.0.113.14     10.34.34.2       00:20:46 02:59 T      Gi1/0/24
```

```
*,203.0.113.15              1/0              00:20:46 stop  3AT   Gi1/0/24
192.168.0.2,203.0.113.15    10.34.34.2       00:20:46 02:59 T     Gi1/0/24
*,203.0.113.12              1/0              00:20:46 stop  3AT   Gi1/0/24
192.168.0.2,203.0.113.12    10.34.34.2       00:20:46 02:59 T     Gi1/0/24
*,203.0.113.13              1/0              00:20:46 stop  3AT   Gi1/0/24
192.168.0.2,203.0.113.13    10.34.34.2       00:20:46 02:59 T     Gi1/0/24
*,203.0.113.19              1/0              00:20:46 stop  3AT   Gi1/0/24
192.168.0.2,203.0.113.19    10.34.34.2       00:20:46 02:59 T     Gi1/0/24
*,203.0.113.18              1/0              00:20:46 stop  3AT   Gi1/0/24
192.168.0.2,203.0.113.18    10.34.34.2       00:20:46 02:59 T     Gi1/0/24
*,203.0.113.17              1/0              00:20:46 stop  3AT   Gi1/0/24
192.168.0.2,203.0.113.17    10.34.34.2       00:20:46 02:59 T     Gi1/0/24
*,203.0.113.16              1/0              00:20:46 stop  3AT   Gi1/0/24
192.168.0.2,203.0.113.16    10.34.34.2       00:20:46 02:59 T     Gi1/0/24
*,203.0.113.40              0/1              00:20:48 02:16 3LAT  Gi1/0/24
*,209.165.201.1             10.34.34.1       00:20:48 02:16 3LT   Gi1/0/24
Device#
```

Step 5 **show ip igmp snooping vlan** *vlan-ID*

Example:

```
Device# show ip igmp snooping vlan 77
```

Displays the IGMP snooping configuration on a VLAN.

```
Device# show  ip igmp  snooping vlan  77

Global IGMP Snooping configuration:
-------------------------------------------
IGMP snooping              : Enabled
IGMPv3 snooping            : Enabled
Report suppression         : Enabled
TCN solicit query          : Disabled
TCN flood query count      : 2
Robustness variable        : 2
Last member query count    : 2
Last member query interval : 1000

Vlan 77:
--------
IGMP snooping                 : Enabled
IGMPv2 immediate leave        : Disabled
Explicit host tracking        : Enabled
Multicast router learning mode : pim-dvmrp
CGMP interoperability mode    : IGMP_ONLY
Robustness variable           : 2
Last member query count       : 2
Last member query interval    : 1000
Device#
```

Feature History for IGMP Explicit Tracking

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 16: Feature Information for IGMP Explicit Tracking

Feature Name	Release	Feature Information
IGMP Explicit Tracking	Cisco IOS XE Everest 16.6.1	This module describes the explicit tracking of hosts, groups, and channels for IGMP. In Cisco IOS XE Everest 16.6.1, this feature was implemented on the following platforms: • Cisco Catalyst 3650 Series Switches • Cisco Catalyst 9300 Series Switches • Cisco Catalyst 9500 Series Switches

C H A P T E R **9**

Configuring Protocol Independent Multicast (PIM)

Prerequisites for PIM

- Before you begin the PIM configuration process, decide which PIM mode to use. This is based on the applications you intend to support on your network. Use the following guidelines:

  - 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.

- Before you configure PIM stub routing, check that you have met these conditions:

  - You must have IP multicast routing configured on both the stub router and the central router. You must also have PIM modeconfigured on the uplink interface of the stub router.

  - You must also configure either Enhanced Interior Gateway Routing Protocol (EIGRP) stub routing or Open Shortest Path First (OSPF) stub routing on the device.

  - The PIM stub router does not route the transit traffic between the distribution routers. Unicast (EIGRP) stub routing enforces this behavior. You must configure unicast stub routing to assist the PIM stub router behavior.

Restrictions for PIM

The following are the restrictions for configuring PIM:

- PIM is not supported when running the LAN Base feature set.

PIMv1 and PIMv2 Interoperability

To avoid misconfiguring multicast routing on your device, review the information in this section.

The Cisco PIMv2 implementation provides interoperability and transition between Version 1 and Version 2, although there might be some minor problems.

You can upgrade to PIMv2 incrementally. PIM Versions 1 and 2 can be configured on different routers and multilayer devices within one network. Internally, all routers and multilayer devices on a shared media network must run the same PIM version. Therefore, if a PIMv2 device detects a PIMv1 device, the Version 2 device downgrades itself to Version 1 until all Version 1 devices have been shut down or upgraded.

PIMv2 uses the BSR to discover and announce RP-set information for each group prefix to all the routers and multilayer devices in a PIM domain. PIMv1, together with the Auto-RP feature, can perform the same tasks as the PIMv2 BSR. However, Auto-RP is a standalone protocol, separate from PIMv1, and is a proprietary Cisco protocol. PIMv2 is a standards track protocol in the IETF.

> **Note** We recommend that you use PIMv2. The BSR function interoperates with Auto-RP on Cisco routers and multilayer devices.

When PIMv2 devices interoperate with PIMv1 devices, Auto-RP should have already been deployed. A PIMv2 BSR that is also an Auto-RP mapping agent automatically advertises the RP elected by Auto-RP. That is, Auto-RP sets its single RP on every router or multilayer device in the group. Not all routers and devices in the domain use the PIMv2 hash function to select multiple RPs.

Sparse-mode groups in a mixed PIMv1 and PIMv2 region are possible because the Auto-RP feature in PIMv1 interoperates with the PIMv2 RP feature. Although all PIMv2 devices can also use PIMv1, we recommend that the RPs be upgraded to PIMv2. To ease the transition to PIMv2, we recommend:

- Using Auto-RP throughout the region.

If Auto-RP is not already configured in the PIMv1 regions, configure Auto-RP.

Restrictions for Configuring PIM Stub Routing

- Only directly connected multicast (IGMP) receivers and sources are allowed in the Layer 2 access domains. The PIM protocol is not supported in access domains.

- In a network using PIM stub routing, the only allowable route for IP traffic to the user is through a device that is configured with PIM stub routing.

- The redundant PIM stub router topology is not supported. Only the nonredundant access router topology is supported by the PIM stub feature.

• PIM stub routing is supported when running the IP Base and IP Services feature sets.

Restrictions for Configuring Auto-RP and BSR

Take into consideration your network configuration, and the following restrictions when configuring Auto-RP and BSR:

Restrictions for Configuring Auto-RP

The following are restrictions for configuring Auto-RP (if used in your network configuration):

• Auto-RP is not supported when running the LAN Base feature set.

• If routed interfaces are configured in sparse mode, Auto-RP can still be used if all devices are configured with a manual RP address for the Auto-RP groups.

• If routed interfaces are configured in sparse mode and you enter the **ip pim autorp listener** global configuration command, Auto-RP can still be used even if all devices are not configured with a manual RP address for the Auto-RP groups.

Restrictions for Configuring BSR

The following are the restrictions for configuring BSR (if used in your network configuration):

• Configure the candidate BSRs as the RP-mapping agents for Auto-RP.

• For group prefixes advertised through Auto-RP, the PIMv2 BSR mechanism should not advertise a subrange of these group prefixes served by a different set of RPs. In a mixed PIMv1 and PIMv2 domain, have backup RPs serve the same group prefixes. This prevents the PIMv2 DRs from selecting a different RP from those PIMv1 DRs, due to the longest match lookup in the RP-mapping database.

Restrictions and Guidelines for Configuring Auto-RP and BSR

The following are restrictions for configuring Auto-RP and BSR (if used in your network configuration):

• If your network is all Cisco routers and multilayer devices, you can use either Auto-RP or BSR.

• If you have non-Cisco routers in your network, you must use BSR.

• If you have Cisco PIMv1 and PIMv2 routers and multilayer devices and non-Cisco routers, you must use both Auto-RP and BSR. If your network includes routers from other vendors, configure the Auto-RP mapping agent and the BSR on a Cisco PIMv2 device. Ensure that no PIMv1 device is located in the path a between the BSR and a non-Cisco PIMv2 device.

> **Note** There are two approaches to using PIMv2. You can use Version 2 exclusively in your network or migrate to Version 2 by employing a mixed PIM version environment.

• Because bootstrap messages are sent hop-by-hop, a PIMv1 device prevents these messages from reaching all routers and multilayer devices in your network. Therefore, if your network has a PIMv1 device in it and only Cisco routers and multilayer devices, it is best to use Auto-RP.

- If you have a network that includes non-Cisco routers, configure the Auto-RP mapping agent and the BSR on a Cisco PIMv2 router or multilayer device. Ensure that no PIMv1 device is on the path between the BSR and a non-Cisco PIMv2 router.

- If you have non-Cisco PIMv2 routers that need to interoperate with Cisco PIMv1 routers and multilayer devices, both Auto-RP and a BSR are required. We recommend that a Cisco PIMv2 device be both the Auto-RP mapping agent and the BSR.

Restrictions for Auto-RP Enhancement

The simultaneous deployment of Auto-RP and bootstrap router (BSR) is not supported.

Information About PIM

Protocol Independent Multicast Overview

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 is defined in RFC 4601, Protocol Independent Multicast - Sparse Mode (PIM-SM)

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.

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. 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 162 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.

Multicast Source Discovery Protocol (MSDP)

Multicast Source Discovery Protocol (MSDP) is used for inter-domain source discovery when PIM SM is used. Each PIM administrative domain has its own RP. In order for the RP in one domain to signal new sources to the RP in the other domain, MSDP is used.

When RP in a domain receives a PIM register message for a new source, with MSDP configured it sends a new source-active (SA) message to all its MSDP peers in other domains. Each intermediate MSDP peer floods this SA message away from the originating RP. The MSDP peers install this SA message in their MSDP

sa-cache. If the RPs in other domains have any join requests for the group in the SA message (indicated by the presence of a (*,G) entry with non empty outgoing interface list), the domain is interested in the group, and the RP triggers an (S,G) join toward the source.

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.

PIM Versions

PIMv2 includes these improvements over PIMv1:

- A single, active rendezvous point (RP) exists per multicast group, with multiple backup RPs. This single RP compares to multiple active RPs for the same group in PIMv1.

- A bootstrap router (BSR) provides a fault-tolerant, automated RP discovery and distribution function that enables routers and multilayer devices to dynamically learn the group-to-RP mappings.

- PIM join and prune messages have more flexible encoding for multiple address families.

- A more flexible hello packet format replaces the query packet to encode current and future capability options.

- Register messages sent to an RP specify whether they are sent by a border router or a designated router.

- PIM packets are no longer inside IGMP packets; they are standalone packets.

PIM Stub Routing

The PIM stub routing feature, available in all of the device software images, reduces resource usage by moving routed traffic closer to the end user.

The PIM stub routing feature supports multicast routing between the distribution layer and the access layer. It supports two types of PIM interfaces, uplink PIM interfaces, and PIM passive interfaces. A routed interface configured with the PIM passive mode does not pass or forward PIM control traffic, it only passes and forwards IGMP traffic.

In a network using PIM stub routing, the only allowable route for IP traffic to the user is through a device that is configured with PIM stub routing. PIM passive interfaces are connected to Layer 2 access domains, such as VLANs, or to interfaces that are connected to other Layer 2 devices. Only directly connected multicast (IGMP) receivers and sources are allowed in the Layer 2 access domains. The PIM passive interfaces do not send or process any received PIM control packets.

When using PIM stub routing, you should configure the distribution and remote routers to use IP multicast routing and configure only the device as a PIM stub router. The device does not route transit traffic between distribution routers. You also need to configure a routed uplink port on the device. The device uplink port cannot be used with SVIs. If you need PIM for an SVI uplink port, you should upgrade to the Network Advantage license.

Note   You must also configure EIGRP stub routing when configuring PIM stub routing on the device

The redundant PIM stub router topology is not supported. The redundant topology exists when there is more than one PIM router forwarding multicast traffic to a single access domain. PIM messages are blocked, and the PIM asset and designated router election mechanisms are not supported on the PIM passive interfaces. Only the nonredundant access router topology is supported by the PIM stub feature. By using a nonredundant topology, the PIM passive interface assumes that it is the only interface and designated router on that access domain.

Figure 9: PIM Stub Router Configuration

In the following figure, the Device A routed uplink port 25 is connected to the router and PIM stub routing is enabled on the VLAN 100 interfaces and on Host 3. This configuration allows the directly connected hosts to receive traffic from multicast source 200.1.1.3.
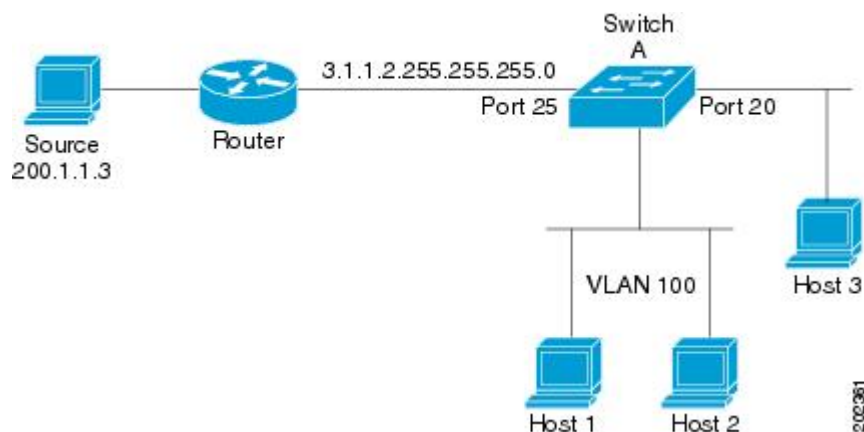
IGMP Helper

PIM stub routing moves routed traffic closer to the end user and reduces network traffic. You can also reduce traffic by configuring a stub router (switch) with the IGMP helper feature.

You can configure a stub router (switch) with the **ip igmp helper-address** *ip-address* interface configuration command to enable the switch to send reports to the next-hop interface. Hosts that are not directly connected to a downstream router can then join a multicast group sourced from an upstream network. The IGMP packets from a host wanting to join a multicast stream are forwarded upstream to the next-hop device when this feature is configured. When the upstream central router receives the helper IGMP reports or leaves, it adds or removes the interfaces from its outgoing interface list for that group.

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.

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 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. 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.

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.

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.

Multicast Boundaries

Administratively-scoped boundaries can be used to limit the forwarding of multicast traffic outside of a domain or subdomain. This approach uses a special range of multicast addresses, called administratively-scoped addresses, as the boundary mechanism. If you configure an administratively-scoped boundary on a routed interface, multicast traffic whose multicast group addresses fall in this range cannot enter or exit this interface, which provides a firewall for multicast traffic in this address range.

> **Note**
> Multicast boundaries and TTL thresholds control the scoping of multicast domains; however, TTL thresholds are not supported by the device. You should use multicast boundaries instead of TTL thresholds to limit the forwarding of multicast traffic outside of a domain or a subdomain.

Figure 10: Administratively-Scoped Boundaries

The following figure shows that Company XYZ has an administratively-scoped boundary set for the multicast address range 239.0.0.0/8 on all routed interfaces at the perimeter of its network. This boundary prevents any multicast traffic in the range 239.0.0.0 through 239.255.255.255 from entering or leaving the network. Similarly, the engineering and marketing departments have an administratively-scoped boundary of 239.128.0.0/16 around the perimeter of their networks. This boundary prevents multicast traffic in the range of 239.128.0.0

through 239.128.255.255 from entering or leaving their respective networks.
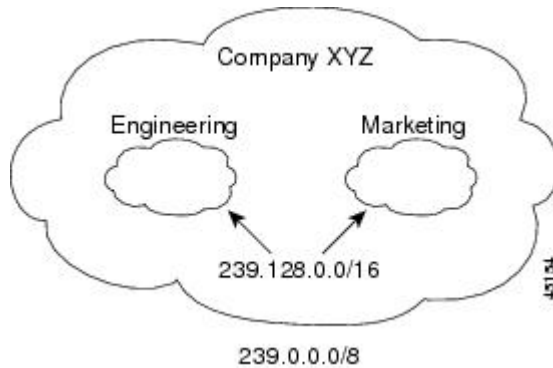
You can define an administratively-scoped boundary on a routed interface for multicast group addresses. A standard access list defines the range of addresses affected. When a boundary is defined, no multicast data packets are allowed to flow across the boundary from either direction. The boundary allows the same multicast group address to be reused in different administrative domains.

The 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 then be reused in domains administered by different organizations. The addresses 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.

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.

Auto-RP Benefits

Auto-RP uses IP multicast to automate the distribution of group-to-RP mappings to all Cisco routers and multilayer devices in a PIM network. Auto-RP has these benefits:

- Easy to use multiple RPs within a network to serve different group ranges.

- Provides load splitting among different RPs and arrangement of RPs according to the location of group participants.

- Avoids inconsistent, manual RP configurations on every router and multilayer device in a PIM network, which can cause connectivity problems.

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.

PIMv2 Bootstrap Router

PIMv2 Bootstrap Router (BSR) is another method to distribute group-to-RP mapping information to all PIM routers and multilayer devices in the network. It eliminates the need to manually configure RP information in every router and device in the network. However, instead of using IP multicast to distribute group-to-RP mapping information, BSR uses hop-by-hop flooding of special BSR messages to distribute the mapping information.

The BSR is elected from a set of candidate routers and devices in the domain that have been configured to function as BSRs. The election mechanism is similar to the root-bridge election mechanism used in bridged LANs. The BSR election is based on the BSR priority of the device contained in the BSR messages that are sent hop-by-hop through the network. Each BSR device examines the message and forwards out all interfaces only the message that has either a higher BSR priority than its BSR priority or the same BSR priority, but with a higher BSR IP address. Using this method, the BSR is elected.

The elected BSR sends BSR messages with a TTL of 1. Neighboring PIMv2 routers or multilayer devices receive the BSR message and multicast it out all other interfaces (except the one on which it was received) with a TTL of 1. In this way, BSR messages travel hop-by-hop throughout the PIM domain. Because BSR messages contain the IP address of the current BSR, the flooding mechanism enables candidate RPs to automatically learn which device is the elected BSR.

Candidate RPs send candidate RP advertisements showing the group range for which they are responsible to the BSR, which stores this information in its local candidate-RP cache. The BSR periodically advertises the contents of this cache in BSR messages to all other PIM devices in the domain. These messages travel hop-by-hop through the network to all routers and devices, which store the RP information in the BSR message in their local RP cache. The routers and devices select the same RP for a given group because they all use a common RP hashing algorithm.

PIM Domain Border

As IP multicast becomes more widespread, the chance of one PIMv2 domain bordering another PIMv2 domain increases. Because two domains probably do not share the same set of RPs, BSR, candidate RPs, and candidate BSRs, you need to constrain PIMv2 BSR messages from flowing into or out of the domain. Allowing messages to leak across the domain borders could adversely affect the normal BSR election mechanism and elect a single BSR across all bordering domains and comingle candidate RP advertisements, resulting in the election of RPs in the wrong domain.

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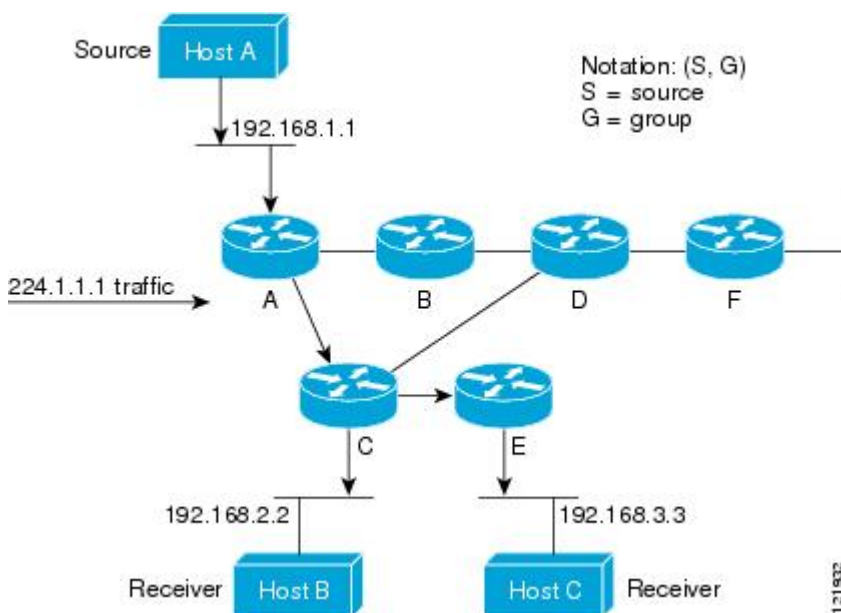
Using standard notation, the SPT for the example shown in the figure would be (192.168.1.1, 224.1.1.1).

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 11: Shared Tree
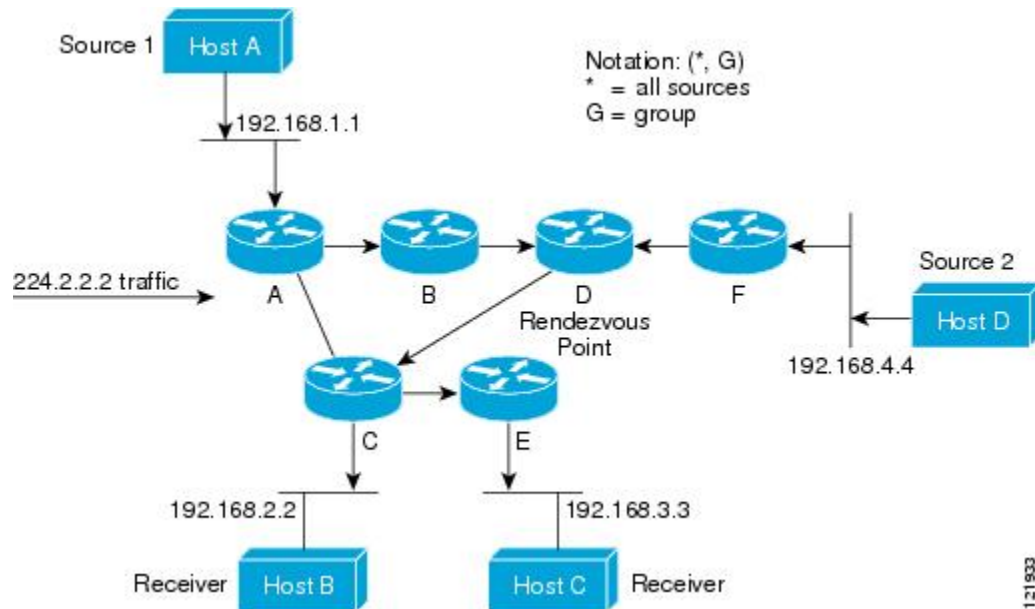
In this example, multicast traffic from the sources, Hosts A and D, travels to the root (Router D) and then down the shared tree to the two receivers, Hosts B and C. Because all sources in the multicast group use a common shared tree, a wildcard notation written as (*, G), pronounced "star comma G", 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.

PIM Shared Tree and Source Tree

By default, members of a group receive data from senders to the group across a single data-distribution tree rooted at the RP.

Figure 12: Shared Tree and Source Tree (Shortest-Path Tree)

The following figure shows this type of shared-distribution tree. Data from senders is delivered to the RP for
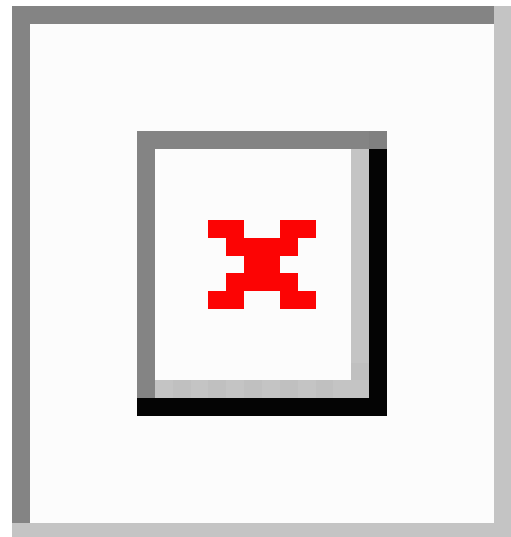
distribution to group members joined to the shared tree.

If the data rate warrants, leaf routers (routers without any downstream connections) on the shared tree can use 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 devices to a source tree upon receiving the first data packet from a source.

This process describes the move from a shared tree to a source tree:

1. A receiver joins a group; leaf Router C sends a join message toward the RP.

2. The RP puts a link to Router C in its outgoing interface list.

3. A source sends data; Router A encapsulates the data in a register message and sends it to the RP.

4. The RP forwards the data down the shared tree to Router C and sends a join message toward the source. At this point, data might arrive twice at Router C, once encapsulated and once natively.

5. When data arrives natively (unencapsulated) at the RP, it 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 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 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. You can configure the PIM device to stay on the shared tree.

The change from shared to source tree happens when the first data packet arrives at the last-hop router. This change depends upon the threshold that is configured by using the **ip pim spt-threshold** global configuration command.

The shortest-path tree requires more memory than the shared tree but reduces delay. You may want to postpone its use. Instead of allowing the leaf router to immediately move to the shortest-path tree, you can specify that the traffic must first reach a threshold.

You can configure when a PIM leaf router 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 multilayer switch triggers a PIM join message toward the source to construct a source tree (shortest-path tree). If the traffic rate from the source drops below the threshold value, the leaf router switches back to the shared tree and sends a prune message toward the source.

You can specify to which groups the shortest-path tree threshold applies by using a group list (a standard access list). If a value of 0 is specified or if the group list is not used, the threshold applies to all groups.

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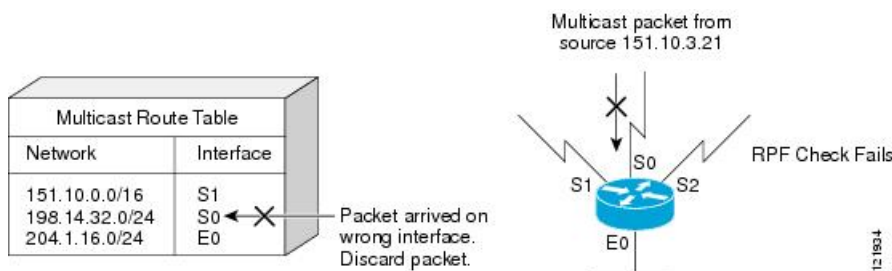
As the figure illustrates, a multicast packet from source 151.10.3.21 is received on serial interface 0 (S0). A check of the unicast route table shows that S1 is the interface this router would use to forward unicast data to 151.10.3.21. Because the packet has arrived on interface S0, the packet is discarded.

The figure shows an example of a successful RPF check.

Figure 14: RPF Check Succeeds
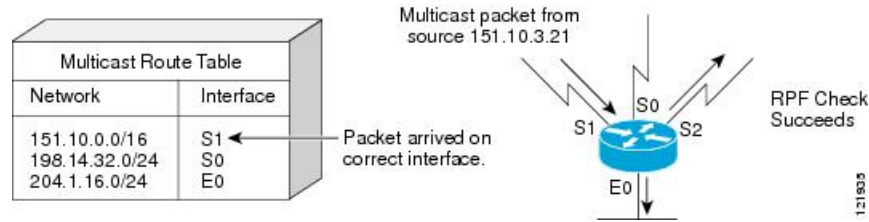
In this example, the multicast packet has arrived on interface S1. The router refers to the unicast routing table and finds that S1 is the correct interface. The RPF check passes, and the packet is forwarded.

PIM uses both source trees and RP-rooted shared trees to forward datagrams. The RPF check is performed differently for each:

- If a PIM router or multilayer device has a source-tree state (that is, an (S, G) entry is present in the multicast routing table), it performs the RPF check against the IP address of the source of the multicast packet.

- If a PIM router or multilayer device has a shared-tree state (and no explicit source-tree state), it performs the RPF check on the RP address (which is known when members join the group).

| Note | DVMRP is not supported on the device. |

Sparse-mode PIM uses the RPF lookup function to decide 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.

Default PIM Routing Configuration

This table displays the default PIM routing configuration for the device.

Table 17: Default Multicast Routing Configuration

Feature	Default Setting
Multicast routing	Disabled on all interfaces.
PIM version	Version 2.
PIM mode	No mode is defined.
PIM stub routing	None configured.
PIM RP address	None configured.
PIM domain border	Disabled.
PIM multicast boundary	None.

Feature	Default Setting
Candidate BSRs	Disabled.
Candidate RPs	Disabled.
Shortest-path tree threshold rate	0 kb/s.
PIM router query message interval	30 seconds.

How to Configure PIM

Enabling PIM Stub Routing

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip pim passive**
5. **end**
6. **show ip pim interface**
7. **show ip igmp groups detail**
8. **show ip mroute**
9. **show running-config**
10. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:**	Specifies the interface on which you want to enable PIM stub routing, and enters interface configuration mode.

	Command or Action	Purpose
	Device(config)# **interface gigabitethernet 1/0/1**	The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. These interfaces must have IP addresses assigned to them.
Step 4	**ip pim passive** Example: Device(config-if)# **ip pim passive**	Configures the PIM stub feature on the interface.
Step 5	**end** Example: Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show ip pim interface** Example: Device# **show ip pim interface**	(Optional) Displays the PIM stub that is enabled on each interface.
Step 7	**show ip igmp groups detail** Example: Device# **show ip igmp groups detail**	(Optional) Displays the interested clients that have joined the specific multicast source group.
Step 8	**show ip mroute** Example: Device# **show ip mroute**	(Optional) Displays the IP multicast routing table.
Step 9	**show running-config** Example: Device# **show running-config**	Verifies your entries.
Step 10	**copy running-config startup-config** Example:	(Optional) Saves your entries in the configuration file.

Command or Action	Purpose
Device# **copy running-config startup-config**	

Configuring a Rendezvous Point

You must have a rendezvous point (RP), if the interface is in sparse-dense mode and if you want to handle the group as a sparse group. You can use these methods:

- By manually assigning an RP to multicast groups.

- As a standalone, Cisco-proprietary protocol separate from PIMv1, which includes:

   - Setting up Auto-RP in a new internetwork

   - Adding Auto-RP to an existing sparse-mode cloud

   - Preventing join messages to false RPs

   - Filtering incoming RP announcement messages

- By using a standards track protocol in the Internet Engineering Task Force (IETF), which includes configuring PIMv2 BSR .

Note You can use Auto-RP, BSR, or a combination of both, depending on the PIM version that you are running and the types of routers in your network. For information about working with different PIM versions in your network, see PIMv1 and PIMv2 Interoperability, on page 156.

Manually Assigning an RP to Multicast Groups

If the rendezvous point (RP) for a group is learned through a dynamic mechanism (such as Auto-RP or BSR), you need not perform this task for that RP.

Senders of multicast traffic announce their existence through register messages received from the source first-hop router (designated router) and forwarded to the RP. Receivers of multicast packets use RPs to join a multicast group by using explicit join messages.

Note RPs are not members of the multicast group; they serve as a *meeting place* for multicast sources and group members.

You can configure a single RP for multiple groups defined by an access list. If there is no RP configured for a group, the multilayer device responds to the group as dense and uses the dense-mode PIM techniques.

This procedure is optional.

SUMMARY STEPS

   1. **enable**

2. **configure terminal**
3. **ip pim rp-address** *ip-address* [*access-list-number*] [**override**]
4. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]
5. **end**
6. **show running-config**
7. **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** Example: Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip pim rp-address** *ip-address* [*access-list-number*] [**override**] Example: Device(config)# **ip pim rp-address 10.1.1.1 20 override**	Configures the address of a PIM RP. By default, no PIM RP address is configured. You must configure the IP address of RPs on all routers and multilayer devices (including the RP). **Note** If there is no RP configured for a group, the device treats the group as dense, using the dense-mode PIM techniques. 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. The access list conditions specify for which groups the device is an RP. • For *ip-address*, enter the unicast address of the RP in dotted-decimal notation. • (Optional) For *access-list-number*, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups. • (Optional) The **override** keyword indicates that if there is a conflict between the RP configured with this command and one learned by Auto-RP or BSR, the RP configured with this command prevails.
Step 4	**access-list** *access-list-number* {**deny**	**permit**} *source* [*source-wildcard*] Example:	Creates a standard access list, repeating the command as many times as necessary.

Command or Action	Purpose
Device(config)# **access-list 25 permit 10.5.0.1 255.224.0.0**	• For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access if the conditions are matched. • The **permit** keyword permits access if the conditions are matched. • For *source*, enter the multicast group address for which the RP should be used. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. The access list is always terminated by an implicit deny statement for everything.
Step 5 **end** Example: Device(config)# **end**	Returns to privileged EXEC mode.
Step 6 **show running-config** Example: Device# **show running-config**	Verifies your entries.
Step 7 **copy running-config startup-config** Example: Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Setting Up Auto-RP in a New Internetwork

Note Omit Step 3 in the following procedure, if you want to configure a PIM router as the RP for the local group.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **ip pim send-rp-announce** *interface-id* **scope** *ttl* **group-list** *access-list-number* **interval** *seconds*

5.   **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]

6.   **ip pim send-rp-discovery scope** *ttl*

7.   **end**

8.   **show running-config**

9.   **show ip pim rp mapping**

10.   **show ip pim rp**

11.   **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	**show running-config** **Example:** Device# **show running-config**	Verifies that a default RP is already configured on all PIM devices and the RP in the sparse-mode network. It was previously configured with the **ip pim rp-address** global configuration command. **Note** This step is not required for spare-dense-mode environments. The selected RP should have good connectivity and be available across the network. Use this RP for the global groups (for example, 224.x.x.x and other global groups). Do not reconfigure the group address range that this RP serves. RPs dynamically discovered through Auto-RP take precedence over statically configured RPs. Assume that it is desirable to use a second RP for the local groups.
Step 3	**configure terminal** **Example:** Device# **configure terminal**	Enters global configuration mode.

Step 4	**ip pim send-rp-announce** *interface-id* **scope** *ttl* **group-list** *access-list-number* **interval** *seconds*
	
	Example:
	
	Device(config)# **ip pim send-rp-announce gigabitethernet**
	1/0/5 scope 20 group-list 10 interval 120
		Configures another PIM device to be the candidate RP for local groups.
		
		   • For *interface-id*, enter the interface type and number that identifies the RP address. Valid interfaces include physical ports, port channels, and VLANs.
		
		   • For **scope** *ttl*, specify the time-to-live value in hops. Enter a hop count that is high enough so that the RP-announce messages reach all mapping agents in the network. There is no default setting. The range is 1 to 255.

	Command or Action	**Purpose**
		• For **group-list** *access-list-number*, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups.
		• For **interval** *seconds*, specify how often the announcement messages must be sent. The default is 60 seconds. The range is 1 to 16383.
Step 5	**access-list** *access-list-number* {**deny**	**permit**} *source* [*source-wildcard*] **Example:** Device(config)# **access-list 10 permit 10.10.0.0**	Creates a standard access list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 3. • The **deny** keyword denies access if the conditions are matched. • The **permit** keyword permits access if the conditions are matched. • For *source*, enter the multicast group address range for which the RP should be used. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. **Note** Recall that the access list is always terminated by an implicit deny statement for everything.
Step 6	**ip pim send-rp-discovery scope** *ttl* **Example:** Device(config)# **ip pim send-rp-discovery scope 50**	Finds a device whose connectivity is not likely to be interrupted, and assign it the role of RP-mapping agent. For **scope** *ttl*, specify the time-to-live value in hops to limit the RP discovery packets. All devices within the hop count from the source device receive the Auto-RP discovery messages. These messages tell other devices which group-to-RP mapping to use to avoid conflicts (such as overlapping group-to-RP ranges). There is no default setting. The range is 1 to 255.
Step 7	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 8	**show running-config** **Example:**	Verifies your entries.

	Command or Action	Purpose
	Device# **show running-config**	
Step 9	**show ip pim rp mapping** **Example:** Device# **show ip pim rp mapping**	Displays active RPs that are cached with associated multicast routing entries.
Step 10	**show ip pim rp** **Example:** Device# **show ip pim rp**	Displays the information cached in the routing table.
Step 11	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Adding Auto-RP to an Existing Sparse-Mode Cloud

This section contains suggestions for the initial deployment of Auto-RP into an existing sparse-mode cloud to minimize disruption of the existing multicast infrastructure.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **ip pim send-rp-announce** *interface-id* **scope** *ttl* **group-list** *access-list-number* **interval** *seconds*
5. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]
6. **ip pim send-rp-discovery scope** *ttl*
7. **end**
8. **show running-config**
9. **show ip pim rp mapping**
10. **show ip pim rp**
11. **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	**show running-config** **Example:** Device# **show running-config**	Verifies that a default RP is already configured on all PIM devices and the RP in the sparse-mode network. It was previously configured with the **ip pim rp-address** global configuration command. **Note** This step is not required for spare-dense-mode environments. The selected RP should have good connectivity and be available across the network. Use this RP for the global groups (for example, 224.x.x.x and other global groups). Do not reconfigure the group address range that this RP serves. RPs dynamically discovered through Auto-RP take precedence over statically configured RPs. Assume that it is desirable to use a second RP for the local groups.
Step 3	**configure terminal** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 4	**ip pim send-rp-announce** *interface-id* **scope** *ttl* **group-list** *access-list-number* **interval** *seconds* **Example:** Device(config)# **ip pim send-rp-announce gigabitethernet** **1/0/5 scope 20 group-list 10 interval 120**	Configures another PIM device to be the candidate RP for local groups. • For *interface-id*, enter the interface type and number that identifies the RP address. Valid interfaces include physical ports, port channels, and VLANs. • For **scope** *ttl*, specify the time-to-live value in hops. Enter a hop count that is high enough so that the RP-announce messages reach all mapping agents in the network. There is no default setting. The range is 1 to 255. • For **group-list** *access-list-number*, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups. • For **interval** *seconds*, specify how often the announcement messages must be sent. The default is 60 seconds. The range is 1 to 16383.
Step 5	**access-list** *access-list-number* {**deny**	**permit**} *source* [*source-wildcard*] **Example:** Device(config)# **access-list 10**	Creates a standard access list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 3.

	Command or Action	Purpose
	`permit  224.0.0.0 15.255.255.255`	• The **deny** keyword denies access if the conditions are matched.
		• The **permit** keyword permits access if the conditions are matched.
		• For *source*, enter the multicast group address range for which the RP should be used.
		• (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore.
		Recall that the access list is always terminated by an implicit deny statement for everything.
Step 6	**ip pim send-rp-discovery scope** *ttl*	Finds a device whose connectivity is not likely to be interrupted, and assigns it the role of RP-mapping agent.
	Example:	For **scope** *ttl*, specify the time-to-live value in hops to limit the RP discovery packets. All devices within the hop count from the source device receive the Auto-RP discovery messages. These messages tell other devices which group-to-RP mapping to use to avoid conflicts (such as overlapping group-to-RP ranges). There is no default setting. The range is 1 to 255.
	`Device(config)# ip pim send-rp-discovery scope 50`	
		Note To remove the device as the RP-mapping agent, use the **no ip pim send-rp-discovery** global configuration command.
Step 7	**end**	Returns to privileged EXEC mode.
	Example:	
	`Device(config)# end`	
Step 8	**show running-config**	Verifies your entries.
	Example:	
	`Device# show running-config`	
Step 9	**show ip pim rp mapping**	Displays active RPs that are cached with associated multicast routing entries.
	Example:	
	`Device#` `show ip pim rp mapping`	
Step 10	**show ip pim rp**	Displays the information cached in the routing table.

	Command or Action	Purpose
	Example: Device# **show ip pim rp**	
Step 11	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Preventing Join Messages to False RPs

Determine whether the **ip pim accept-rp** command was previously configured throughout the network by using the **show running-config** privileged EXEC command. If the **ip pim accept-rp** command is not configured on any device, this problem can be addressed later. In those routers or multilayer devices already configured with the **ip pim accept-rp** command, you must enter the command again to accept the newly advertised RP.

To accept all RPs advertised with Auto-RP and reject all other RPs by default, use the **ip pim accept-rp auto-rp** global configuration command.

This procedure is optional.

Filtering Incoming RP Announcement Messages

You can add configuration commands to the mapping agents to prevent a maliciously configured router from masquerading as a candidate RP and causing problems.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip pim rp-announce-filter rp-list** *access-list-number* **group-list** *access-list-number*
4. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]
5. **end**
6. **show running-config**
7. **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.

	Command or Action	**Purpose**
Step 2	**configure terminal** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip pim rp-announce-filter rp-list** *access-list-number* **group-list** *access-list-number* **Example:** Device(config)# **ip pim rp-announce-filter rp-list 10 group-list 14**	Filters incoming RP announcement messages. Enter this command on each mapping agent in the network. Without this command, all incoming RP-announce messages are accepted by default. For **rp-list** *access-list-number*, configure an access list of candidate RP addresses that, if permitted, is accepted for the group ranges supplied in the **group-list** *access-list-number* variable. If this variable is omitted, the filter applies to all multicast groups. If more than one mapping agent is used, the filters must be consistent across all mapping agents to ensure that no conflicts occur in the group-to-RP mapping information.
Step 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **Example:** Device(config)# **access-list 10 permit 10.8.1.0 255.255.224.0**	Creates a standard access list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access if the conditions are matched. • The **permit** keyword permits access if the conditions are matched. • Create an access list that specifies from which routers and multilayer devices the mapping agent accepts candidate RP announcements (rp-list ACL). • Create an access list that specifies the range of multicast groups from which to accept or deny (group-list ACL). • For *source*, enter the multicast group address range for which the RP should be used. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. The access list is always terminated by an implicit deny statement for everything.

	Command or Action	Purpose
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring PIMv2 BSR

The process for configuring PIMv2 BSR may involve the following optional tasks:

- Defining the PIM domain border

- Defining the IP multicast boundary

- Configuring candidate BSRs

- Configuring candidate RPs

Defining the PIM Domain Border

Perform the following steps to configure the PIM domain border. This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip pim bsr-border**
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

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	**configure terminal** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specifies the interface to be configured, and enters interface configuration mode. The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. These interfaces must have IP addresses assigned to them.
Step 4	**ip pim bsr-border** **Example:** Device(config-if)# **ip pim bsr-border**	Defines a PIM bootstrap message boundary for the PIM domain. Enter this command on each interface that connects to other bordering PIM domains. This command instructs the device to neither send nor receive PIMv2 BSR messages on this interface. **Note** To remove the PIM border, use the **no ip pim bsr-border** interface configuration command.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

Command or Action	Purpose
Device# **copy running-config startup-config**	

Defining the IP Multicast Boundary

You define a multicast boundary to prevent Auto-RP messages from entering the PIM domain. You create an access list to deny packets destined for 224.0.1.39 and 224.0.1.40, which carry Auto-RP information.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **deny** *source* [*source-wildcard*]
4. **interface** *interface-id*
5. **ip multicast boundary** *access-list-number*
6. **end**
7. **show running-config**
8. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **deny** *source* [*source-wildcard*] **Example:** Device(config)# **access-list 12 deny 224.0.1.39** **access-list 12 deny 224.0.1.40**	Creates a standard access list, repeating the command as many times as necessary. • For *access-list-number*, the range is 1 to 99. • The **deny** keyword denies access if the conditions are matched. • For *source*, enter multicast addresses 224.0.1.39 and 224.0.1.40, which carry Auto-RP information. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the

	Command or Action	Purpose
		source. Place ones in the bit positions that you want to ignore. The access list is always terminated by an implicit deny statement for everything.
Step 4	**interface** *interface-id* **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Specifies the interface to be configured, and enters interface configuration mode. The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. These interfaces must have IP addresses assigned to them.
Step 5	**ip multicast boundary** *access-list-number* **Example:** Device(config-if)# **ip multicast boundary 12**	Configures the boundary, specifying the access list you created in Step 2.
Step 6	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring Candidate BSRs

You can configure one or more candidate BSRs. The devices serving as candidate BSRs should have good connectivity to other devices and be in the backbone portion of the network.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip pim bsr-candidate** *interface-id hash-mask-length* [*priority*]
4. **end**
5. **show running-config**
6. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip pim bsr-candidate** *interface-id hash-mask-length* [*priority*] **Example:** Device(config)# **ip pim bsr-candidate gigabitethernet 1/0/3 28 100**	Configures your device to be a candidate BSR. • For *interface-id*, enter the interface on this device from which the BSR address is derived to make it a candidate. This interface must be enabled with PIM. Valid interfaces include physical ports, port channels, and VLANs. • For *hash-mask-length*, specify the mask length (32 bits maximum) that is to be ANDed with the group address before the 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. • (Optional) For *priority*, enter a number from 0 to 255. The BSR with the larger priority is preferred. If the priority values are the same, the device with the highest IP address is selected as the BSR. The default is 0.
Step 4	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring the Candidate RPs

You can configure one or more candidate RPs. Similar to BSRs, the RPs should also have good connectivity to other devices and be in the backbone portion of the network. An RP can serve the entire IP multicast address space or a portion of it. Candidate RPs send candidate RP advertisements to the BSR.

This procedure is optional.

Before you begin

When deciding which devices should be RPs, consider these options:

- In a network of Cisco routers and multilayer devices where only Auto-RP is used, any device can be configured as an RP.

- In a network that includes only Cisco PIMv2 routers and multilayer devices and with routers from other vendors, any device can be used as an RP.

- In a network of Cisco PIMv1 routers, Cisco PIMv2 routers, and routers from other vendors, configure only Cisco PIMv2 routers and multilayer devices as RPs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip pim rp-candidate** *interface-id* [**group-list** *access-list-number*]
4. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]
5. **end**
6. **show running-config**
7. **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 pim rp-candidate** *interface-id* [**group-list** *access-list-number*] **Example:** Device(config)# **ip pim rp-candidate gigabitethernet 1/0/5 group-list 10**	Configures your device to be a candidate RP. • For *interface-id*, specify the interface whose associated IP address is advertised as a candidate RP address. Valid interfaces include physical ports, port channels, and VLANs. • (Optional) For **group-list** *access-list-number*, enter an IP standard access list number from 1 to 99. If no group-list is specified, the device is a candidate RP for all groups.
Step 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **Example:** Device(config)# **access-list 10 permit 239.0.0.0 0.255.255.255**	Creates a standard access list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access if the conditions are matched. The **permit** keyword permits access if the conditions are matched. • For *source*, enter the number of the network or host from which the packet is being sent. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. The access list is always terminated by an implicit deny statement for everything.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:**	Verifies your entries.

	Command or Action	Purpose
	Device# **show running-config**	
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring Sparse Mode with Auto-RP

Before you begin

- 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).

- 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. **interface** *type number*
6. **ip pim sparse-mode**
7. **exit**
8. Repeat Steps 1 through 9 on all PIM interfaces.
9. **ip pim send-rp-announce** {*interface-type interface-number | ip-address*} **scope** *ttl-value* [**group-list** *access-list*] [**interval** *seconds*] [**bidir**]
10. **ip pim send-rp-discovery** [*interface-type interface-number*] **scope** *ttl-value* [**interval** *seconds*]
11. **ip pim rp-announce-filter rp-list** *access-list* **group-list** *access-list*
12. **interface** *type number*

13. **ip multicast boundary** *access-list* [**filter-autorp**]
14. **end**
15. **show ip pim autorp**
16. **show ip pim rp** [**mapping**] [*rp-address*]
17. **show ip igmp groups** [*group-name | group-address| interface-type interface-number*] [**detail**]
18. **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	**interface** *type number* **Example:**	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 6	**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 7	**exit** **Example:**	Exits interface configuration mode and returns to global configuration mode.
Step 8	Repeat Steps 1 through 9 on all PIM interfaces.	--
Step 9	**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.

	Command or Action	Purpose
		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 10	**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. • 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.

	Command or Action	Purpose
Step 11	**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 12	**interface** *type number*	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 13	**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 14	**end**	Returns to global configuration mode.
Step 15	**show ip pim autorp**	(Optional) Displays the Auto-RP information.
Step 16	**show ip pim rp** [**mapping**] [*rp-address*]	(Optional) Displays RPs known in the network and shows how the device learned about each RP.
Step 17	**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 18	**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.

Delaying the Use of PIM Shortest-Path Tree

Perform these steps to configure a traffic rate threshold that must be reached before multicast routing is switched from the source tree to the shortest-path tree.

This procedure is optional.

SUMMARY STEPS

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

3. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]
4. **ip pim spt-threshold** {*kbps* | **infinity**} [**group-list** *access-list-number*]
5. **end**
6. **show running-config**
7. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.

Step 3	**access-list** *access-list-number* {**deny**	**permit**} *source* [*source-wildcard*]
		
		Example:
		
		Device(config)# **access-list 16 permit 225.0.0.0 0.255.255.255**
			Creates a standard access list.
			
			   • For *access-list-number*, the range is 1 to 99.
			
			   • The **deny** keyword denies access if the conditions are matched.
			
			   • The **permit** keyword permits access if the conditions are matched.
			
			   • For *source*, specify the multicast group to which the threshold will apply.
			
			   • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore.
			
			The access list is always terminated by an implicit deny statement for everything.

Step 4	**ip pim spt-threshold** {*kbps*	**infinity**} [**group-list** *access-list-number*]
		
		Example:
		
		Device(config)# **ip pim spt-threshold infinity group-list 16**
			Specifies the threshold that must be reached before moving to shortest-path tree (spt).
			
			   • For *kbps*, specify the traffic rate in kilobits per second. The default is 0 kbps.
			
			   Note Because of device hardware limitations, 0 kbps is the only valid entry even though the range is 0 to 4294967.

	Command or Action	Purpose
		• Specify **infinity** if you want all sources for the specified group to use the shared tree, never switching to the source tree.
		• (Optional) For **group-list** *access-list-number*, specify the access list created in Step 2. If the value is 0 or if the group list is not used, the threshold applies to all groups.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Modifying the PIM Router-Query Message Interval

PIM routers and multilayer devices send PIM router-query messages to find which device will be the designated router (DR) for each LAN segment (subnet). The DR is responsible for sending IGMP host-query messages to all hosts on the directly connected LAN.

With PIM DM operation, the DR has meaning only if IGMPv1 is in use. IGMPv1 does not have an IGMP querier election process, so the elected DR functions as the IGMP querier. With PIM-SM operation, the DR is the device that is directly connected to the multicast source. It sends PIM register messages to notify the RP that multicast traffic from a source needs to be forwarded down the shared tree. In this case, the DR is the device with the highest IP address.

This procedure is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip pim query-interval** *seconds*
5. **end**
6. **show ip igmp interface** [*interface-id*]

7. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** Device(config)# **interface** **gigabitethernet 1/0/1**	Specifies the interface to be configured, and enters interface configuration mode. The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. These interfaces must have IP addresses assigned to them.
Step 4	**ip pim query-interval** *seconds* **Example:** Device(config-if)# **ip pim** **query-interval 45**	Configures the frequency at which the device sends PIM router-query messages. The default is 30 seconds. The range is 1 to 65535.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show ip igmp interface** [*interface-id*] **Example:** Device# **show ip igmp interface**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

Command or Action	Purpose
Device# **copy running-config startup-config**	

Verifying PIM Operations

Verifying IP Multicast Operation in a PIM-SM or a PIM-SSM Network

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.

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 on the First Hop Router

Enter these commands on the first hop router to verify IP multicast operations on the first hop router:

SUMMARY STEPS

1. **enable**
2. **show ip mroute** [*group-address*]
3. **show ip mroute active** [*kb/s*]

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 mroute** [*group-address*] **Example:** Device# **show ip mroute 239.1.2.3** (*, 239.1.2.3), 00:18:10/stopped, RP 172.16.0.1,	Confirms that the F flag has been set for mroutes on the first hop router.

	Command or Action	Purpose
	``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, 00:18:10/00:03:19``` | |
| Step 3 | **show ip mroute active** [*kb/s*]

Example:

```Device# 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)``` | Displays 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. |

Verifying IP Multicast on Routers Along the SPT

Enter these commands on routers along the SPT to verify IP multicast operations 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

	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	**show ip mroute** [*group-address*]
	
	Example:
	
	``Device# 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,
	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,
	00:15:34/00:03:02```
		Confirms the RPF neighbor towards the source for a particular group or groups.

Step 3	**show ip mroute active**
	
	Example:
	
	``Device# 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)```
		Displays 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.

Verifying IP Multicast Operation on the Last Hop Router

Enter these commands on the last hop router to verify IP multicast operations on the last hop router:

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

	Command or Action	**Purpose**
Step 1	**enable** **Example:** Device> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip igmp groups** **Example:** Device# **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	Verifies 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.
Step 3	**show ip pim rp mapping** **Example:** Device# **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	Confirms 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 do not appear in the output of the **show ip pim rp mapping**command.
Step 4	**show ip mroute** **Example:** Device# **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, 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, 00:02:49/00:03:04 (*, 224.0.1.39), 00:10:05/stopped, RP 0.0.0.0, flags: DC	Verifies that the mroute table is being populated properly on the last hop router.

Command or Action	Purpose

``
    Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse,
00:05:15/00:00:00
    GigabitEthernet0/0, Forward/Sparse,
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*]
 	
Example:
 	
``
Device# 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
``
	Verifies 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.

Step 6 **show ip mfib** **Example:** ``` Device# show ip mfib ```	Displays the forwarding entries and interfaces in the IP Multicast Forwarding Information Base (MFIB).

	Command or Action	Purpose
Step 7	**show ip pim interface count** **Example:** ``` Device# 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 ```	Confirms that multicast traffic is being forwarded on the last hop router.

Step 8	**show ip mroute count**
	
	Example:
	
	``
	Device# 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
	``
		Confirms that multicast traffic is being forwarded on the last hop router.

Step 9	**show ip mroute active** [*kb/s*] **Example:** ``` Device# show ip mroute active Active IP Multicast Sources - sending >= 4 kbps ```	Displays 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.

Command or Action	Purpose
| ```
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)
``` | **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. |

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

Follow these steps to configure a router to respond to multicast pings. Perform the task on all the interfaces of a router and on all the routers participating in the multicast network:

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.

	Command or Action	Purpose
Step 4	**ip igmp join-group** *group-address*	(Optional) Configures an interface on the router to join the specified group.
		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

Follow these steps 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.

Monitoring and Troubleshooting PIM

Monitoring PIM Information

Use the privileged EXEC commands in the following table to monitor your PIM configurations.

Table 18: PIM Monitoring Commands

Command	Purpose
show ip pim all-vrfs tunnel [**tunnel** *tunnel_number*	**verbose**]	Displays all VRFs.
show ip pim autorp	Displays global auto-RP information.
show ip pim boundary	Displays information about mroutes filtered by administratively scoped IPv4 multicast boundaries configured on an interface.
show ip pim interface	Displays information about interfaces configured for Protocol Independent Multicast (PIM).
show ip pim neighbor	Displays the PIM neighbor information.
show ip pim rp[*group-name*	*group-address*]	Displays RP routers associated with a sparse-mode multicast group. This command is available in all software images.
show ip pim tunnel [**tunnel**	**verbose**]	Displays information about Protocol Independent Multicast (PIM) tunnel interfaces
show ip pim vrf { **word** { **all-vrfs**	**autorp**	**boundary**	**bsr-router**	**interface**	**mdt**	**neighbor**	**rp**	**rp-hash**	**tunnel** } }	Displays the VPN routing/forwarding instance.
show ip igmp groups detail	Displays the interested clients that have joined the specific multicast source group.

Monitoring the RP Mapping and BSR Information

Use the privileged EXEC mode in the following table to verify the consistency of group-to-RP mappings:

Table 19: RP Mapping Monitoring Commands

Command	Purpose
show ip pim rp [*hostname* or *IP address* | **mapping** [*hostname* or *IP address* | **elected** | **in-use**] | **metric** [*hostname* or *IP address*]]	Displays all available RP mappings and metrics. This tells you how the device learns of the RP (through the BSR or the Auto-RP mechanism). • (Optional) For the *hostname*, specify the IP name of the group about which to display RPs. • (Optional) For the *IP address*, specify the IP address of the group about which to display RPs. • (Optional) Use the **mapping** keyword to display all group-to-RP mappings of which the Cisco device is aware (either configured or learned from Auto-RP). • (Optional) Use the **metric** keyword to display the RP RPF metric.
show ip pim rp-hash *group*	Displays the RP that was selected for the specified group. That is, on a PIMv2 router or multilayer device, confirms that the same RP is the one that a PIMv1 system chooses. For *group*, enter the group address for which to display RP information.

Use the privileged EXEC commands in the following table to monitor BSR information:

Table 20: BSR Monitoring Commands

Command	Purpose
show ip pim bsr	Displays information about the elected BSR.
show ip pim bsr-router	Displays information about the BSRv2.

Troubleshooting PIMv1 and PIMv2 Interoperability Problems

When debugging interoperability problems between PIMv1 and PIMv2, check these in the order shown:

1. Verify RP mapping with the **show ip pim rp-hash** privileged EXEC command, making sure that all systems agree on the same RP for the same group.

2. Verify interoperability between different versions of DRs and RPs. Make sure that the RPs are interacting with the DRs properly (by responding with register-stops and forwarding decapsulated data packets from registers).

Configuration Examples for PIM

Example: Enabling PIM Stub Routing

In this example, IP multicast routing is enabled, Switch A PIM uplink port 25 is configured as a routed uplink port with **spare-dense-mode** enabled. PIM stub routing is enabled on the VLAN 100 interfaces and on Gigabit Ethernet port 20.

```
Device(config)# ip multicast-routing distributed
Device(config)# interface GigabitEthernet3/0/25
Device(config-if)# no switchport
Device(config-if)# ip address 3.1.1.2 255.255.255.0
Device(config-if)# ip pim sparse-dense-mode
Device(config-if)# exit
Device(config)# interface vlan100
Device(config-if)# ip pim passive
Device(config-if)# exit
Device(config)# interface GigabitEthernet3/0/20
Device(config-if)# ip pim passive
Device(config-if)# exit
Device(config)# interface vlan100
Device(config-if)# ip address 100.1.1.1 255.255.255.0
Device(config-if)# ip pim passive
Device(config-if)# exit
Device(config)# interface GigabitEthernet3/0/20
Device(config-if)# no switchport
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# ip pim passive
Device(config-if)# end
```

Example: Verifying PIM Stub Routing

To verify that PIM stub is enabled for each interface, use the **show ip pim interface** privileged EXEC command:

```
Device# show ip pim interface
Address Interface Ver/ Nbr Query DR DR
Mode Count Intvl Prior
3.1.1.2 GigabitEthernet3/0/25 v2/SD 1 30 1 3.1.1.2
100.1.1.1 Vlan100 v2/P 0 30 1 100.1.1.1
10.1.1.1 GigabitEthernet3/0/20 v2/P 0 30 1 10.1.1.1
```

Example: Manually Assigning an RP to Multicast Groups

This example shows how to configure the address of the RP to 147.106.6.22 for multicast group 225.2.2.2 only:

```
Device(config)# access-list 1 permit 225.2.2.2 0.0.0.0
Device(config)# ip pim rp-address 147.106.6.22 1
```

Example: Configuring Auto-RP

This example shows how to send RP announcements out all PIM-enabled interfaces for a maximum of 31 hops. The IP address of port 1 is the RP. Access list 5 describes the group for which this device serves as RP:

```
Device(config)# ip pim send-rp-announce gigabitethernet1/0/1 scope 31 group-list 5
Device(config)# access-list 5 permit 224.0.0.0 15.255.255.255
```

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
```

Example: Defining the IP Multicast Boundary to Deny Auto-RP Information

This example shows a portion of an IP multicast boundary configuration that denies Auto-RP information:

```
Device(config)# access-list 1 deny 224.0.1.39
Device(config)# access-list 1 deny 224.0.1.40
Device(config)# access-list 1 permit all
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip multicast boundary 1
```

Example: Filtering Incoming RP Announcement Messages

This example shows a sample configuration on an Auto-RP mapping agent that is used to prevent candidate RP announcements from being accepted from unauthorized candidate RPs:

```
Device(config)# ip pim rp-announce-filter rp-list 10 group-list 20
Device(config)# access-list 10 permit host 172.16.5.1
Device(config)# access-list 10 permit host 172.16.2.1
Device(config)# access-list 20 deny 239.0.0.0 0.0.255.255
Device(config)# access-list 20 permit 224.0.0.0 15.255.255.255
```

The mapping agent accepts candidate RP announcements from only two devices, 172.16.5.1 and 172.16.2.1. The mapping agent accepts candidate RP announcements from these two devices only for multicast groups that fall in the group range of 224.0.0.0 to 239.255.255.255. The mapping agent does not accept candidate

RP announcements from any other devices in the network. Furthermore, the mapping agent does not accept candidate RP announcements from 172.16.5.1 or 172.16.2.1 if the announcements are for any groups in the 239.0.0.0 through 239.255.255.255 range. This range is the administratively scoped address range.

Example: Preventing Join Messages to False RPs

If all interfaces are in sparse mode, use a default-configured RP to support the two well-known groups 224.0.1.39 and 224.0.1.40. Auto-RP uses these two well-known groups to collect and distribute RP-mapping information. When this is the case and the **ip pim accept-rp auto-rp** command is configured, another **ip pim accept-rp** command accepting the RP must be configured as follows:

```
Device(config)# ip pim accept-rp 172.10.20.1 1
Device(config)# access-list 1 permit 224.0.1.39
Device(config)# access-list 1 permit 224.0.1.40
```

Example: Configuring Candidate BSRs

This example shows how to configure a candidate BSR, which uses the IP address 172.21.24.18 on a port as the advertised BSR address, uses 30 bits as the hash-mask-length, and has a priority of 10.

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip address 172.21.24.18 255.255.255.0
Device(config-if)# ip pim sparse-mode
Device(config-if)# ip pim bsr-candidate gigabitethernet1/0/2 30 10
```

Example: Configuring Candidate RPs

This example shows how to configure the device to advertise itself as a candidate RP to the BSR in its PIM domain. Standard access list number 4 specifies the group prefix associated with the RP that has the address identified by a port. That RP is responsible for the groups with the prefix 239.

```
Device(config)# ip pim rp-candidate gigabitethernet1/0/2 group-list 4
Device(config)# access-list 4 permit 239.0.0.0 0.255.255.255
```

Additional References for PIM

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	

Standards and RFCs

Standard/RFC	Title
PIM is defined in RFC 4601 and in these Internet Engineering Task Force (IETF) Internet drafts.	• *Protocol Independent Multicast (PIM): Motivation and Architecture* • *Protocol Independent Multicast (PIM), Sparse Mode Protocol Specification* • *draft-ietf-idmr-igmp-v2-06.txt, Internet Group Management Protocol, Version 2*

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 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/support

Feature History and Information for PIM

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.

Configuring PIM MIB Extension for IP Multicast

Information About PIM MIB Extension for IP Multicast

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 Configure PIM MIB Extension for 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**]	Enables a device to send PIM notifications.

	Command or Action	Purpose

	Example:
	
	`Device(config)# snmp-server enable traps pim`
	`neighbor-change`
		• **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 PIM MIB Extensions

Example Enabling PIM MIB Extensions for IP Multicast

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-mode
```

Additional References for PIM MIB Extension for IP Multicast

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	

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 History and Information for PIM MIB Extension

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.

CHAPTER **11**

Configuring MSDP

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.

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 device 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 devices 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.

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.

Note

Figure 15: MSDP Running Between RP Peers

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

1. When a PIM designated device (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.

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 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 device 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.

Note For more information about SA messages, see the SA Message Origination Receipt and Processing, on page 222 section.

SA Request Messages

SA request messages are used to request a list of active sources for a specific group. These messages are sent to an MSDP SA cache that maintains a list of active (S, G) pairs in its SA cache. Join latency can be reduced by using SA request messages to request the list of active sources for a group instead of having to wait up to 60 seconds for all active sources in the group to be readvertised by originating RPs.

SA Response Messages

SA response messages are sent by the MSDP peer in response to an SA request message. SA response messages contain the IP address of the originating RP and one or more (S, G) pairs of the active sources in the originating RP's domain that are stored in the cache.

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.

Default MSDP Peers

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 Device B is connected to the Internet through two Internet service providers (ISPs), one that owns Device A and the other that owns Device 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, Device B identifies Device A as its default MSDP peer. Device B advertises SA messages to both Device A and Device C, but accepts SA messages either from Device A only or Device C only. If Device A is the first default peer in the configuration, it will be used if it is up and running. Only if Device A is not running will Device B accept SA messages from Device C.

The ISP will also likely use a prefix list to define which prefixes it will accept from the customer device. 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

Device B advertises SAs to Device A and Device C, but uses only Device A or Device C to accept SA messages. If Device A is first in the configuration, it will be used if it is up and running. Only when Device A is not running will Device B accept SAs from Device 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 device 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 request messages from a specified peer based on match criteria defined in a route map by configuring the device to only receive SA messages that 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 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.

SA Request Messages

You can configure a noncaching device to send SA request messages to one or more specified MSDP peers.

If an noncaching RP has an MSDP peer that is caching SAs, you can reduce the join latency for a noncaching peer by enabling the noncaching peer to send SA request messages. When a host requests a join to a particular group, the noncaching RP sends an SA request message to its caching peers. If a peer has cached source information for the group in question, it sends the information to the requesting RP with an SA response message. The requesting RP uses the information in the SA response but does not forward the message to any other peers. If a noncaching RP receives an SA request, it sends an error message back to the requestor.

✎

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 configured commands are automatically added to the running configuration.

SA Request Filters

By default, a device honors all outgoing SA request messages from its MSDP peers; that is, it sends cached source information to requesting MSDP peers in SA response messages. You can control the outgoing SA request messages that a device will honor from specified peers by creating an SA request filter. An SA request filter controls the outgoing SA requests that the device will honor from MSDP peers as follows:

• You can filter all SA request messages from a specified peer by configuring the device to ignore all SA requests from the specified MSDP peer.

• You can filter a subset of SA request messages from a specified peer based on groups defined in a standard access list by configuring the device to honor only SA request messages from the MSDP peer that match the groups defined in a standard access list. SA request messages from the specified peer for other groups will be ignored.

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:**	Enables MSDP and configures an MSDP peer as specified by the DNS name or IP address.

	Command or Action	Purpose
	`Device(config)# ip msdp peer 192.168.1.2 connect-source loopback0`	**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 239 section or the Configuring an MSDP Mesh Group, on page 240 section. • 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** **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 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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	

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 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

	Command or Action	**Purpose**
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.

	Command or Action	Purpose
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.

	Command or Action	**Purpose**
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.

	Command or Action	Purpose
Step 4	Repeat Step 3 to configure outgoing filter lists for additional MSDP peers.	--
Step 5	**exit**	Exits global configuration mode and returns to privileged EXEC mode.
	Example:	
	`Device(config)# exit`	

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**	Enables privileged EXEC mode.
	Example:	• Enter your password if prompted.
	`Device> enable`	
Step 2	**configure terminal**	Enters global configuration mode.
	Example:	
	`Device# configure terminal`	
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*]	Enables a filter for incoming MSDP SA messages.
	Example:	

	Command or Action	Purpose
	Device(config)# ip msdp sa-filter in 192.168.1.3	
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:**	Exits global configuration mode and returns to privileged EXEC mode.

Command or Action	Purpose
Device(config)# exit	

Requesting Source Information from MSDP Peers

Perform this optional task to enable a device to request source information from MSDP peers.

✎

Note   Because SA caching is enabled by default and cannot be explicitly enabled or disabled in earlier Cisco software releases, performing this task is seldom needed.

SUMMARY STEPS

1. **enable**
2. **configure   terminal**
3. **ip msdp sa-request** {*peer-address* | *peer-name*}
4. Repeat Step 3 to specify that the device send SA request messages to additional MSDP caching 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-request** {*peer-address*	*peer-name*} **Example:** Device(config)# ip msdp sa-request 192.168.10.1	Specifies that the device send SA request messages to the specified MSDP peer.
Step 4	Repeat Step 3 to specify that the device send SA request messages to additional MSDP caching peers.	--
Step 5	**exit** **Example:** Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Controlling the Response to Outgoing SA Request Messages from MSDP Peers Using SA Request Filters

Perform this optional task to control the outgoing SA request messages that the device will honor from MSDP peers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp filter-sa-request** {*peer-address* | *peer-name*} [**list** *access-list*]
4. Repeat Step 3 to configure SA request filters 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 filter-sa-request** {*peer-address*	*peer-name*} [**list** *access-list*] **Example:** `Device(config)# ip msdp filter sa-request 172.31.2.2 list 1`	Enables a filter for outgoing SA request messages. **Note** Only one SA request filter can be configured per MSDP peer.
Step 4	Repeat Step 3 to configure SA request filters for additional MSDP peers.	--
Step 5	**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 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.

	Command or Action	Purpose
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. • 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.

	Command or Action	Purpose
Step 3	**snmp-server host** *host* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth**| **priv** | **noauth**]}] *community-string* [**udp-port** *port-number*] **msdp**	Specifies the recipient (host) for MSDP traps or informs.
	Example:	
	`Device# snmp-server host examplehost msdp`	
Step 4	**exit**	Exits global configuration mode and returns to privileged EXEC mode.
	Example:	
	`Device(config)# exit`	

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
!
```

Example: Configuring a Default MSDP Peer

The figure illustrates a scenario where default MSDP peers might be used. In the figure, a customer that owns Device B is connected to the internet through two ISPs, one that owns Device A and the other that owns Device 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, Device B identifies Device A as its default MSDP peer. Device B advertises SA messages to both Device A and Device C, but accepts SA messages either from Device A only or Device C only. If Device A is the first default peer in the configuration, it will be used if it is up and running. Only if Device A is not running will Device B accept SA messages from Device C.

The ISP will also likely use a prefix list to define which prefixes it will accept from the customer device. 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.

Note

Figure 17: Default MSDP Peer Scenario

Device B advertises SAs to Device A and Device C, but uses only Device A or Device C to accept SA messages. If Device A is first in the configuration file, it will be used if it is up and running. Only when Device A is not running will Device B accept SAs from Device 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 device 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 Device A and Device 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.

Device A Configuration

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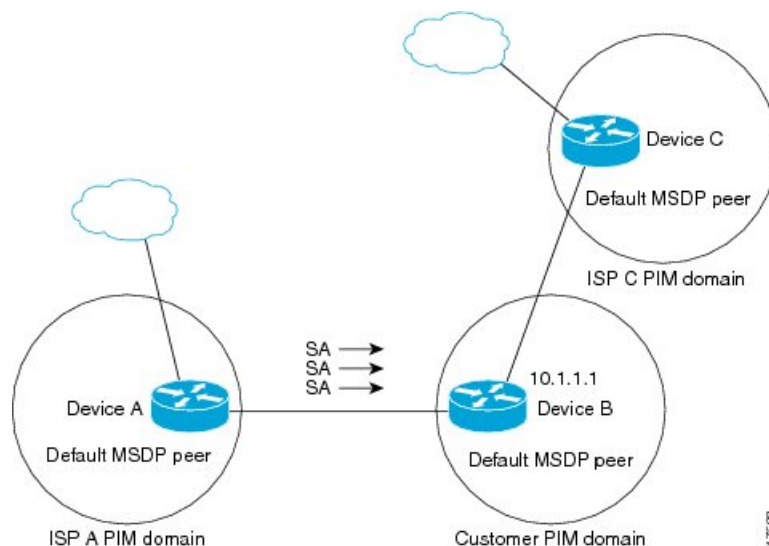ip msdp default-peer 10.1.1.1
ip msdp default-peer 10.1.1.1 prefix-list site-b ge 32
ip prefix-list site-b permit 10.0.0.0/8
```

Device 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
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv6 commands	*IPv6 Command References*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

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 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 History and Information for Multicast Source Discovery Protocol

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.

Configuring SSM

Prerequisites for Configuring SSM

The following are the prerequisites for configuring source-specific multicast (SSM) and SSM mapping:

- Before you configure SSM mapping, you must perform the following tasks:
    - Enable IP multicast routing.
    - Enable PIM sparse mode.
    - Configure SSM.

- Before you configure static SSM mapping, you must configure access control lists (ACLs) that define the group ranges to be mapped to source addresses.

- 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.

Note You can use a product such as *Cisco Network Registrar* to add records to a running DNS server.

Restrictions for Configuring SSM

The following are the restrictions for configuring SSM:

- To run SSM with IGMPv3, SSM must be supported in the Cisco IOS router, the host where the application is running, and the application itself.

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

- IGMP Snooping—IGMPv3 uses new membership report messages that might not be correctly recognized by older IGMP snooping devices.

- 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) 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, they do not benefit from these existing mechanisms. Instead, both receivers receive all (S, G) channel traffic and filter out the unwanted traffic on input. Because SSM can re-use the group addresses in the SSM range for many independent applications, this situation can lead to decreased traffic filtering in a switched network. For this reason, it is important to use random IP addresses from the SSM range for an application to minimize the chance for re-use 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 guarantees that multiple receivers to different channels within the same application service never experience traffic aliasing in networks that include Layer 2 devices.

- 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).

  The opposite situation occurs with 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 is deleted and only reestablished after packets from the source arrive again through the RPT (rendezvous point tree). 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.

The following are the restrictions for configuring 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. When you enable both SSM mapping and IGMPv3 and the hosts already support IGMPv3 (but not SSM), the hosts send IGMPv3 group reports. SSM mapping does not support these IGMPv3 group reports, and the router does not correctly associate sources with these reports.

Information About SSM

The source-specific multicast (SSM) feature is an extension of IP multicast in which datagram traffic is forwarded to receivers from only those multicast sources that the receivers have explicitly joined. For multicast groups configured for SSM, only SSM distribution trees (no shared trees) are created.

This section describes how to configure source-specific multicast (SSM). For a complete description of the SSM commands in this section, refer to the *IP Multicast Command Reference*.

SSM Components Overview

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. The device supports the following components that support SSM implementation:

- Protocol independent multicast source-specific mode (PIM-SSM)

  PIM-SSM is the routing protocol that supports the implementation of SSM and is derived from PIM sparse mode (PIM-SM).

- Internet Group Management Protocol version 3 (IGMPv3)

SSM and Internet Standard Multicast (ISM)

The current 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 the 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.

The ISM 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 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. 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 uses IGMP and includes modes membership reports, which are supported only in IGMP version 3.

SSM IP Address Range

SSM can coexist with the ISM service by applying the SSM delivery model to a configured subset of the IP multicast group address range. Cisco IOS software allows SSM configuration for the IP multicast address range of 224.0.0.0 through 239.255.255.255. When an SSM range is defined, existing IP multicast receiver applications do not receive any traffic when they try to use an address in the SSM range (unless the application is modified to use an explicit (S, G) channel subscription).

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 required for interdomain PIM-SM (for example, MSDP, Auto-RP, or bootstrap router [BSR]) if only SSM service is needed.

If SSM is deployed in a network already configured for PIM-SM, only the last-hop routers support SSM. Routers that are not directly connected to receivers do not require support for SSM. In general, these not-last-hop routers must only run PIM-SM in the SSM range and might need additional access control configuration to suppress MSDP signalling, registering, or PIM-SM shared tree operations from occurring within the SSM range.

Use the **ip pim ssm** global configuration command to configure the SSM range and to enable SSM. 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, and no (S, G) rendezvous point tree (RPT) or (*, G) RPT messages are generated. Incoming messages related to 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).

- No MSDP source-active (SA) messages within the SSM range are accepted, generated, or forwarded.

SSM Mapping

In a typical set-top box (STB) deployment, each TV channel uses one separate IP multicast group and has one active server host sending the TV channel. A single server can 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, the report addresses the well-known TV server for the TV channel associated with the multicast group.

When SSM mapping is configured, if a router receives an IGMPv1 or IGMPv2 membership report for a particular group, the router translates this report into one or more channel memberships for the well-known sources associated with this group.

When the router receives an IGMPv1 or IGMPv2 membership report for a group, the router uses SSM mapping to determine one or more source IP addresses for the group. SSM mapping then translates the membership report as an IGMPv3 report and continues as if it had received an IGMPv3 report. The router then sends PIM joins and continues to be joined to these groups as long as it continues to receive the IGMPv1 or IGMPv2 membership reports, and the SSM mapping for the group remains the same.

SSM mapping enables the last hop router to determine the source addresses either by a statically configured table on the router or through a DNS server. When the statically configured table or the DNS mapping changes, the router leaves the current sources associated with the joined groups.

Static SSM Mapping

With static SSM mapping, you can configure the last hop router to use a static map to determine the sources that are sending to groups. Static SSM mapping requires that you configure ACLs to define group ranges.

After configuring the ACLs to define group ranges, you can then map the groups permitted by those ACLs to sources by using the **ip igmp ssm-map static** global configuration command.

You can configure static SSM mapping in smaller networks when a DNS is not needed or to locally override DNS mappings. When configured, static SSM mappings take precedence over DNS mappings.

DNS-Based SSM Mapping

You can use DNS-based SSM mapping to configure the last hop router to perform a reverse DNS lookup to determine sources sending to groups. When DNS-based SSM mapping is configured, the router constructs a domain name that includes the group address and performs a reverse lookup into the DNS. The router looks up IP address resource records and uses them 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.

The SSM mapping mechanism that enables the last hop router to join multiple sources for a group can provide source redundancy for a TV broadcast. In this context, the last hop router provides redundancy using SSM mapping to simultaneously join two video sources for the same TV channel. However, to prevent the last hop router from duplicating the video traffic, the video sources must use a server-side switchover mechanism. One video source is active, and 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. Thus, the server-side switchover mechanism ensures that only one of the servers is actively sending video traffic for the TV channel.

To look up one or more source addresses for a group that includes G1, G2, G3, and G4, you must configure these DNS records 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
```

See your DNS server documentation for more information about configuring DNS resource records.

How to Configure SSM

For a complete description of the source-specific multicast (SSM) commands in this section, see the *IP Multicast Command Reference, Cisco IOS XE Release 3SE (Catalyst 3850 Switches)*.

Configuring SSM

Follow these steps to configure SSM:

This procedure is optional.

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. **enable**
2. **configure terminal**
3. **ip pim ssm** [**default** | **range** *access-list*]

4. **interface** *type number*
5. **ip pim** {**sparse-mode** | }
6. **ip igmp version 3**
7. **end**
8. **show running-config**
9. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**ip pim ssm** [**default**	**range** *access-list*] **Example:** Device(config)# **ip pim ssm range 20**	Defines the SSM range of IP multicast addresses.
Step 4	**interface** *type number* **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Selects an interface that is connected to hosts on which IGMPv3 can be enabled, and enters the interface configuration mode. The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. These interfaces must have IP addresses assigned to them.
Step 5	**ip pim** {**sparse-mode**	} **Example:** Device(config-if)# **ip pim sparse-mode**	Enables PIM on an interface.
Step 6	**ip igmp version 3** **Example:**	Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2.

	Command or Action	Purpose
	Device(config-if)# **ip igmp version 3**	
Step 7	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 8	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring Source Specific Multicast Mapping

The Source Specific Multicast (SSM) mapping feature supports SSM transition when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. You can use SSM mapping to leverage SSM for video delivery to legacy STBs that do not support IGMPv3 or for applications that do not use the IGMPv3 host stack.

Configuring Static SSM Mapping

Follow these steps to configure static SSM Mapping:

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. **end**
7. **show running-config**
8. **copy running-config startup-config**

DETAILED STEPS

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

	Command or Action	**Purpose**
	Example:	• Enter your password if prompted.
	Device> **enable**	
Step 2	**configure terminal**	Enters global configuration mode.
	Example:	
	Device# **configure terminal**	
Step 3	**ip igmp ssm-map enable**	Enables SSM mapping for groups in the configured SSM range.
	Example:	**Note** By default, this command enables DNS-based SSM mapping.
	Device(config)# ip igmp ssm-map enable	
Step 4	**no ip igmp ssm-map query dns**	(Optional) Disables DNS-based SSM mapping.
	Example:	**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.
	Device(config)# no ip igmp ssm-map query dns	
Step 5	**ip igmp ssm-map static** *access-list source-address*	Configures static SSM mapping.
	Example:	• 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.
	Device(config)# ip igmp ssm-map static 11 172.16.8.11	**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 device determines the source addresses associated with the group by walking each configured **ip igmp ssm-map static** command. The device associates up to 20 sources per group.
		Repeat Step to configure additional static SSM mappings, if required.
Step 6	**end**	Returns to privileged EXEC mode.
	Example:	
	Device(config)# **end**	
Step 7	**show running-config**	Verifies your entries.
	Example:	

	Command or Action	Purpose
	Device# **show running-config**	
Step 8	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring DNS-Based SSM Mapping

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, 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 false DNS setup with an empty root zone or a root zone that points back to itself.

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. **ip name-server** *server-address1* [*server-address2...server-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:** 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 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 for DNS-based SSM mapping. • By default, the software uses the ip-addr.arpa domain prefix.
Step 6	**ip name-server** *server-address1* [*server-address2...server-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)# **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.

Configuring Static Traffic Forwarding with SSM Mapping

Follow these steps to configure static traffic forwarding with SSM mapping on the last hop router:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip igmp static-group** *group-address* **source ssm-map**
5. **end**
6. **show running-config**
7. **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** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** Device(config)# **interface gigabitethernet 1/0/1**	Selects an interface on which to statically forward traffic for a multicast group using SSM mapping, and enters interface configuration mode. The specified interface must be one of the following: • A routed port—A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI—A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. These interfaces must have IP addresses assigned to them. **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:**	Configures SSM mapping to statically forward a (S, G) channel from the interface.

	Command or Action	Purpose
	Device(config-if)# **ip igmp static-group 239.1.2.1 source ssm-map**	Use this command if you want to statically forward SSM traffic for certain groups. Use DNS-based SSM mapping to determine the source addresses of the channels.
Step 5	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** Device# **show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** Device# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Monitoring SSM

Use the privileged EXEC commands in the following table to monitor SSM.

Table 21: Commands for Monitoring SSM

Command	Purpose
show ip igmp groups detail	Displays the (S, G) channel subscription through IGMPv3.
show ip mroute	Displays whether a multicast group supports SSM service or whether a source-specific host report was received.

Monitoring SSM Mapping

Use the privileged EXEC commands in the following table to monitor SSM mapping.

Table 22: SSM Mapping Monitoring Commands

Command	Purpose
Device# **show ip igmp ssm-mapping**	Displays information about SSM mapping.

Command	Purpose
Device#**show ip igmp ssm-mapping** *group-address*	Displays the sources that SSM mapping uses for a particular group.
Device#**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 router and that were learned through IGMP.
Device#**show host**	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.
Device#**debug ip igmp** *group-address*	Displays the IGMP packets received and sent and IGMP host-related events.

Where to Go Next for SSM

You can configure the following:

- IGMP

- Wireless Multicast

- PIM

- IP Multicast Routing

- Service Discovery Gateway

Feature History and Information for SSM

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.

CHAPTER 13

Configuring the Service Discovery Gateway

Restrictions for Configuring the Service Discovery Gateway

The following are restrictions for configuring the Service Discovery Gateway:

- The Service Discovery Gateway does not support topologies with multiple hops. All network segments must be connected directly to it. The Service Discovery Gateway can learn services from all connected segments to build its cache and respond to requests acting as a proxy.

- The use of third-party mDNS servers or applications are not supported with this feature.

Information about the Service Discovery Gateway and mDNS

mDNS

mDNS was defined to achieve zero configuration, with zero configuration being defined as providing the following features:

- Addressing—Allocating IP addresses to hosts

- Naming—Using names to refer to hosts instead of IP addresses

- Service discovery—Finding services automatically on the network

With mDNS, network users no longer have to assign IP addresses, assign host names, or type in names to access services on the network. Users only need to ask to see what network services are available, and choose from a list.

With mDNS, *addressing* is accomplished through the use of DHCP/DHCPv6 or IPv4 and IPv6 Link Local scoped addresses. The benefit of zero-configuration occurs when no infrastructure services such as DHCP or DNS are present and self-assigned link-local addressing can be used. The client can then select a random IPv4 address in the link-local range (169.254.0.0/24) or use its IPv6 link-local address (FE80::/10) for communication.

With mDNS, *naming* (name-to-address translation on a local network using mDNS) queries are sent over the local network using link-local scoped IP multicast. Because these DNS queries are sent to a multicast address (IPv4 address 224.0.0.251 or IPv6 address FF02::FB), no single DNS server with global knowledge is required to answer the queries. When a service or device sees a query for any service it is aware of, it provides a DNS response with the information from its cache.

With mDNS, *service discovery* is accomplished by browsing. An mDNS query is sent out for a given service type and domain, and any device that is aware of matching services replies with service information. The result is a list of available services for the user to choose from.

The mDNS protocol (mDNS-RFC), together with DNS Service Discovery (DNS-SD-RFC) achieves the zero-configuration addressing, naming, and service discovery.

mDNS-SD

Multicast DNS Service Discovery (mDNS-SD) uses DNS protocol semantics and multicast over well-known multicast addresses to achieve zero configuration service discovery. DNS packets are sent to and received on port 5353 using a multicast address of 224.0.0.251 and its IPv6 equivalent FF02::FB.

Because mDNS uses a link-local multicast address, its scope is limited to a single physical or logical LAN. If the networking reach needs to be extended to a distributed campus or to a wide-area environment consisting of many different networking technologies, mDNS gateway is implemented. An mDNS gateway provides a transport for mDNS packets across Layer 3 boundaries by filtering, caching, and redistributing services from one Layer 3 domain to another.

Service Discovery Gateway

The Service Discovery Gateway feature enables multicast Domain Name System (mDNS) to operate across Layer 3 boundaries (different subnets). An mDNS gateway provides transport for service discovery across Layer 3 boundaries by filtering, caching, and redistributing services from one Layer 3 domain (subnet) to another. Prior to implementation of this feature, mDNS was limited in scope to within a subnet because of the use of link-local scoped multicast addresses. This feature enhances Bring Your Own Device (BYOD).

mDNS Gateway and Subnets

You need to enable an mDNS gateway for service discovery to operate across subnets. You can enable mDNS gateway for a device or for an interface.

Note You need to configure service routing globally before configuring at the interface level.

After the device or interface is enabled, you can redistribute service discovery information across subnets. You can create service policies and apply filters on either incoming service discovery information (called IN-bound filtering) or outgoing service discovery information (called OUT-bound filtering).

Note   If redistribution is enabled globally, global configuration is given higher priority than interface configuration.

Figure 18: Sample Networking Scenario

For example, if the mDNS gateway functionality is enabled on the router in this figure, then service information can be sent from one subnet to another and vice-versa. For example, the printer and fax service information being advertised in the network with IP address 192.0.2.6 are redistributed to the network with IP address 198.51.100.4. The printer and fax service information in the network with IP address 192.0.2.6 is learned by mDNS-enabled hosts and devices in the other network.

Filtering

After configuring the mDNS gateway and subnets, you can filter services that you want to redistribute. While creating a service list, the **permit** or **deny** command options are used:

- The **permit** command option allows you to permit or transport specific service list information.

- The **deny** option allows you to deny service list information that is available to be transported to other subnets.

You need to include a sequence number when using the **permit** or **deny** command option. The same service list name can be associated with multiple sequence numbers and each sequence number will be mapped to a rule.

Note   If no filters are configured, then the default action is to deny service list information to be transported through the device or interface.

Query is another option provided when creating service lists. You can create queries using a service list. If you want to browse for a service, then active queries can be used. This function is helpful to keep the records refreshed in the cache.

Note   Active queries can only be used globally and cannot be used at the interface level.

A service end-point (such as a printer or fax) sends unsolicited announcements when a service starts up. After that, it sends unsolicited announcements whenever a network change event occurs (such as an interface coming up or going down). The device always respond to queries.

After creating a service list and using the **permit** or **deny** command options, you can filter using match statements (commands) based on *service-instance*, *service-type*, or *message-type* (announcement or query).

How to Configure the Service Discovery Gateway

Configuring the Service List

This procedure describes how to create a service list, apply a filter for the service list, and configure parameters for the service list name.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-list mdns-sd** *service-list-name* {**deny** *sequence-number* | **permit** *sequence-number* | **query**}
4. **match message-type** {**announcement** | **any** | **query**}
5. **match service-instance** { *LINE* }
6. **match service-type** {*LINE* }
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.

	Command or Action	Purpose
Step 3	**service-list mdns-sd** *service-list-name* {**deny** *sequence-number* | **permit** *sequence-number* | **query**} **Example:** Device(config)# **service-list mdns-sd sl1 permit 3** Device(config)# **service-list mdns-sd sl4 query**	Enters mDNS service discovery service list mode. In this mode, you can: • Create a service list and apply a filter on the service list according to the **permit** or **deny** option applied to the sequence number. • Create a service list and associate a query for the service list name if the **query** option is used. **Note** The sequence number sets the priority of the rule. A rule with a lower sequence number is selected first and the service announcement or query is allowed or denied accordingly. You define the sequence number as per your network requirements.
Step 4	**match message-type** {**announcement** | **any** | **query**} **Example:** Device(config-mdns-sd-sl)# **match message-type announcement**	(Optional) Sets the message type to match. You can match the following message types: • announcement • any • query These commands configure the parameters for the service list name that is created in step 2. If the **match message-type** is an announcement, then the service list rule only allows service advertisements or announcements for the device. If the **match message-type** is a query, then only a query from the client for a certain service in the network is allowed. Multiple service maps of the same name with different sequence numbers can be created and the evaluation of the filters will be ordered on the sequence number. Service lists are an ordered sequence of individual statements, each one has a permit or deny result. Evaluation of service list consists of a list scan, in a predetermined order, and an evaluation of the criteria of each statement that matches. A list scan is stopped once the first statement match is found and an action permit/deny associated with the statement match is performed. The default action after scanning through the entire list is to deny. **Note** You cannot use the **match** command if you have used the **query** option in the previous step. The **match** command can be used only for the **permit** or **deny** option.

	Command or Action	**Purpose**
Step 5	**match service-instance** { *LINE* } **Example:** Device(config-mdns-sd-sl)## **match service-instance servInst 1**	(Optional) Sets the service instance to match. This command configures the parameters for the service list name that is created in step 2. **Note** You cannot use the **match** command if you have used the **query** option in the previous step. The **match** command can be used only for the **permit** or **deny** option.
Step 6	**match service-type** {*LINE* } **Example:** Device(config-mdns-sd-sl)# **match service-type _ipp._tcp**	(Optional) Sets the value of the mDNS service type string to match. This command configures the parameters for the service list name that is created in step 2. **Note** You cannot use the **match** command if you have used the **query** option in the previous step. The **match** command can be used only for the **permit** or **deny** option.
Step 7	**end** **Example:** Device(config-mdns-sd-sl)# **end**	Returns to privileged EXEC mode.

What to do next

Proceed to enable the mDNS gateway and redistribution of services.

Enabling mDNS Gateway and Redistributing Services

After enabling mDNS gateway for a device, you can apply filters (apply IN-bound filtering or OUT-bound filtering) and active queries by using **service-policy** and **service-policy-query** commands, respectively. You can redistribute services and service announcements using the **redistribute mdns-sd** command, and set some part of the system memory for cache using the **cache-memory-max** command.

✎

Note By default, mDNS gateway is disabled on all interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-routing mdns-sd**
4. **service-policy** *service-policy-name* {**IN** | **OUT**}
5. **redistribute mdns-sd**

6. **cache-memory-max** *cache-config-percentage*
7. **service-policy-query** *service-list-query-name service-list-query-periodicity*
8. **exit**
9. **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	**service-routing mdns-sd** **Example:** Device (config)# **service-routing mdns-sd**	Enables mDNS gateway functionality for a device and enters multicast DNS configuration (config-mdns) mode. **Note** This command enables the mDNS function globally. **Note** Enter the **service-routing mdns-sd source-interface** *if-name* command in either global-config or interface-config mode, to specify an alternate source interface for outgoing mDNS packets, so its IP address can be used when there is none configured on the outgoing interface.
Step 4	**service-policy** *service-policy-name* {**IN** | **OUT**} **Example:** Device (config-mdns)# **service-policy serv-pol1 IN**	(Optional) For a service list, applies a filter on incoming service discovery information (IN-bound filtering) or outgoing service discovery information (OUT-bound filtering).
Step 5	**redistribute mdns-sd** **Example:** Device (config-mdns)# **redistribute mdns-sd**	(Optional) Redistributes services or service announcements across subnets. **Note** If redistribution is enabled globally, global configuration is given higher priority than interface configuration.
Step 6	**cache-memory-max** *cache-config-percentage* **Example:**	(Optional) Sets some part of the system memory (in percentage) for cache.

	Command or Action	**Purpose**
	Device (config-mdns)# **cache-memory-max 20**	**Note** By default, 10 percent of the system memory is set aside for cache. You can override the default value by using this command.
Step 7	**service-policy-query** *service-list-query-name* *service-list-query-periodicity* **Example:** Device (config-mdns)# **service-policy-query sl-query1 100**	(Optional) Configures service list-query periodicity.
Step 8	**exit** **Example:** Device (config-mdns)#**exit**	(Optional) Returns to global configuration mode.
Step 9	**end** **Example:** Device(config)# **end**	Returns to privileged EXEC mode.

Monitoring Service Discovery Gateway

Table 23: Monitoring Service Discovery Gateway

Command	**Purpose**
show mdns requests [**detail** | **name** *record-name*| **type** *record-type* [**name** *record-name*]]	This command displays information for outstanding mDNS requests, including record name and record type information.
show mdns cache [**interface** *type number* | **name** *record-name* [**type** *record-type*]| **type** *record-type*]	This command displays mDNS cache information.
show mdns statistics {**all** | **service-list** *list-name* | **service-policy** {**all** | **interface** *type number* }}	This command displays mDNS statistics.

Configuration Examples

Example: Specify Alternative Source Interface for Outgoing mDNS Packets

The following example displays how to specify an alternate source interface for outgoing mDNS packets, so its IP address can be used when there is none configured on the outgoing interface.

```
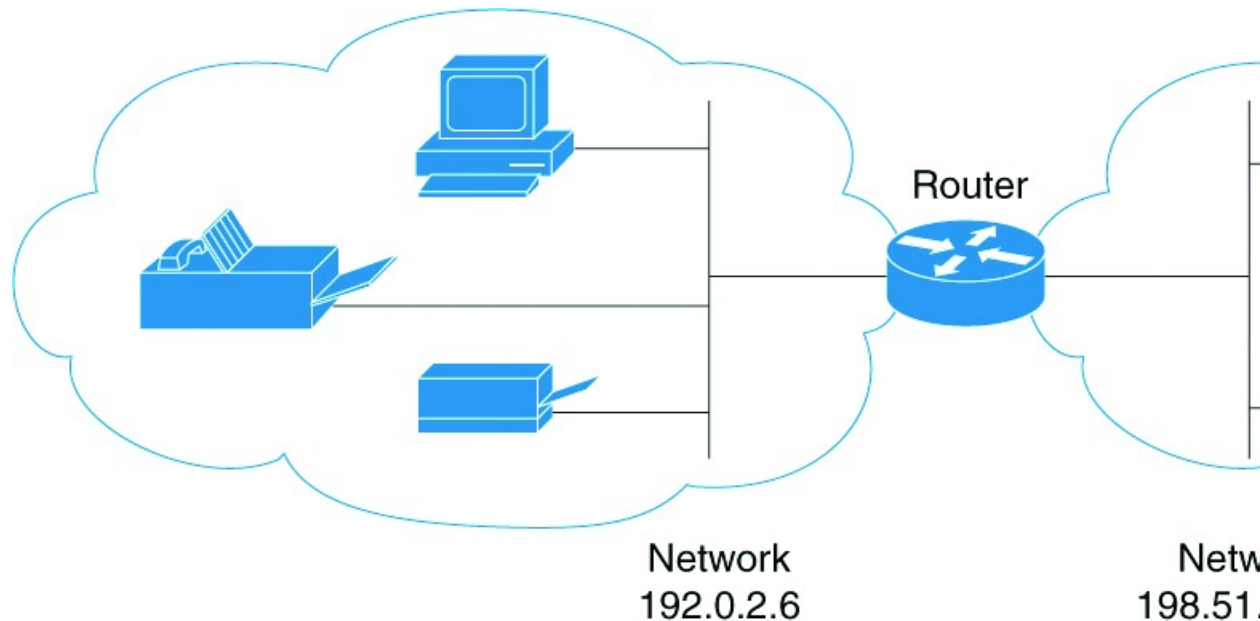Device(config)# service-routing mdns-sd
Device(config-mdns)# source-interface if-name
```

Example: Redistribute Service Announcements

The following example displays how to redistribute service announcements received on one interface over all the interfaces or over a specific interface.

```
Device(config)# service-routing mdns-sd
Device(config-mdns)# Redistribute mdns-sd if-name
```

Example: Creating a Service-List, Applying a Filter and Configuring Parameters

The following example shows the creation of a service-list sl1. The **permit** command option is being applied on sequence number 3 and all services with message-type announcement are filtered and available for transport across various subnets associated with the device.

```
Device# configure terminal
Device(config)# service-list mdns-sd sl1 permit 3
Device(config-mdns-sd-sl)#match message-type announcement
Device(config-mdns)# exit
```

Example: Enabling mDNS Gateway and Redistributing Services

The following example shows how to enable an mDNS gateway for a device and enable redistribution of services across subnets. IN-bound filtering is applied on the service-list serv-pol1. Twenty percent of system memory is made available for cache and service-list-query periodicity is configured at 100 seconds.

```
Device# configure terminal
Device# service-routing mdns-sd
Device(config-mdns)# service-policy serv-pol1 IN
Device(config-mdns)# redistribute mdns-sd
Device(config-mdns)# cache-memory-max 20
Device(config-mdns)# service-policy-query sl-query1 100
Device(config-mdns)# exit
```

Example: Global mDNS Configuration

The following example displays how to globally configure mDNS.

```
Device# configure terminal
Device(config)# service-list mdns-sd mypermit-all permit 10
Device(config-mdns-sd-s1)# exit
Device(config)# service-list mdns-sd querier query
Device(config-mdns-sd-s1)# service-type _dns._udp
Device(config-mdns-sd-s1)# end
Device# configure terminal
Device(config)# service-routing mdns-sd
Device(config-mdns)# service-policy mypermit-all IN
Device(config-mdns)# service-policy mypermit-all OUT
```

Example: Interface mDNS Configuration

The following example displays how to configure mDNS for an interface.

```
Device(config)#interface Vlan136
Device(config-if)# description *** Mgmt VLAN ***
Device(config-if)# ip address 9.7.136.10 255.255.255.0
Device(config-if)# ip helper-address 9.1.0.100
Device(config-if)# service-routing mdns-sd
Device(config-if-mdns-sd)#  service-policy mypermit-all IN
Device(config-if-mdns-sd)# service-policy mypermit-all OUT
Device(config-if-mdns-sd)#  service-policy-query querier 60
```

Where to Go Next for Configuring Services Discovery Gateway

You can configure the following:

Additional References

Related Documents

Related Topic	Document Title
Configuring DNS	*IP Addressing: DNS Configuration Guide, Cisco IOS XE Release 3SE*
DNS conceptual information	'Information About DNS' section in *IP Addressing: DNS Configuration Guide, Cisco IOS XE Release 3SE*
Platform-independent configuration information	*IP Addressing: DNS Configuration Guide, Cisco IOS XE Release 3SE*

Error Message Decoder

Description	Link
To help you research and resolve system error messages in this release, use the Error Message Decoder tool.	https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

Standards and RFCs

Standard/RFC	Title
RFC 6763	*DNS-Based Service Discovery*
Multicast DNS Internet-Draft	Multicast

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	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 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 History and Information for Services Discovery Gateway

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.

IP Multicast Optimization: Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

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 19: 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. • The group list is a standard access list that controls which groups the SPT threshold applies to. If a value

	Command or Action	**Purpose**
		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 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.

```
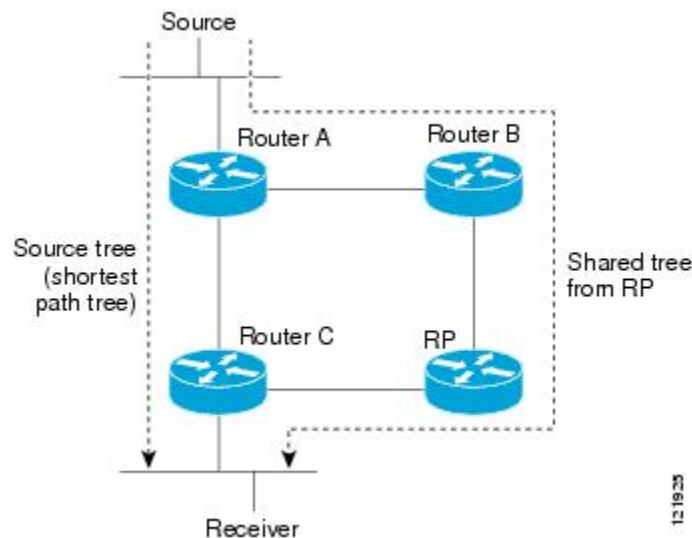interface ethernet 0
 ip pim query-interval 1
 .
 .
 .
 !
```

```
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 History and Information for Optimizing PIM Sparse Mode in a Large IP Multicast Deployment

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.

CHAPTER 15

IP Multicast Optimization: Multicast Subsecond Convergence

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.

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 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.
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.

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

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 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 History and Information for Multicast Subsecond Convergence

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.

CHAPTER 16

IP Multicast Optimization: IP Multicast Load Splitting across Equal-Cost Paths

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), 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, 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.

Note Although the following illustration and example uses routers in the configuration, any device (router or switch) can be used.

Figure 20: 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 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.

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, 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 300 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 300 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 301 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.

Note Although the following illustration and example uses routers in the configuration, any device (router or switch) can be used.

Figure 21: 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 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.

Note   Although the following illustration and example uses routers in the configuration, any device (router or switch) can be used.

Figure 22: 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.

	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.
Note	

- 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* 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.

	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.
Note	

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 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:** `Device(config)# ip multicast multipath`	Enables ECMP multicast load splitting based on source address using the S-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.

	Command or Action	**Purpose**
		• 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.
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.

	Command or Action	Purpose
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:** `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 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:

```
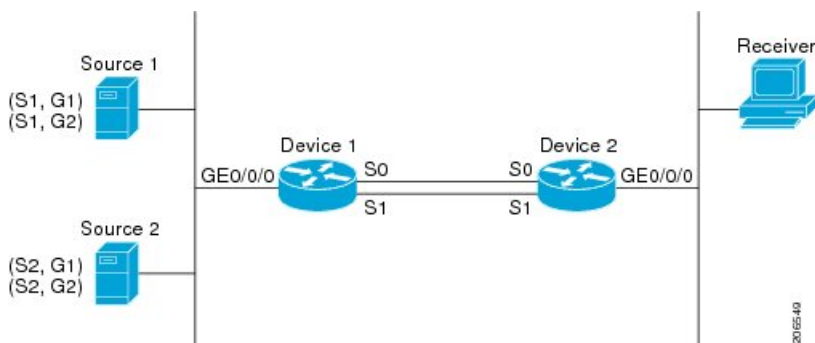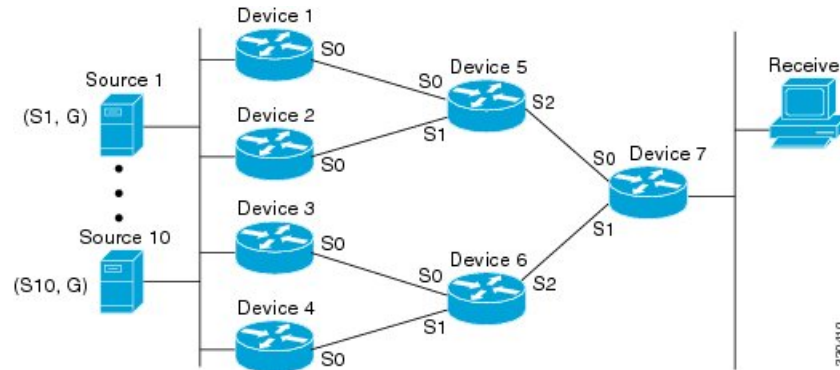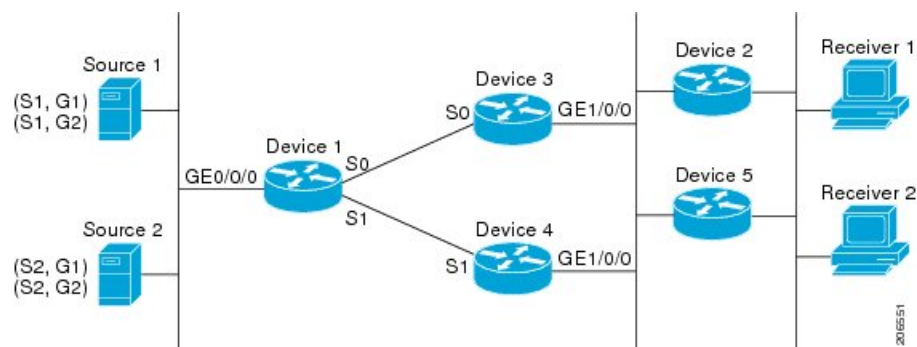ip multicast multipath
```

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 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 History and Information for Load Splitting IP Multicast Traffic over ECMP

Release	Modification
Cisco IOS XE 3.3SE	This feature was introduced.

CHAPTER **17**

IP Multicast Optimization: SSM Channel Based Filtering for Multicast

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.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
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.

```
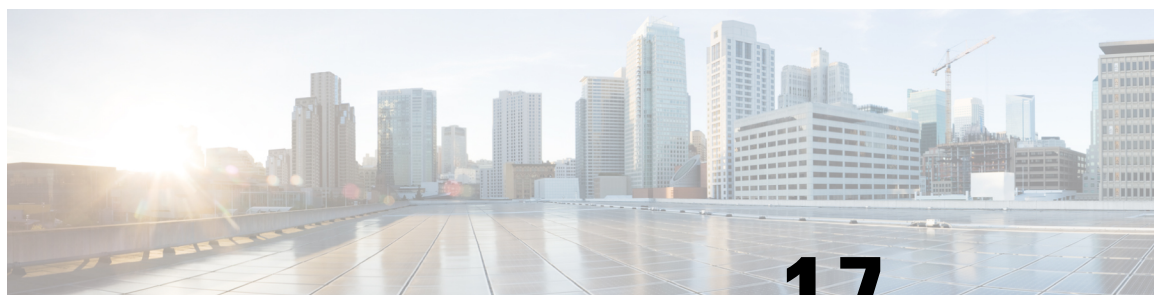configure terminal
 ip access-list extended acc_grp1
 permit ip host 0.0.0.0 232.1.1.1 0.0.0.255
 permit ip host 181.1.2.201 host 232.1.1.1
```

```
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 History and Information for SSM Channel Based Filtering for Multicast Boundaries

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.

IP Multicast Optimization: IGMP State Limit

Prerequisites for IGMP 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*.

- ALL ACLs must be configured. For information, see the see the " Creating an IP Access List and Applying It to an Interface " module of the *Security Configuration Guide: Access Control Lists* guide.

Restrictions for IGMP State Limit

You can configure only one global limit per device and one limit per interface.

Information About IGMP State Limit

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.

How to Configure IGMP State Limit

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.

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.

	Command or Action	**Purpose**
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.

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).

	Command or Action	**Purpose**

Step 5	Do one of the following:
	
	   • **exit**
	   • **end**
	
	Example:
	
	Device(config-if)# exit
	
	Device(config-if)# 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.
		
		• 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.

Configuration examples for IGMP State Limit

Configuring IGMP State Limiters Example

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.

Note   Although the following illustration and example uses routers in the configuration, any device (router or switch) can be used.

Figure 23: 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 router 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 router. 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 0/0/0 for the SD channels and Internet, Voice, and VoD services being offered to subscribers:

```
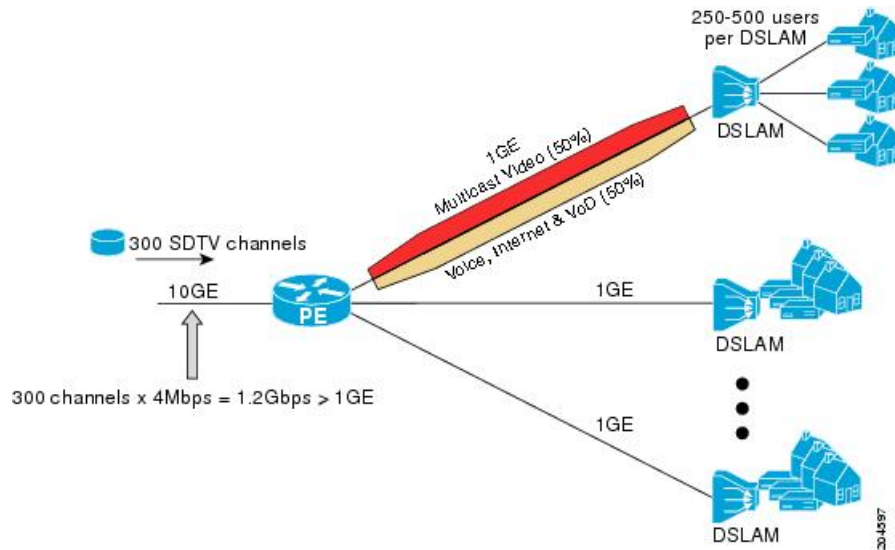interface GigabitEthernet0/0/0
description --- Interface towards the DSLAM ---
.
.
.
ip igmp limit 125
```

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	

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 History and Information for IGMP State Limit

Release	Modification
Cisco IOS XE 3.3SECisco IOS XE 3.3SE	This feature was introduced.