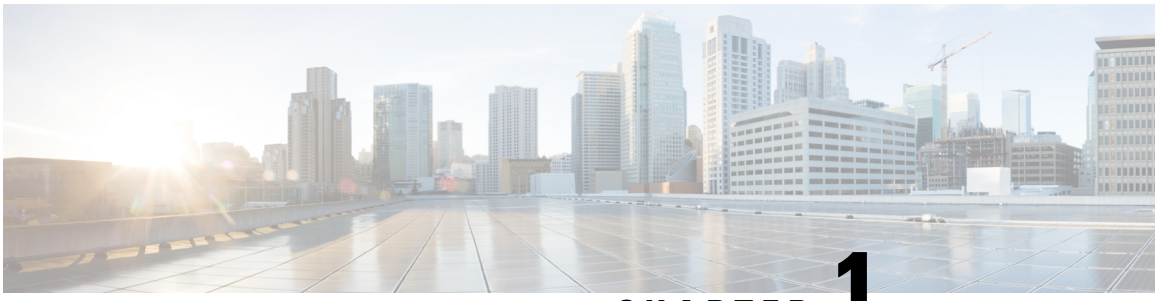
IP Multicast: PIM Configuration Guide, Cisco IOS XE Gibraltar 16.12.x (Cisco ASR 920 Series)

First Published: 2019-07-31

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

CONTENTS



CHAPTER 1

Configuring Basic IP Multicast

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

Prerequisites for Configuring Basic IP Multicast

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

Restrictions for Configuring Basic IP Multicast

HSRP/GLBP aware PIM is not supported.

Information About Configuring Basic IP Multicast

Auto-RP Overview

The Role of Auto-RP in a PIM Network

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

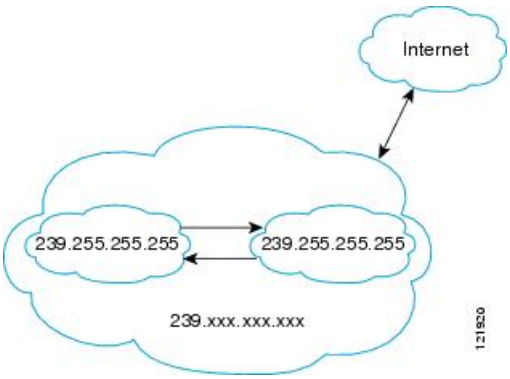
Thus, all routers automatically discover which RP to use for the groups they support. The Internet Assigned Numbers Authority (IANA) has assigned two group addresses, 224.0.1.39 and 224.0.1.40, for Auto-RP.

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

IP Multicast Boundary

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

Figure 1: Address Scoping at Boundaries



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

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

You can configure the **filter-autorp** keyword to examine and filter Auto-RP discovery and announcement messages at the administratively scoped boundary. Any Auto-RP group range announcements from the Auto-RP packets that are denied by the boundary access control list (ACL) are removed. An Auto-RP group

range announcement is permitted and passed by the boundary only if all addresses in the Auto-RP group range are permitted by the boundary ACL. If any address is not permitted, the entire group range is filtered and removed from the Auto-RP message before the Auto-RP message is forwarded.

Benefits of Auto-RP in a PIM Network

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

BSR Overview

BSR Election and Functionality

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

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

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

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

BSR Border Interface

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

Static RP Overview

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

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

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

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

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

SSM Overview

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

SSM Components

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

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

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

How SSM Differs from Internet Standard Multicast

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

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

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

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

SSM Operations

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

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

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

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

IGMPv3 Host Signaling

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

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

Benefits of Source Specific Multicast

IP Multicast Address Management Not Required

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

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

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

Denial of Service Attacks from Unwanted Sources Inhibited

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

Easy to Install and Manage

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

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

Ideal for Internet Broadcast Applications

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

- The ability to provide Internet broadcast services through SSM without the need for unique IP multicast addresses allows content providers to easily offer their service (IP multicast address allocation has been a serious problem for content providers in the past).

- The prevention against DoS attacks is an important factor for Internet broadcast services because, with their exposure to a large number of receivers, they are the most common targets for such attacks.
- The ease of installation and operation of SSM makes it ideal for network operators, especially in those cases where content needs to be forwarded between multiple independent PIM domains (because there is no need to manage MSDP for SSM between PIM domains).

How to Configure Basic IP Multicast

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

Configuring Sparse Mode with Auto-RP

Before you begin

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

Note

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

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

SUMMARY STEPS

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

10. Repeat Steps 1 through 9 on all PIM interfaces.
11. **ip pim send-rp-announce** {*interface-type interface-number* | *ip-address*} **scope** *ttl-value* [**group-list** *access-list*] [**interval** *seconds*] [**bidir**]
12. **ip pim send-rp-discovery** [*interface-type interface-number*] **scope** *ttl-value* [**interval** *seconds*]
13. **ip pim rp-announce-filter rp-list** *access-list* **group-list** *access-list*
14. **no ip pim dm-fallback**
15. **interface** *type number*
16. **ip multicast boundary** *access-list* [**filter-autorp**]
17. **end**
18. **show ip pim autorp**
19. **show ip pim rp** [**mapping**] [*rp-address*]
20. **show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**]
21. **show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing distributed** **Example:** `Device(config)# ip multicast-routing`	Enables IP multicast routing.
Step 4	Either perform Steps 5 through 7 or perform Steps 6 and 8.	--
Step 5	**ip pim autorp listener** **Example:** `Device(config)# ip pim autorp listener`	Causes IP multicast traffic for the two Auto-RP groups 224.0.1.39 and 224.0.1.40 to be PIM dense mode flooded across interfaces operating in PIM sparse mode. • Skip this step if you are configuring sparse-dense mode in Step 8.
Step 6	**interface** *type number* **Example:** `Device(config)# interface Gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which PIM can be enabled.

	Command or Action	Purpose
Step 7	**ip pim sparse-mode** **Example:** `Device(config-if)# ip pim sparse-mode`	Enables PIM sparse mode on an interface. When configuring Auto-RP in sparse mode, you must also configure the Auto-RP listener in the next step. • Skip this step if you are configuring sparse-dense mode in Step 8.
Step 8	**ip pim sparse-dense-mode** **Example:** `Device(config-if)# ip pim sparse-dense-mode`	Enables PIM sparse-dense mode on an interface. • Skip this step if you configured sparse mode in Step 7.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	Repeat Steps 1 through 9 on all PIM interfaces.	--
Step 11	**ip pim send-rp-announce** {*interface-type interface-number* | *ip-address*} **scope** *ttl-value* [**group-list** *access-list*] [**interval** *seconds*] [**bidir**] **Example:** `Device(config)# ip pim send-rp-announce loopback0 scope 31 group-list 5`	Sends RP announcements out all PIM-enabled interfaces. • Perform this step on the RP device only. • Use the *interface-type* and *interface-number* arguments to define which IP address is to be used as the RP address. • Use the *ip-address* argument to specify a directly connected IP address as the RP address. **Note** If the *ip-address* argument is configured for this command, the RP-announce message will be sourced by the interface to which this IP address is connected (that is, the source address in the IP header of the RP-announce message is the IP address of that interface). • This example shows that the interface is enabled with a maximum of 31 hops. The IP address by which the device wants to be identified as RP is the IP address associated with loopback interface 0. Access list 5 describes the groups for which this device serves as RP.
Step 12	**ip pim send-rp-discovery** [*interface-type interface-number*] **scope** *ttl-value* [**interval** *seconds*] **Example:** `Device(config)# ip pim send-rp-discovery loopback 1 scope 31`	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.

	Command or Action	Purpose
		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.
Step 13	**ip pim rp-announce-filter rp-list** *access-list* **group-list** *access-list* **Example:** `Device(config)# ip pim rp-announce-filter rp-list 1 group-list 2`	Filters incoming RP announcement messages sent from candidate RPs (C-RPs) to the RP mapping agent. • Perform this step on the RP mapping agent only.
Step 14	**no ip pim dm-fallback** **Example:** `Device(config)# no ip pim dm-fallback`	(Optional) Prevents PIM dense mode fallback. • Skip this step if all interfaces have been configured to operate in PIM sparse mode. **Note** The **no ip pim dm-fallback** command behavior is enabled by default if all the interfaces are configured to operate in PIM sparse mode (using the **ip pim sparse-mode** command).
Step 15	**interface** *type number* **Example:**	Selects an interface that is connected to hosts on which PIM can be enabled.

	Command or Action	Purpose
	`Device(config)# interface gigabitethernet 1/0/0`	
Step 16	**ip multicast boundary** *access-list* [**filter-autorp**] **Example:** `Device(config-if)# ip multicast boundary 10 filter-autorp`	Configures an administratively scoped boundary. • Perform this step on the interfaces that are boundaries to other devices. • The access list is not shown in this task. • An access list entry that uses the **deny** keyword creates a multicast boundary for packets that match that entry.
Step 17	**end** **Example:** `Device(config-if)# end`	Returns to global configuration mode.
Step 18	**show ip pim autorp** **Example:** `Device# show ip pim autorp`	(Optional) Displays the Auto-RP information.
Step 19	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `Device# show ip pim rp mapping`	(Optional) Displays RPs known in the network and shows how the device learned about each RP.
Step 20	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**] **Example:** `Device# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the device and that were learned through Internet Group Management Protocol (IGMP). • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 21	**show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*] **Example:** `Device# show ip mroute cbone-audio`	(Optional) Displays the contents of the IP multicast routing (mroute) table.

What to Do Next

Proceed to the " Verifying IP Multicast Operation " module.

Configuring Sparse Mode with a Single Static RP

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

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

Before you begin

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing distributed** **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables PIM on an interface. You must use sparse mode.
Step 6	Repeat Steps 1 through 5 on every interface that uses IP multicast.	--
Step 7	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 8	**ip pim rp-address** *rp-address* [*access-list*] [**override**] **Example:** `Router(config)# ip pim rp-address 192.168.0.0`	Configures the address of a PIM RP for a particular group. • The optional *access-list* argument is used to specify the number or name a standard access list that defines the multicast groups to be statically mapped to the RP. **Note** If no access list is defined, the RP will map to all multicast groups, 224/4. • The optional **override** keyword is used to specify that if dynamic and static group-to-RP mappings are used together and there is an RP address conflict, the RP address configured for a static group-to-RP mapping will take precedence. **Note** If the **override** keyword is not specified and there is RP address conflict, dynamic group-to-RP mappings will take precedence over static group-to-RP mappings.
Step 9	**end** **Example:** `Router(config)# end`	Ends the current configuration session and returns to EXEC mode.
Step 10	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `Router# show ip pim rp mapping`	(Optional) Displays RPs known in the network and shows how the router learned about each RP.

	Command or Action	Purpose
Step 11	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**] **Example:** `Router# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the router and that were learned through IGMP. • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 12	**show ip mroute** **Example:** `Router# show ip mroute`	(Optional) Displays the contents of the IP mroute table.

What to Do Next

Proceed to the " Verifying IP Multicast Operation " module.

Configuring Source Specific Multicast

This section describes how to configure Source Specific Multicast (SSM).

Before you begin

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	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 distributed** **Example:** `Device(config)# ip multicast-routing`	Enables IP multicast routing.
Step 4	**ip pim ssm** {**default** | **range** *access-list*} **Example:** `Device(config)# ip pim ssm default`	Configures SSM service. • The **default** keyword defines the SSM range access list as 232/8. • The **range** keyword specifies the standard IP access list number or name that defines the SSM range.
Step 5	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.
Step 6	**ip pim sparse-mode** **Example:** `Device(config-if)# ip pim sparse-mode`	Enables PIM on an interface. You must use sparse mode.
Step 7	Repeat Steps 1 through 6 on every interface that uses IP multicast.	--
Step 8	**ip igmp version 3** **Example:** `Device(config-if)# ip igmp version 3`	Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2. Version 3 is required by SSM.
Step 9	Repeat Step 8 on all host-facing interfaces.	--
Step 10	**end** **Example:** `Device(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 11	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**] **Example:**	(Optional) Displays the multicast groups having receivers that are directly connected to the device and that were learned through IGMP.

	Command or Action	Purpose
	`Device# show ip igmp groups`	• A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 12	**show ip mroute** **Example:** `Device# show ip mroute`	(Optional) Displays the contents of the IP mroute table. • This command displays whether a multicast group is configured for SSM service or a source-specific host report has been received.

What to Do Next

Proceed to the " Verifying IP Multicast Operation " module.

Configuration Examples for Basic IP Multicast

Example: Sparse Mode with Auto-RP

The following example configures sparse mode with Auto-RP:

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

BSR and RFC 2362 Interoperable Candidate RP Example

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

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

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

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

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

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

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

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

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

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

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

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

Example: Sparse Mode with a Single Static RP

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

```
ip multicast-routing
interface gigiabitethernet 1/0/0
 ip pim sparse-mode
ip pim rp-address 192.168.1.1
```

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

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

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

SSM with IGMPv3 Example

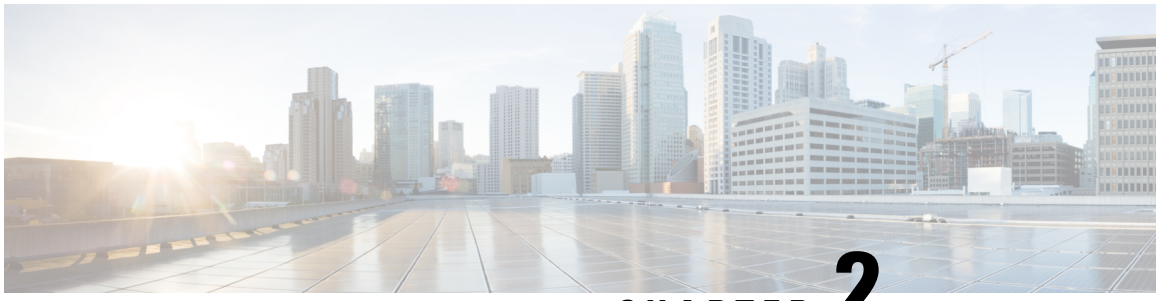
The following example shows how to configure a device (running IGMPv3) for SSM:

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

SSM Filtering Example

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

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



CHAPTER 2

Configuring Source Specific Multicast

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

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Source Specific Multicast

Legacy Applications Within the SSM Range Restrictions

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

IGMP v3lite and URD Require a Cisco Last Hop Router

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

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

Address Management Restrictions

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

IGMP Snooping and CGMP Limitations

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

URD Intercept URL Limitations

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

State Maintenance Limitations

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

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

HSIL Limitations

As explained in the IGMP v3lite Host Signalling, on page 24 concept, the HSIL tries to determine if the host operating system supports IGMPv3. This check is made so that a single application can be used both on hosts where the operating system has been upgraded to IGMPv3 and on hosts where the operating system only supports IGMPv1 or IGMPv2.

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

Information About Source Specific Multicast

SSM Overview

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

SSM Components

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

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

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

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

How SSM Differs from Internet Standard Multicast

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

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

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

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

SSM Operations

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

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

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

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

IGMPv3 Host Signaling

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

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

Benefits of Source Specific Multicast

IP Multicast Address Management Not Required

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

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

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

Denial of Service Attacks from Unwanted Sources Inhibited

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

Easy to Install and Manage

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

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

Ideal for Internet Broadcast Applications

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

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

IGMP v3lite Host Signalling

IGMP v3lite is a Cisco-developed transitional solution for application developers to immediately start programming SSM applications. It allows you to write and run SSM applications on hosts that do not yet support IGMPv3 in their operating system kernel.

Applications must be compiled with the Host Side IGMP Library (HSIL) for IGMP v3lite. This software provides applications with a subset of the IGMPv3 applications programming interface (API) that is required to write SSM applications. HSIL was developed for Cisco by Talarian and is available from the following web page:

http://www.talarianmulticast.com/cgi-bin/igmpdownld

One part of the HSIL is a client library linked to the SSM application. It provides the SSM subset of the IGMPv3 API to the SSM application. If possible, the library checks whether the operating system kernel supports IGMPv3. If it does, then the API calls simply are passed through to the kernel. If the kernel does not support IGMPv3, then the library uses the IGMP v3lite mechanism.

When using the IGMP v3lite mechanism, the library tells the operating system kernel to join to the whole multicast group, because joining to the whole group is the only method for the application to receive traffic for that multicast group (if the operating system kernel only supports IGMPv1 or IGMPv2). In addition, the library signals the (S, G) channel subscriptions to an IGMP v3lite server process, which is also part of the HSIL. A server process is needed because multiple SSM applications may be on the same host. This server process will then send IGMP v3lite-specific (S, G) channel subscriptions to the last hop Cisco IOS router, which needs to be enabled for IGMP v3lite. This router will then “see” both the IGMPv1 or IGMPv2 group membership report from the operating system kernel and the (S, G) channel subscription from the HSIL daemon. If the router sees both of these messages, it will interpret them as an SSM (S, G) channel subscription and join to the channel through PIM-SSM. We recommend referring to the documentation accompanying the HSIL software for further information on how to utilize IGMP v3lite with your application.

IGMP v3lite is supported by Cisco only through the API provided by the HSIL, not as a function of the router independent of the HSIL. By default, IGMP v3lite is disabled. When IGMP v3lite is configured through the **ip igmp v3lite** interface configuration command on an interface, it will be active only for IP multicast addresses in the SSM range.

How to Configure Source Specific Multicast

Configuring SSM

To configure SSM, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. **ip pim ssm** [**default** | **range***access-list*]
2. **interface** *type number*
3. **ip pim** {**sparse-mode** | **sparse-dense-mode**}
4. Do one of the following:
   - **ip igmp version 3**
   - **ip igmp v3lite**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ip pim ssm** [**default**	**range***access-list*] **Example:** `Router(config)# ip pim ssm default`	Defines the SSM range of IP multicast addresses.
Step 2	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/1`	Selects an interface that is connected to hosts on which IGMPv3, IGMP v3lite, and URD can be enabled.
Step 3	**ip pim** {**sparse-mode**	**sparse-dense-mode**} **Example:** `Router(config-if)# ip pim sparse-mode`	Enables PIM on an interface. You must use either sparse mode or sparse-dense mode.

Step 4	Do one of the following:
	• **ip igmp version 3**
	• **ip igmp v3lite**
	Example:
	`Router(config-if)# ip igmp version 3`
	`or`
	`Router(config-if)# ip igmp v3lite`
		Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2.
		or
		Enables the acceptance and processing of IGMP v3lite membership reports on an interface.
		or
		Enables interception of TCP packets sent to the reserved URD port 465 on an interface and processing of URD channel subscription reports.

Monitoring SSM

In Native Multicast, when IGMPv3 multicast groups with PIM-SSM and IGMP Snooping configurations (Layer2 multicast and Layer3 multicast) are enabled together, the maximum supported scale is 1938 groups.

Command	Purpose
`Router# show ip igmp groups detail`	Displays the (S, G) channel subscription through IGMPv3 or IGMP v3lite.
`Router# show ip mroute`	Displays whether a multicast group supports SSM service or whether a source-specific host report was received.

Configuration Examples of Source Specific Multicast

SSM with IGMPv3 Example

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

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

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mcl/allreleasemcl/all-book.html

Standards and RFCs

Standard/RFC	Title
No specific Standards and RFCs are supported by the features in this document.	—

MIBs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

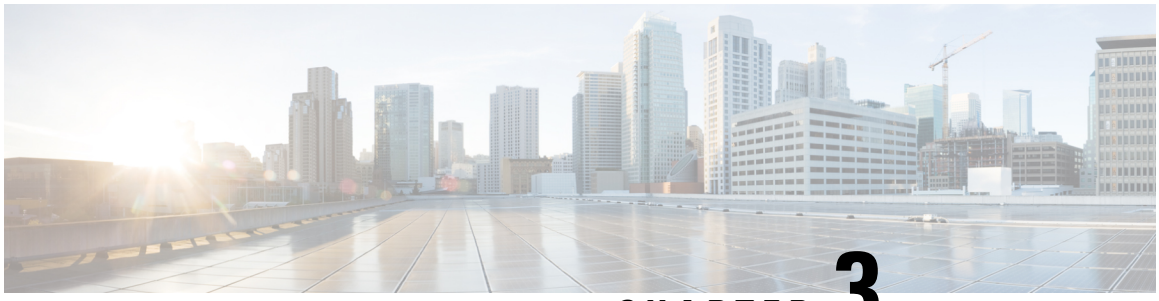
Feature Information for Configuring SSM

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 1: Feature Information for Configuring SSM

Feature Name	Releases	Feature Information
Configuring SSM	Cisco IOS XE Release 3.14.0S	This feature was introduced on the Cisco ASR 920 Series Aggregation Services Router (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D, ASR-920-10SZ-PD, ASR-920-24SZ-IM, ASR-920-24SZ-M, ASR-920-24TZ-M) .



CHAPTER 3

SSM Mapping

The Source Specific Multicast (SSM) Mapping feature extends the Cisco suite of SSM transition tools, which also includes URL Rendezvous Directory (URD) and Internet Group Management Protocol Version 3 Lite (IGMP v3lite). SSM mapping supports SSM transition in cases where neither URD nor IGMP v3lite is available, or when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. SSM mapping enables you to leverage SSM for video delivery to legacy set-top boxes (STBs) that do not support IGMPv3 or for applications that do not take advantage of the IGMPv3 host stack.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SSM Mapping

One option available for using SSM mapping is to install it together with a Domain Name System (DNS) server to simplify administration of the SSM Mapping feature in larger deployments.

Before you can configure and use SSM mapping with DNS lookups, you need to add records to a running DNS server. If you do not already have a DNS server running, you need to install one.

Restrictions for SSM Mapping

- The SSM Mapping feature does not share the benefit of full SSM. SSM mapping takes a group G join from a host and identifies this group with an application associated with one or more sources, therefore, it can only support one such application per group G. Nevertheless, full SSM applications may still share the same group also used in SSM mapping.
- Enable IGMPv3 with care on the last hop router when you rely solely on SSM mapping as a transition solution for full SSM.

Information About SSM Mapping

SSM Components

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

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

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

Benefits of Source Specific Multicast

IP Multicast Address Management Not Required

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

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

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

Denial of Service Attacks from Unwanted Sources Inhibited

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

Easy to Install and Manage

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

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

Ideal for Internet Broadcast Applications

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

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

SSM Transition Solutions

The Cisco IOS suite of SSM transition solutions consists of the following transition solutions that enable the immediate development and deployment of SSM services, without the need to wait for the availability of full IGMPv3 support in host operating systems and SSM receiver applications:

- Internet Group Management Protocol Version 3 lite (IGMP v3lite)
- URL Rendezvous Directory (URD)
- SSM mapping

IGMP v3lite is a solution for application developers that allows immediate development of SSM receiver applications switching to IGMPv3 as soon as it becomes available.

For more information about IGMP v3lite, see the " Configuring Source Specific Multicast " module.

URD is an SSM transition solution for content providers and content aggregators that allows them to deploy receiver applications that are not yet SSM enabled (through support for IGMPv3) by enabling the receiving applications to be started and controlled through a web browser.

For more information about URD, see the see the " Configuring Source Specific Multicast " module.

SSM mapping supports SSM transition in cases where neither URD nor IGMP v3lite are available, or when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons.

SSM Mapping Overview

SSM mapping supports SSM transition when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. Using SSM to deliver live streaming video to legacy STBs that do not support IGMPv3 is a typical application of SSM mapping.

In a typical STB deployment, each TV channel uses one separate IP multicast group and has one active server host sending the TV channel. A single server may of course send multiple TV channels, but each to a different group. In this network environment, if a router receives an IGMPv1 or IGMPv2 membership report for a particular group G, the report implicitly addresses the well-known TV server for the TV channel associated with the multicast group.

SSM mapping introduces a means for the last hop router to discover sources sending to groups. When SSM mapping is configured, if a router receives an IGMPv1 or IGMPv2 membership report for a particular group G, the router translates this report into one or more (S, G) channel memberships for the well-known sources associated with this group.

When the router receives an IGMPv1 or IGMPv2 membership report for group G, the router uses SSM mapping to determine one or more source IP addresses for group G. SSM mapping then translates the membership report as an IGMPv3 report INCLUDE (G, [S1, G], [S2, G]...[Sn, G] and continues as if it had received an IGMPv3 report. The router then sends out PIM joins toward (S1, G) to (Sn, G) and continues to be joined to these groups as long as it continues to receive the IGMPv1 or IGMPv2 membership reports and as long as the SSM mapping for the group remains the same. SSM mapping, thus, enables you to leverage SSM for video delivery to legacy STBs that do not support IGMPv3 or for applications that do not take advantage of the IGMPv3 host stack.

SSM mapping enables the last hop router to determine the source addresses either by a statically configured table on the router or by consulting a DNS server. When the statically configured table is changed, or when the DNS mapping changes, the router will leave the current sources associated with the joined groups.

Static SSM Mapping

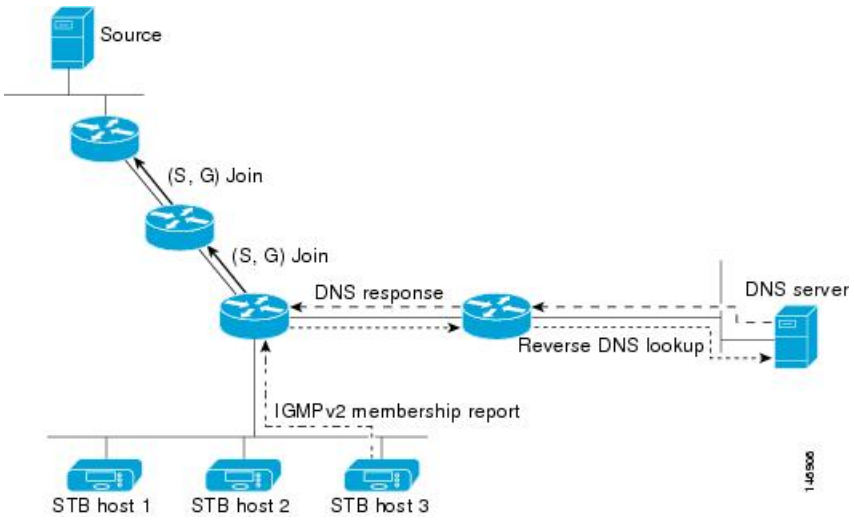
SSM static mapping enables you to configure the last hop router to use a static map to determine the sources sending to groups. Static SSM mapping requires that you configure access lists (ACLs) to define group ranges. The groups permitted by those ACLs then can be mapped to sources using the **ip igmp static ssm-map** global configuration command.

You can configure static SSM mapping in smaller networks when a DNS is not needed or to locally override DNS mappings that may be temporarily incorrect. When configured, static SSM mappings take precedence over DNS mappings.

DNS-Based SSM Mapping

DNS-based SSM mapping enables you to configure the last hop router to perform a reverse DNS lookup to determine sources sending to groups (see the figure below). When DNS-based SSM mapping is configured, the router constructs a domain name that includes the group address G and performs a reverse lookup into the DNS. The router looks up IP address resource records (IP A RRs) to be returned for this constructed domain name and uses the returned IP addresses as the source addresses associated with this group. SSM mapping supports up to 20 sources for each group. The router joins all sources configured for a group.

Figure 2: DNS-Based SSM-Mapping



The SSM mapping mechanism that enables the last hop router to join multiple sources for a group can be used to provide source redundancy for a TV broadcast. In this context, the redundancy is provided by the last hop router using SSM mapping to join two video sources simultaneously for the same TV channel. However, to prevent the last hop router from duplicating the video traffic, it is necessary that the video sources utilize a server-side switchover mechanism where one video source is active while the other backup video source is passive. The passive source waits until an active source failure is detected before sending the video traffic for the TV channel. The server-side switchover mechanism, thus, ensures that only one of the servers is actively sending the video traffic for the TV channel.

To look up one or more source addresses for a group G that includes G1, G2, G3, and G4, the following DNS resource records (RRs) must be configured on the DNS server:

G4.G3.G2.G1 [*multicast-domain*] [*timeout*]	IN A *source-address-1*
	IN A *source-address-2*
	IN A *source-address-n*

The *multicast-domain* argument is a configurable DNS prefix. The default DNS prefix is in-addr.arpa. You should only use the default prefix when your installation is either separate from the internet or if the group names that you map are global scope group addresses (RFC 2770 type addresses that you configure for SSM) that you own.

The *timeout* argument configures the length of time for which the router performing SSM mapping will cache the DNS lookup. This argument is optional and defaults to the timeout of the zone in which this entry is configured. The timeout indicates how long the router will keep the current mapping before querying the DNS

server for this group. The timeout is derived from the cache time of the DNS RR entry and can be configured for each group/source entry on the DNS server. You can configure this time for larger values if you want to minimize the number of DNS queries generated by the router. Configure this time for a low value if you want to be able to quickly update all routers with new source addresses.

Note Refer to your DNS server documentation for more information about configuring DNS RRs.

To configure DNS-based SSM mapping in the software, you must configure a few global commands but no per-channel specific configuration is needed. There is no change to the configuration for SSM mapping if additional channels are added. When DNS-based SSM mapping is configured, the mappings are handled entirely by one or more DNS servers. All DNS techniques for configuration and redundancy management can be applied to the entries needed for DNS-based SSM mapping.

SSM Mapping Benefits

- The SSM Mapping feature provides almost the same ease of network installation and management as a pure SSM solution based on IGMPv3. Some additional configuration is necessary to enable SSM mapping.
- The SSM benefit of inhibition of DoS attacks applies when SSM mapping is configured. When SSM mapping is configured the only segment of the network that may still be vulnerable to DoS attacks are receivers on the LAN connected to the last hop router. Since those receivers may still be using IGMPv1 and IGMPv2, they are vulnerable to attacks from unwanted sources on the same LAN. SSM mapping, however, does protect those receivers (and the network path leading towards them) from multicast traffic from unwanted sources anywhere else in the network.
- Address assignment within a network using SSM mapping needs to be coordinated, but it does not need assignment from outside authorities, even if the content from the network is to be transited into other networks.

How to Configure SSM Mapping

Configuring Static SSM Mapping

Perform this task to configure the last hop router in an SSM deployment to use static SSM mapping to determine the IP addresses of sources sending to groups.

Before you begin

- Enable IP multicast routing, enable PIM sparse mode, and configure SSM before performing this task. For more information, see the “Configuring Basic Multicast ”module.
- Before you configure static SSM mapping, you must configure ACLs that define the group ranges to be mapped to source addresses.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ip igmp ssm-map enable**
4. **no ip igmp ssm-map query dns**
5. **ip igmp ssm-map static** *access-list source-address*
6. Repeat Step 5 to configure additional static SSM mappings, if required.
7. **end**
8. **show running-config**
9. **copy running-config start-up config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp ssm-map enable** **Example:** `Device(config)# ip igmp ssm-map enable`	Enables SSM mapping for groups in the configured SSM range. **Note** By default, this command enables DNS-based SSM mapping.
Step 4	**no ip igmp ssm-map query dns** **Example:** `Device(config)# no ip igmp ssm-map query dns`	(Optional) Disables DNS-based SSM mapping. **Note** Disable DNS-based SSM mapping if you only want to rely on static SSM mapping. By default, the **ip igmp ssm-map** command enables DNS-based SSM mapping.
Step 5	**ip igmp ssm-map static** *access-list source-address* **Example:** `Device(config)# ip igmp ssm-map static 11 172.16.8.11`	Configures static SSM mapping. • The ACL supplied for the *access-list* argument defines the groups to be mapped to the source IP address entered for the *source-address* argument. **Note** You can configure additional static SSM mappings. If additional SSM mappings are configured and the router receives an IGMPv1 or IGMPv2 membership report for a group in the SSM range, the Cisco IOS XE software determines the source addresses associated with the group by walking each configured **ip igmp ssm-map static** command. The Cisco IOS XE software associates up to 20 sources per group.

	Command or Action	Purpose
Step 6	Repeat Step 5 to configure additional static SSM mappings, if required.	--
Step 7	**end** **Example:** `Device(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 8	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 9	**copy running-config start-up config** **Example:** `Device# copy running-config start-up config`	(Optional) Saves your entries in the configuration file.

What to Do Next

Proceed to the Configuring DNS-Based SSM Mapping (CLI), on page 36 or to the Verifying SSM Mapping Configuration and Operation, on page 39.

Configuring DNS-Based SSM Mapping (CLI)

Perform this task to configure the last hop router to perform DNS lookups to learn the IP addresses of sources sending to a group.

Before you begin

- Enable IP multicast routing, enable PIM sparse mode, and configure SSM before performing this task. For more information, see the "Configuring Basic Multicast"module.
- Before you can configure and use SSM mapping with DNS lookups, you need to be able to add records to a running DNS server. If you do not already have a DNS server running, you need to install one.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp ssm-map enable**
4. **ip igmp ssm-map query dns**
5. **ip domain multicast** *domain-prefix*
6. **ipname-server** *server-address1* [*server-address2server-address6*]
7. Repeat Step 6 to configure additional DNS servers for redundancy, if required.
8. **end**
9. **show running-config**

10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp ssm-map enable** **Example:** `Device(config)# ip igmp ssm-map enable`	Enables SSM mapping for groups in a configured SSM range.
Step 4	**ip igmp ssm-map query dns** **Example:** `Device(config)# ip igmp ssm-map query dns`	(Optional) Enables DNS-based SSM mapping. • By default, the **ip igmp ssm-map** command enables DNS-based SSM mapping. Only the **no**form of this command is saved to the running configuration. **Note** Use this command to reenable DNS-based SSM mapping if DNS-based SSM mapping is disabled.
Step 5	**ip domain multicast** *domain-prefix* **Example:** `Device(config)# ip domain multicast ssm-map.cisco.com`	(Optional) Changes the domain prefix used by the Cisco IOS XE software for DNS-based SSM mapping. • By default, the software uses the ip-addr.arpa domain prefix.
Step 6	**ipname-server** *server-address1* [*server-address2server-address6*] **Example:** `Device(config)# ip name-server 10.48.81.21`	Specifies the address of one or more name servers to use for name and address resolution.
Step 7	Repeat Step 6 to configure additional DNS servers for redundancy, if required.	--
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 9	**show running-config** **Example:**	Verifies your entries.

	Command or Action	Purpose
	`Device# show running-config`	
Step 10	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

What to Do Next

Configuring Static Traffic Forwarding with SSM Mapping

Perform this task to configure static traffic forwarding with SSM mapping on the last hop router. Static traffic forwarding can be used in conjunction with SSM mapping to statically forward SSM traffic for certain groups. When static traffic forwarding with SSM mapping is configured, the last hop router uses DNS-based SSM mapping to determine the sources associated with a group. The resulting (S, G) channels are then statically forwarded.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp static-group** *group-address* **source ssm-map**
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	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Selects an interface on which to statically forward traffic for a multicast group using SSM mapping and enters interface configuration mode. **Note** Static forwarding of traffic with SSM mapping works with either DNS-based SSM mapping or statically-configured SSM mapping.
Step 4	**ip igmp static-group** *group-address* **source ssm-map** **Example:**	Configures SSM mapping to be used to statically forward a (S, G) channel out of the interface.

	Command or Action	Purpose
	`Device(config-if)# ip igmp static-group 232.1.2.1 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-if)# end`	Returns to privileged EXEC mode.

What to Do Next

Proceed to the Verifying SSM Mapping Configuration and Operation, on page 39.

Verifying SSM Mapping Configuration and Operation

Perform this optional task to verify SSM mapping configuration and operation.

SUMMARY STEPS

1. **enable**
2. **show ip igmp ssm-mapping**
3. **show ip igmp ssm-mapping** *group-address*
4. **show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**]
5. **show host**
6. **debug ip igmp** *group-address*

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
> enable
```

Step 2 **show ip igmp ssm-mapping**

(Optional) Displays information about SSM mapping.

The following example shows how to display information about SSM mapping configuration. In this example, SSM static mapping and DNS-based SSM mapping are enabled.

Example:

```
# show ip igmp ssm-mapping
SSM Mapping  : Enabled
DNS Lookup   : Enabled
```

```
Mcast domain : ssm-map.cisco.com
Name servers : 10.0.0.3
               10.0.0.4
```

Step 3 **show ip igmp ssm-mapping** *group-address*

(Optional) Displays the sources that SSM mapping uses for a particular group.

The following example shows how to display information about the configured DNS-based SSM mapping. In this example, the router has used DNS-based mapping to map group 232.1.1.4 to sources 172.16.8.5 and 172.16.8.6. The timeout for this entry is 860000 milliseconds (860 seconds).

Example:

```
# show ip igmp ssm-mapping 232.1.1.4
Group address: 232.1.1.4
Database     : DNS
DNS name     : 4.1.1.232.ssm-map.cisco.com
Expire time  : 860000
Source list  : 172.16.8.5
             : 172.16.8.6
```

Step 4 **show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**]

(Optional) Displays the multicast groups with receivers that are directly connected to the router and that were learned through IGMP.

The following is sample output from the **show ip igmp groups** command with the *group-address* argument and **detail** keyword. In this example the “M” flag indicates that SSM mapping is configured.

Example:

```
# show ip igmp group 232.1.1.4 detail
            Interface:      GigabitEthernet2/0/0
            Group:          232.1.1.4 SSM
            Uptime:         00:03:20
            Group mode:     INCLUDE
            Last reporter:  0.0.0.0
            CSR Grp Exp:    00:02:59
            Group source list: (C - Cisco Src Report, U - URD, R - Remote,
                                S - Static, M - SSM Mapping)
              Source Address   Uptime    v3 Exp   CSR Exp   Fwd  Flags
              172.16.8.3       00:03:20  stopped   00:02:59  Yes  CM
              172.16.8.4       00:03:20  stopped   00:02:59  Yes  CM
              172.16.8.5       00:03:20  stopped   00:02:59  Yes  CM
              172.16.8.6       00:03:20  stopped   00:02:59  Yes  CM
```

Step 5 **show host**

(Optional) Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of hostnames and addresses.

The following is sample output from the **show host**command. Use this command to display DNS entries as they are learned by the router.

Example:

```
# show host
Default domain is cisco.com
Name/address lookup uses domain service
Name servers are 10.48.81.21
Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
```

```
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined
Host                       Port   Flags      Age   Type    Address(es)
10.0.0.0.ssm-map.cisco.c  None  (temp, OK)   0     IP     172.16.8.5
                                                         172.16.8.6
                                                         172.16.8.3
```

172.16.8.4

Step 6 **debug ip igmp** *group-address*

(Optional) Displays the IGMP packets received and sent and IGMP host-related events.

The following is sample output from the **debug ip igmp**command when SSM static mapping is enabled. The following output indicates that the router is converting an IGMPv2 join for group G into an IGMPv3 join:

Example:

```
IGMP(0): Convert IGMPv2 report (*,232.1.2.3) to IGMPv3 with 2 source(s) using STATIC.
```

The following is sample output from the **debug ip igmp** command when DNS-based SSM mapping is enabled. The following output indicates that a DNS lookup has succeeded:

Example:

```
IGMP(0): Convert IGMPv2 report (*,232.1.2.3) to IGMPv3 with 2 source(s) using DNS.
```

The following is sample output from the **debug ip igmp** command when DNS-based SSM mapping is enabled and a DNS lookup has failed:

IGMP(0): DNS source lookup failed for (*, 232.1.2.3), IGMPv2 report failed

Configuration Examples for SSM Mapping

SSM Mapping Example

The following configuration example shows a router configuration for SSM mapping. This example also displays a range of other IGMP and SSM configuration options to show compatibility between features. Do not use this configuration example as a model unless you understand all of the features used in the example.

Note Address assignment in the global SSM range 232.0.0.0/8 should be random. If you copy parts or all of this sample configuration, make sure to select a random address range but not 232.1.1.x as shown in this example. Using a random address range minimizes the possibility of address collision and may prevent conflicts when other SSM content is imported while SSM mapping is used.

```
!
no ip domain lookup
ip domain multicast ssm.map.cisco.com
ip name-server 10.48.81.21
!
!
```

```
ip multicast-routing distributed
ip igmp ssm-map enable
ip igmp ssm-map static 10 172.16.8.10
ip igmp ssm-map static 11 172.16.8.11
!
!
.
.
.
!
interface GigabitEthernet0/0/0
 description Sample IGMP Interface Configuration for SSM-Mapping Example
 ip address 10.20.1.2 255.0.0.0
 ip pim sparse-mode
 ip igmp last-member-query-interval 100
 ip igmp static-group 232.1.2.1 source ssm-map
 ip igmp version 3
 ip igmp explicit-tracking
 ip igmp limit 2
 ip igmp v3lite
 ip urd
!
.
.
.
!
ip pim ssm default
!
access-list 10 permit 232.1.2.10
access-list 11 permit 232.1.2.0 0.0.0.255
!
```

This table describes the significant commands shown in the SSM mapping configuration example.

Table 2: SSM Mapping Configuration Example Command Descriptions

Command	Description
no ip domain lookup	Disables IP DNS-based hostname-to-address translation. **Note** The **no ip domain-list** command is shown in the configuration only to demonstrate that disabling IP DNS-based hostname-to-address translation does not conflict with configuring SSM mapping. If this command is enabled, the Cisco IOS XE software will try to resolve unknown strings as hostnames.
ip domain multicast ssm-map.cisco.com	Specifies ssm-map.cisco.com as the domain prefix for SSM mapping.
ip name-server 10.48.81.21	Specifies 10.48.81.21 as the IP address of the DNS server to be used by SSM mapping and any other service in the software that utilizes DNS.
ip multicast-routing	Enables IP multicast routing.
ip igmp ssm-map enable	Enables SSM mapping.

Command	Description
ip igmp ssm-map static 10 172.16.8.10	Configures the groups permitted by ACL 10 to use source address 172.16.8.10. • In this example, ACL 10 permits all groups in the 232.1.2.0/25 range except 232.1.2.10.
ip igmp ssm-map static 11 172.16.8.11	Configures the groups permitted by ACL 11 to use source address 172.16.8.11. • In this example, ACL 11 permits group 232.1.2.10.
ip pim sparse-mode	Enables PIM sparse mode.
ip igmp last-member-query-interval 100	Reduces the leave latency for IGMPv2 hosts. **Note** This command is not required for configuring SSM mapping; however, configuring this command can be beneficial for IGMPv2 hosts relying on SSM mapping.
ip igmp static-group 232.1.2.1 source ssm-map	Configures SSM mapping to be used to determine the sources associated with group 232.1.2.1. The resulting (S, G) channels are statically forwarded.
ip igmp version 3	Enables IGMPv3 on this interface. **Note** This command is shown in the configuration only to demonstrate that IGMPv3 can be configured simultaneously with SSM mapping; however, it is not required.
ip igmp explicit-tracking	Minimizes the leave latency for IGMPv3 host leaving a multicast channel. **Note** This command is not required for configuring SSM mapping.
ip igmp limit 2	Limits the number of IGMP states resulting from IGMP membership states on a per-interface basis. **Note** This command is not required for configuring SSM mapping.
ip igmp v3lite	Enables the acceptance and processing of IGMP v3lite membership reports on this interface. **Note** This command is shown in the configuration only to demonstrate that IGMP v3lite can be configured simultaneously with SSM mapping; however, it is not required.
ip urd	Enables interception of TCP packets sent to the reserved URD port 465 on an interface and processing of URD channel subscription reports. **Note** This command is shown in the configuration only to demonstrate that URD can be configured simultaneously with SSM mapping; however, it is not required.

Command	Description
ip pim ssm default	Configures SSM service. The **default** keyword defines the SSM range access list as 232/8.
access-list 10 permit 232.1.2.10 **access-list 11 permit 232.1.2.0 0.0.0.255**	Configures the ACLs to be used for static SSM mapping. **Note** These are the ACLs that are referenced by the **ip igmp ssm-map static** commands in this configuration example.

DNS Server Configuration Example

To configure DNS-based SSM mapping, you need to create a DNS server zone or add records to an existing zone. If the routers that are using DNS-based SSM mapping are also using DNS for other purposes besides SSM mapping, you should use a normally-configured DNS server. If DNS-based SSM mapping is the only DNS implementation being used on the router, you can configure a fake DNS setup with an empty root zone, or a root zone that points back to itself.

The following example shows how to create a zone and import the zone data using Network Registrar:

```
Router> zone 1.1.232.ssm-map.cisco.com. create primary file=named.ssm-map
100 Ok
Router> dns reload
100 Ok
```

The following example shows how to import the zone files from a named.conf file for BIND 8:

```
Router> ::import named.conf /etc/named.conf
Router> dns reload
100 Ok:
```

Note Network Registrar version 8.0 and later support import BIND 8 format definitions.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Standards and RFCs

Standard/RFC	Title
No specific Standards and RFCs are supported by the features in this document.	—

MIBs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

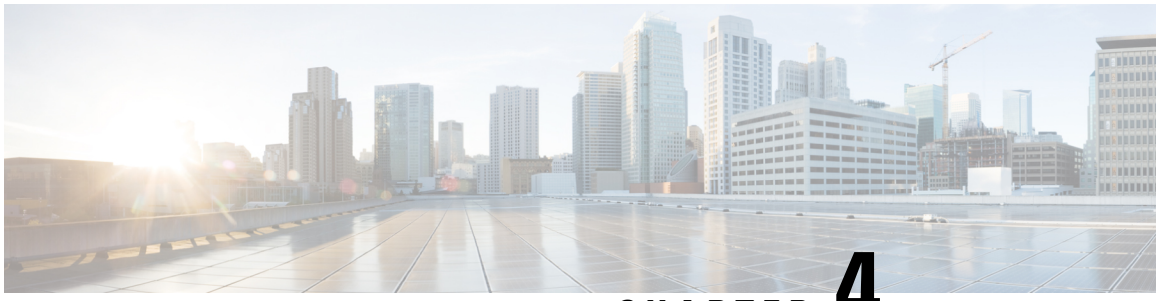
Feature Information for SSM Mapping

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

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

Table 3: Feature Information for SSM Mapping

Feature Name	Releases	Feature Information
SSM Mapping	Cisco IOS XE Release 3.14.0S	This feature was introduced on the Cisco ASR 920 Series Aggregation Services Router (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D, ASR-920-10SZ-PD, ASR-920-24SZ-IM, ASR-920-24SZ-M, ASR-920-24TZ-M) .



CHAPTER 4

Configuring Multicast Admission Control

This module describes how to implement multicast admission control in an IP multicast network. Multicast admission control features are configured on multicast-enabled routers to prevent control plane overload, ensure proper resource allocation, and provide multicast Call Admission Control (CAC) capabilities.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Configuring Multicast Admission Control

IP multicast is enabled and the Protocol Independent Multicast (PIM) interfaces are configured using the tasks described in the “Configuring Basic IP Multicast” module.

Information About Configuring Multicast Admission Control

Multicast Admission Control

As the popularity of network video applications grows among consumers, admission control functions--which govern transmission and reception of multicast traffic based on available network resources--are vital. Without admission control, some users may receive degraded multicast streams, rendering programs unwatchable, and others may receive a “Network Busy” message or nothing at all as network resources are overtaxed. Network admission control is important in maintaining a high quality of experience for digital video consumers.

The goals of multicast admission control features, therefore, are as follows:

- Protect the router from control plane overload to ensure that memory and CPU resources on multicast-enabled routers are not overrun by multicast route (mroute) states or denial-of-service (DoS) attacks from multicast packets.
- Enable proper resource allocation (on a global, per MVRF, or per interface basis) to ensure that multicast services are delivered to subscribers per their IP Service Level Agreements (SLAs) and to minimize the effects of DoS attacks on subscribers.
- Provide multicast CAC capabilities to prevent bandwidth resources (interfaces, subnetworks) from being congested and to enable service providers to offer more flexible and refined content and subscriber-based policies.

Multicast Admission Control Features

The Cisco IOS software supports the following multicast admission control features:

- Global and Per MVRF Mroute State Limit

The **ip multicast route-limit** command allows for the configuration of global and per MVRF state limiters, which impose limits on the number of multicast routes (mroutes) that can be added to the global table or to a particular Multicast Virtual Routing and Forwarding (MVRF) table.

- IGMP State Limit

This feature allows for the configuration of IGMP state limiters, which impose limits on mroute states resulting from Internet Group Management Protocol (IGMP) membership reports (IGMP joins).

- Per Interface Mroute State Limit

This feature allows for the configuration of per interface mroute state limiters, which impose mroute state limits for different access control list (ACL)-classified sets of multicast traffic on an interface.

- Bandwidth-Based CAC for IP Multicast

This feature allows for the configuration of bandwidth-based multicast CAC policies, which allow for bandwidth-based CAC on a per interface basis.

These admission control features may be invoked by service providers and enterprise network administrators based on different criteria, including the service package an end user has purchased or the privileges an enterprise user is entitled to.

Global and Per MVRF Mroute State Limit

The **ip multicast route-limit** command allows for the configuration of global and per MVRF mroute state limiters, which impose limits on the number of mroutes that can be added to the global table or to a particular MVRF table, respectively.

Global mroute state limiters are used to limit the number of mroutes that can be added to the global table on a router. Configuring a global mroute state limiter can protect a router in the event of a multicast DoS attack (by preventing mroutes from overrunning the router).

Per VRF mroute state limiters are used to limit the number of mroutes that can be added to an MVRF table on a Multicast VPN (MVPN) provider edge (PE) router. Configuring per MVRF mroute state limits can be used to ensure the fair sharing of mroutes between different MVRFs on an MVPN PE router.

Global and Per MVRF Mroute State Limit Feature Design

Global and per MVRF mroute state limiters are configured using the **ip multicast route-limit** command in global configuration mode. The syntax of the **ip multicast route-limit** command is as follows:

ip multicast [**vrf** *vrf-name*] **route-limit** *limit* [*threshold*]

Issuing the **ip multicast route-limit**command without the optional **vrf** keyword and *vrf-name* arguments configures a global mroute state limiter. The optional **vrf** keyword and *vrf-name* arguments are used with the **ip multicast limit** command to configure per MVRF mroute state limiters.

Note When configuring global and per VRF mroute state limiters, you can only configure one limit for the global table and one limit per MVRF table.

The value specified for the required *limit* argument defines the maximum number of mroutes that can be added to either the global table or a particular MVRF table, respectively.

Note Global and per MVRF mroute state limiters operate independently and can be used alone or together, depending upon the admission control requirements of your network.

In addition, for both global and per MVRF mroute state limiters, the optional *threshold* argument is available to set mroute threshold limits.

Mechanics of Global and Per MVRF Mroute State Limiters

The mechanics of global and per MVRF mroute state limiters are as follows:

- Each time the state for an mroute is created on a router, the Cisco IOS software checks to see if the limit for the global mroute state limiter (if the mroute is associated with the global table) or the limit for the per MVRF mroute state limiter (if the mroute is associated with the MVRF table) has been reached.
- States for mroutes that exceed the configured limit for the global or the per MVRF mroute state limiter are not created on the router, and a warning message in the following format is generated:

```
% MROUTE-4-ROUTELIMIT : <current mroute count> exceeded multicast route-limit of
<mroute limit value>
```

- When an mroute threshold limit is also configured for the global or the per MVRF mroute state limiter, each time the state for an mroute is created on a router, the Cisco IOS software also checks to see if the mroute threshold limit has been reached. If the mroute threshold limit is exceeded, a warning message in the following format is generated:

```
% MROUTE-4-ROUTELIMITWARNING : multicast route-limit warning <current mroute count> threshold
 <mroute threshold value>
```

Warning messages continue to be generated until the number of mroutes exceeds the configured limit or until the number of mroute states falls below the configured mroute threshold limit.

MSDP SA Limit

The **ip msdp sa-limit** command allows for the configuration of MSDP SA limiters, which impose limits on the number of MSDP Source Active (SA) messages that an MSDP-enabled router can accept (can be cached) from an MSDP peer. This command provides a means to protect an MSDP-enabled router from denial of service (DoS) attacks.

MSDP SA Limit Feature Design

MSDP SA limiters are configured using the **ip msdp sa-limit** command in global configuration mode. The syntax of the **ip msdp sa-limit** command is as follows:

ip msdp [**vrf** *vrf-name*] **sa-limit** {*peer-address* | *peer-name*} *sa-limit*

For the required *peer-address* argument or *peer-name* argument, specify either the MSDP peer address or MSDP peer name of the peer to be limited.

For the required *sa-limit* argument, specify the maximum number of SA messages that can be accepted (cached) from the specified peer. The range is from 1 to 2147483646.

Note In an MVPN environment, the optional **vrf** keyword and *vrf-name* argument are used to specify the MVRF associated with the MSDP peer. When an MVRF is specified, the MSDP SA limiter is applied to the specified MSDP peer associated with the specified MVRF.

Mechanics of MSDP SA Limiters

- When MSDP SA limiters are configured, the router maintains a per-peer count of SA messages stored in the SA cache.
- SA messages that exceed the limit configured for an MSDP peer are ignored.
- If the router receives SA messages in excess of the configured limit from an MSDP peer, a warning in the following format is generated (once a minute) until the cache is cleared:

```
%MSDP-4-SA_LIMIT: SA from peer <peer address or name>, RP <RP address> for <mroute> exceeded
 sa-limit of <configured SA limit for MSDP peer>
```

Tips for Configuring MSDP SA Limiters

- We recommended that you configure MSDP SA limiters for all MSDP peerings on the router.

- An appropriately low MSDP SA limit should be configured on peerings with a stub MSDP region (an MSDP peer that may have some further downstream peers but does not act as a transit for SA messages across the rest of the Internet).
- An appropriately high SA limit should be configured for all MSDP peerings that act as transits for MSDP SA messages across the Internet.

IGMP State Limit

The IGMP State Limit feature allows for the configuration of IGMP state limiters, which impose limits on mroute states resulting from IGMP membership reports (IGMP joins) on a global or per interface basis. Membership reports exceeding the configured limits are not entered into the IGMP cache. This feature can be used to prevent DoS attacks or to provide a multicast CAC mechanism in network environments where all the multicast flows roughly utilize the same amount of bandwidth.

Note IGMP state limiters impose limits on the number of mroute states resulting from IGMP, IGMP v3lite, and URL Rendezvous Directory (URD) membership reports on a global or per interface basis.

IGMP State Limit Feature Design

- Configuring IGMP state limiters in global configuration mode specifies a global limit on the number of IGMP membership reports that can be cached.
- Configuring IGMP state limiters in interface configuration mode specifies a limit on the number of IGMP membership reports on a per interface basis.
- Use ACLs to prevent groups or channels from being counted against the interface limit. A standard or an extended ACL can be specified. A standard ACL can be used to define the (*, G) state to be excluded from the limit on an interface. An extended ACLs can be used to define the (S, G) state to be excluded from the limit on an interface. An extended ACL also can be used to define the (*, G) state to be excluded from the limit on an interface, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list.
- You can only configure one global limit per device and one limit per interface.

Mechanics of IGMP State Limiters

The mechanics of IGMP state limiters are as follows:

- Each time a router receives an IGMP membership report for a particular group or channel, the Cisco IOS software checks to see if either the limit for the global IGMP state limiter or the limit for the per interface IGMP state limiter has been reached.
- If only a global IGMP state limiter has been configured and the limit has not been reached, IGMP membership reports are honored. When the configured limit has been reached, subsequent IGMP membership reports are then ignored (dropped) and a warning message in one of the following formats is generated:
  -
    ```
    %IGMP-6-IGMP_GROUP_LIMIT: IGMP limit exceeded for <group (*, group address)> on
    <interface type number> by host <ip address>
    ```

- ```
  %IGMP-6-IGMP_CHANNEL_LIMIT: IGMP limit exceeded for <channel (source address, group
   address)> on <interface type number> by host <ip address>
  ```

- If only per interface IGMP state limiters are configured, then each limit is only counted against the interface on which it was configured.
- If both a global IGMP state limiter and per interface IGMP state limiters are configured, the limits configured for the per interface IGMP state limiters are still enforced but are constrained by the global limit.

Per Interface Mroute State Limit

The Per Interface Mroute State Limit feature provides the capability to limit the number of mroute states on an interface for different ACL-classified sets of multicast traffic. This feature can be used to prevent DoS attacks or to provide a multicast CAC mechanism when all the multicast flows roughly utilize the same amount of bandwidth.

The Per Interface Mroute State Limit feature essentially is a complete superset of the IGMP State Limit feature (with the exception that it does not support a global limit). The Per Interface Mroute State Limit feature, moreover, is more flexible and powerful (albeit more complex) than the IGMP State Limit feature but is not intended to be a replacement for it because there are applications that suit both features.

The main differences between the Per Interface Mroute State Limit feature and the IGMP State Limit feature are as follows:

- The Per Interface Mroute State Limit feature allows multiple limits to be configured on an interface, whereas the IGMP State Limit feature allows only one limit to be configured on an interface. The Per Interface Mroute State Limit feature, thus, is more flexible than the IGMP State Limit feature in that it allows multiple limits to be configured for different sets of multicast traffic on an interface.
- The Per Interface Mroute State Limit feature can be used to limit both IGMP and PIM joins, whereas the IGMP State Limit feature can only be used to limit IGMP joins. The IGMP State Limit feature, thus, is more limited in application in that it is best suited to be configured on an edge router to limit the number of groups that receivers can join on an outgoing interface. The Per Interface Mroute State Limit feature has a wider application in that it can be configured to limit IGMP joins on an outgoing interface, to limit PIM joins (for Any Source Multicast [ASM] groups or Source Specific Multicast [SSM] channels) on an outgoing interface connected to other routers, to limit sources behind an incoming interface from sending multicast traffic, or to limit sources directly connected to an incoming interface from sending multicast traffic.

Note Although the PIM Interface Mroute State Limit feature allows you to limit both IGMP and PIM joins, it does not provide the capability to limit PIM or IGMP joins separately because it does not take into account whether the state is created as a result of an IGMP or PIM join. As such, the IGMP State Limit feature is more specific in application because it specifically limits IGMP joins.

- The Per Interface Mroute State Limit feature allows you to specify limits according to the direction of traffic; that is, it allows you to specify limits for outgoing interfaces, incoming interfaces, and for incoming interfaces having directly connected multicast sources. The IGMP State Limit feature, however, only can be used to limit outgoing interfaces. The Per Interface State Mroute State Limit feature, thus, is wider
  ```

in scope in that it can be used to limit mroute states for both incoming and outgoing interfaces from both sources and receivers, whereas the IGMP State Limit feature is more narrow in scope in that it can only be used to limit mroute states for receivers on an LAN by limiting the number of IGMP joins on an outgoing interface.

Both the IGMP State Limit and Per Interface Mroute State Limit features provide a rudimentary multicast CAC mechanism that can be used to provision bandwidth utilization on an interface when all multicast flows roughly utilize the same amount of bandwidth. The Bandwidth-Based CAC for IP Multicast feature, however, offers a more flexible and powerful alternative for providing multicast CAC in network environments where IP multicast flows utilize different amounts of bandwidth.

Per Interface Mroute State Limit Feature Design

The Per Interface Mroute State Limit feature is configured using the **ip multicast limit** command in interface configuration mode. An **ip multicast limit** command configured on an interface is called an per interface mroute state limiter. A per interface mroute state limiter is defined by direction, ACL, and maximum number of mroutes. Each per interface mroute state limiter maintains a counter to ensure that the maximum number of mroutes is not exceeded.

The following forms of the **ip multicast limit**command are available to configure per interface mroute state limiters:

- **ip multicast limit** *access-list max-entries*

This command limits mroute state creation for an ACL-classified set of traffic on an interface when the interface is an outgoing (egress) interface, and limits mroute outgoing interface list (olist) membership when the interface is an incoming (ingress) Reverse Path Forwarding (RPF) interface.

This type of per interface mroute state limiter limits mroute state creation--by accounting each time an mroute permitted by the ACL is created or deleted--and limits mroute olist membership--by accounting each time that an mroute olist member permitted by the ACL is added or removed.

Entering this form of the command (that is, with no optional keywords) is equivalent to specifying the **ip multicast limit rpf**and **ip multicast limit out** forms of the command.

- **ip multicast limit connected** *access-list max-entries*

This command limits mroute state creation for an ACL-classified set of multicast traffic on an incoming (RPF) interface that is directly connected to a multicast source by accounting each time that an mroute permitted by the ACL is created or deleted.

- **ip multicast limit out** *access-list max-entries*

This command limits mroute olist membership on an outgoing interface for an ACL-classified set of multicast traffic by accounting each time that an mroute olist member permitted by the ACL is added or removed.

- **ip multicast limit rpf** *access-list max-entries*

This command limits mroute state creation for an ACL-classified set of multicast traffic on an incoming (RPF) interface by accounting each time an mroute permitted by the ACL is created or deleted.

For the required *access-list* argument, specify the ACL that defines the IP multicast traffic to be limited on an interface. A standard or extended ACL can be specified. Standard ACLs can be used to define the (*, G) state to be limited on an interface. Extended ACLs can be used to define the (S, G) state to be limited on an interface. Extended ACLs also can be used to define the (*, G) state to be limited on an interface, by specifying

0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list.

Mechanics of Per Interface Mroute State Limiters

The mechanics of per interface mroute state limiters are as follows:

- Each time the state for an mroute is created or deleted and each time an olist member is added or removed, the software searches for a corresponding per interface mroute state limiter that matches the mroute.
- When an mroute is created or deleted, the software searches for a per interface mroute state limiter configured on the incoming (RPF) interface that matches the mroute to be created or deleted. When an olist member is added or removed, the software searches for a per interface mroute state limiter configured on the outgoing interface that matches the mroute to be added or removed.
- A top-down search is performed using the list of configured per interface mroute state limiters. Only per interface mroute state limiters that match the direction of traffic are considered. The first per interface mroute state limiter that matches is used for limiting (sometimes referred to as accounting). A match is found when the ACL permits the mroute state.
- When a match is found, the counter of the per interface mroute state limiter is updated (increased or decreased). If no per interface mroute state limiter is found that matches an mroute, no accounting is performed for the mroute (because there is no counter to update).
- The amount with which to update the counter is called the cost (sometimes referred to as the cost multiplier). The default cost is 1.

Note A per interface mroute state limiter always allows the deletion of an mroute or the removal of an interface from the olist. In those cases, the respective per interface mroute state limiter decreases the counter by the value of the cost multiplier. In addition, RPF changes to an existing mroute are always allowed (in order to not affect existing traffic). However, a per interface mroute state limiter only allows the creation of an mroute or the addition of an mroute olist member if adding the cost does not exceed the maximum number of mroutes permitted.

Tips for Configuring Per Interface Mroute State Limiters

- To ensure that all mroutes are accounted, you can configure a per interface mroute state limiter whose ACL contains a permit-any statement and set the value of zero (0) for maximum entries. Configuring an mroute state limiter in this manner effectively denies all fall through states, which may be a way to prevent a multicast DoS attack in and out of the interface.
- When creating an ACL, remember that, by default, the end of the ACL contains an implicit deny-any statement for everything if it did not find a match before reaching the end.
- An explicit deny statement for a specific mroute in an ACL can be used to specify the state that will not match the ACLwhich will prevent the ACL from being accounted. If an mroute matches a deny statement, the search immediately continues to the next configured mroute state limiter. Configuring an explicit deny statement in an ACL can be more efficient than forcing the mroute to fall through an ACL by using an implicit deny-any statement at the end of the ACL.

Bandwidth-Based CAC for IP Multicast

The Bandwidth-Based CAC for IP Multicast feature enhances the Per Interface Mroute State Limit feature by implementing a way to count per interface mroute state limiters using cost multipliers (referred to as *bandwidth-based multicast CAC policies*). This feature can be used to provide bandwidth-based multicast CAC on a per interface basis in network environments where the multicast flows utilize different amounts of bandwidth.

Bandwidth-Based CAC for IP Multicast Feature Design

Bandwidth-based multicast CAC policies are configured using the **ip multicast limit cost** command in global configuration mode. The syntax of the **ip multicast limit cost** command is as follows:

ip multicast [**vrf** *vrf-name*] **limit cost** *access-list cost-multiplier*

For the required *access-list* argument, specify the ACL that defines the IP multicast traffic for which to apply a cost. A standard or extended ACL can be specified. Standard ACLs can be used to define the (*, G) state. Extended ACLs can be used to define the (S, G) state. Extended ACLs also can be used to define the (*, G) state, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list.

For the required *cost-multiplier* argument, specify the cost value to be applied to mroutes that match the ACL associated with the bandwidth-based multicast CAC policy. The range is 0 to 2147483647.

Note In an MVPN environment, the optional **vrf** keyword and *vrf-name* argument are used to specify that the cost be applied only to mroutes associated with MVRF specified for the *vrf-name* argument.

Mechanics of the Bandwidth-Based Multicast CAC Policies

The mechanics of bandwidth-based multicast CAC policies are as follows:

- Once an mroute matches an ACL configured for a per interface mroute state limiter, the Cisco IOS software performs a top-down search from the global or per MVRF list of configured bandwidth-based multicast CAC policies to determine if a cost should be applied to the mroute.
- A cost is applied to the first bandwidth-based CAC policy that matches the mroute. A match is found when the ACL applied to the bandwidth-based CAC policy permits the mroute state.
- The counter for the mroute state limiter either adds or subtracts the cost configured for the *cost-multiplier* argument. If no costs are configured or if the mroute does not match any of the configured bandwidth-based CAC polices, the default cost of 1 is used.

Tips for Configuring Bandwidth-Based CAC Policies for IP Multicast

- To ensure that a particular cost applies to all mroutes being limited, you can configure a bandwidth-based CAC policy whose ACL contains a **permit any** statement. Configuring a bandwidth-based CAC policy in this manner effectively ensures that the default cost is not applied to any mroutes being limited.
- Configuring a bandwidth-based CAC policy with a cost of 0 for the *cost-multiplier* argument can be used to skip the accounting of certain mroutes (for example, to prevent Auto-RP groups or a specific multicast channel from being accounted).

- An explicit deny statement for a specific mroute in an ACL can be used to specify the state that will not match the ACL (thus, preventing the ACL from being accounted). If an mroute matches a deny statement, the search immediately continues to the next configured bandwidth-based CAC policy. Configuring an explicit deny statement in an ACL can be more efficient than forcing the mroute to fall through an ACL (by means of the implicit **deny any** statement at the end of the ACL).

How to Configure Multicast Admission Control

Configuring Global and Per MVRF Mroute State Limiters

Perform the following optional tasks to configure global and per MVRF mroute state limiters.

Global mroute state limiters are used to limit the number of mroutes that can be added to the global table on a router. Configuring a global mroute state limiter can protect a router in the event of a multicast DoS attack (by preventing mroutes from overrunning the router).

Per VRF mroute state limiters are used to limit the number of mroutes that can be added to an MVRF table on an MVPN PE router. Configuring per MVRF mroute state limits can be used to ensure the fair sharing of mroutes between different MVRFs on an MVPN PE router.

Note Global and per MVRF mroute state limiters operate independently and can be used alone or together, depending upon the admission control requirements of your network.

Note When configuring global and per VRF mroute state limiters, you can only configure one limit for the global table and one limit per MVRF table.

The following tasks explain how to configure global and per MVRF mroute state limiters:

Prerequisites

- These tasks assume that IP multicast has been enabled and that the PIM interfaces have been configured using the tasks described in the " Configuring Basic IP Multicast " module.
- Before configuring per MVRF mroute state limiters, the MVRFs on the PE router must be configured using the tasks described in the " Configuring Multicast VPN " module.

Configuring a Global Mroute State Limiter

Perform this task to limit the number of mroutes that can be added to the global table. States for mroutes that exceed the global mroute limit will not be created.

SUMMARY STEPS

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

3. **ip multicast route-limit** *limit* [*threshold*]
4. **end**
5. **show ip mroute count**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast route-limit** *limit* [*threshold*] **Example:** `Router(config)# ip multicast route-limit 1500 1460`	Limits the number of mroutes that can be added to the global table. • For the required *limit* argument, specify the limit on the number of mroutes that can be added to the global table. The range is from 1 to 2147483647. • Use the optional *threshold*argument to set an mroute threshold limit. The range is from 1 to 2147483647. • Maximum number of mroute state limits supported globally is 1000.
Step 4	**end** **Example:** `Router(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 5	**show ip mroute count** **Example:** `Router# show ip mroute count`	(Optional) Displays mroute data and packet count statistics. • Use this command to verify the number of mroutes in the global table.

What to Do Next

Proceed to the Configuring Per MVRF Mroute State Limiters, on page 57 task to configure per MVRF mroute state limiters on a PE router.

Configuring Per MVRF Mroute State Limiters

Perform this optional task to configure per MVRF mroute state limiters to limit the number of mroutes that can be added to a particular MVRF table. This feature can be configured on a PE router to ensure the fair

sharing of mroutes between different MVRFs on the router. States for mroutes that exceed the per MVRF mroute limiter are not created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast vrf** *vrf-name* **route-limit** *limit* [*threshold*]
4. Repeat Step 3 to configure additional per VRF mroute state limiters for other VRFs on an MVPN PE router.
5. **end**
6. **show ip mroute vrf** *vrf-name* **count**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast vrf** *vrf-name* **route-limit** *limit* [*threshold*] **Example:** `Router(config)# ip multicast vrf red route-limit 1500 1460`	Limits the number of mroutes that can be added to a particular MVRF table. • For the **vrf** keyword and *vrf-name* argument, specify the MVRF for which to apply the limit. • For the required *limit* argument, specify the limit on the number of mroutes that can be added to the MVRF table (for the specified MVRF). The range is from 1 to 2147483647. • Use the optional *threshold*argument to set an mroute threshold limit. The range is from 1 to 2147483647 • Maximum number of mroute state limits supported on MVRF level is 1000.
Step 4	Repeat Step 3 to configure additional per VRF mroute state limiters for other VRFs on an MVPN PE router.	--
Step 5	**end** **Example:** `Router(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show ip mroute vrf** *vrf-name* **count** **Example:** `Router# show ip mroute vrf red count`	(Optional) Displays mroute data and packet count statistics related to the specified MVRF. • Use this command to verify the number of mroutes in a particular MVRF table.

Configuring MSDP SA Limiters

Perform this optional task to limit the overall number of SA messages that the router can accept from specified MSDP peers. Performing this task protects an MSDP-enabled router from distributed DoS attacks.

Note We recommend that you perform this task for all MSDP peerings on the router.

Before you begin

This task assumes that you are running MSDP and have configured MSDP peers using the tasks described in the " Using MSDP to Interconnect Multiple PIM-SM Domains " module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip msdp** [**vrf** *vrf-name*] **sa-limit** {*peer-address* | *peer-name*} *sa-limit*
4. Repeat Step 3 to configure additional per MVRF mroute state limiters for other MVRFs on an MVPN PE router.
5. **end**
6. **show ip msdp count**
7. **show ip msdp peer** [*peer-address* | *peer-name*]
8. **show ip msdp summary**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip msdp** [**vrf** *vrf-name*] **sa-limit** {*peer-address* | *peer-name*} *sa-limit* **Example:** `Router(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. • Use the optional **vrf** keyword and *vrf-name* argument to specify the MVRF associated with the MSDP peer. When an MVRF is specified, the MSDP SA limiter is applied to the specified MSDP peer associated with the specified MVRF. • For the required *peer-address* argument or *peer-name* argument, specify either the MSDP peer address or MSDP peer name of the peer to be limited. • For the required *sa-limit* argument, specify the maximum number of SA messages that can be accepted (cached) from the specified peer. The range is from 1 to 2147483646.
Step 4	Repeat Step 3 to configure additional per MVRF mroute state limiters for other MVRFs on an MVPN PE router.	--
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip msdp count** **Example:** `Router# show ip msdp count`	(Optional) Displays the number of sources and groups originated in MSDP SA messages and the number of SA messages from an MSDP peer in the SA cache.
Step 7	**show ip msdp peer** [*peer-address* | *peer-name*] **Example:** `Router# show ip msdp peer`	(Optional) Displays detailed information about MSDP peers. **Note** The output of this command displays the number of SA messages received from MSDP peers that are stored in the cache.
Step 8	**show ip msdp summary** **Example:** `Router# show ip msdp summary`	(Optional) Displays MSDP peer status. **Note** The output of this command displays a per-peer "SA Count" field that displays the number of SAs stored in the SA cache.

Configuring IGMP State Limiters

Note IGMP state limiters impose limits on the number of mroute states resulting from IGMP, IGMP v3lite, and URD membership reports on a global or per interface basis.

Prerequisites

- These tasks assume that IP multicast has been enabled and that the PIM interfaces have been configured using the tasks described in the " Configuring Basic IP Multicast " module.
- All ACLs you intend to apply to per interface IGMP state limiters should be configured prior to beginning this configuration task; otherwise, IGMP membership reports for all groups and channels are counted against the configured limits. For information about how to configure ACLs, see the " Creating an IP Access List and Applying It to an Interface " module.

Configuring Global IGMP State Limiters

Perform this optional task to configure one global IGMP state limiter per device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp limit** *number*
4. **end**
5. **show ip igmp groups**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp limit** *number* **Example:** `Device(config)# ip igmp limit 150`	Configures a global limit on the number of mroute states resulting from IGMP membership reports (IGMP joins).
Step 4	**end** **Example:** `Device(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 5	**show ip igmp groups** **Example:** `Device# show ip igmp groups`	(Optional) Displays the multicast groups with receivers that are directly connected to the device and that were learned through IGMP.

What to Do Next

Proceed to the Configuring Per Interface IGMP State Limiters, on page 62 task to configure per interface IGMP state limiters.

Configuring Per Interface IGMP State Limiters

Perform this optional task to configure a per interface IGMP state limiter.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip igmp limit** *number* [**except** *access-list*]
5. Do one of the following:
   - **exit**
   - **end**
6. **show ip igmp interface** [*type number*]
7. **show ip igmp groups**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0`	Enters interface configuration mode. • Specify an interface that is connected to hosts.
Step 4	**ip igmp limit** *number* [**except** *access-list*] **Example:** `Device(config-if)# ip igmp limit 100`	Configures a per interface limit on the number of mroutes states created as a result of IGMP membership reports (IGMP joins).
Step 5	Do one of the following: • **exit** • **end**	• (Optional) Ends the current configuration session and returns to global configuration mode. Repeat steps 3 and 4 to configure a per interface limiter on another interface.

	Command or Action	Purpose

	Example:
	`Device(config-if)# exit`
	`Device(config-if)# end`
		• Ends the current configuration session and returns to privileged EXEC mode.

Step 6	**show ip igmp interface** [*type number*] **Example:** `Device# show ip igmp interface`	(Optional) Displays information about the status and configuration of IGMP and multicast routing on interfaces.
Step 7	**show ip igmp groups** **Example:** `Device# show ip igmp groups`	(Optional) Displays the multicast groups with receivers that are directly connected to the device and that were learned through IGMP.

Configuring Per Interface Mroute State Limiters

Perform this task to prevent DoS attacks or to provide a multicast CAC mechanism for controling bandwidth when all multicast flows utilize approximately the same amount of bandwidth.

Before you begin

All ACLs to be applied to per interface mroute state limiters must be configured prior to beginning this configuration task; otherwise, the limiters are ignored. For information about how to configure ACLs, see the " Creating an IP Access List and Applying It to an Interface " module of the *Security Configuration Guide: Access Control Lists* guide.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip multicast limit** [**connected** | **out** | **rpf**] *access-list max-entries*
5. Repeat Step 4 to configure additional per interface mroute state limiters on this interface.
6. Repeat Steps 3 and Step 4 to configure per interface mroute state limiters on additional interfaces.
7. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0`	Enters interface configuration mode for the specified interface type and number.
Step 4	**ip multicast limit** [**connected** | **out** | **rpf**] *access-list max-entries* **Example:** `Device(config-if)# ip multicast limit 15 100`	Configures per interface mroute state limiters.
Step 5	Repeat Step 4 to configure additional per interface mroute state limiters on this interface.	--
Step 6	Repeat Steps 3 and Step 4 to configure per interface mroute state limiters on additional interfaces.	--
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

What to Do Next

Proceed to the Monitoring Per Interface Mroute State Limiters and Bandwidth-Based Multicast CAC Policies task to monitor per interface mroute state limiters.

Configuring Bandwidth-Based Multicast CAC Policies

Perform this optional task to configure bandwidth-based multicast CAC policies. Bandwidth-based multicast CAC policies provide the capability to assign costs to mroutes that are being limited by per interface mroute state limiters. This task can be used to provide bandwidth-based multicast CAC on a per interface basis in network environments where the multicast flows utilize different amounts of bandwidth. Bandwidth-based multicast CAC policies can be applied globally or per MVRF.

Before you begin

- This task assumes that IP multicast has been enabled and that the PIM interfaces have been configured using the tasks described in the " Configuring Basic IP Multicast " module.
- All ACLs you intend to apply to bandwidth-based multicast CAC policies should be configured prior to beginning this configuration task; otherwise, the limiters are ignored. For information about how to configure ACLs, see the " Creating an IP Access List and Applying It to an Interface " module.

Note You can omit Steps 3 to 7 if you have already configured the per interface mroute state limiters for which to apply costs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip multicast limit** [**connected** | **out** | **rpf**] *access-list max-entries*
5.
6.
7. Repeat Step 4 if you want to configure additional mroute state limiters on the interface.
8. Repeat Step 3 and Step 4 if you want to configure mroute state limiters on additional interfaces.
9. **exit**
10. **ip multicast** [**vrf** *vrf-name*] **limit cost** *access-list cost-multiplier*
11. Repeat Step 8 if you want to apply additional costs to mroutes.
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0`	Enters interface configuration mode for the specified interface type and number.
Step 4	**ip multicast limit** [**connected**	**out**	**rpf**] *access-list max-entries* **Example:** `Router(config-if)# ip multicast limit acl-test 100` **Example:**	Configures mroute state limiters on an interface. • Specify the **ip multicast limit** command with no optional keywords to limit mroute state creation for an ACL-classified set of traffic on an interface when the interface is an outgoing (egress) interface, and to limit mroute olist membership when the interface is an incoming (ingress) Reverse Path Forwarding (RPF) interface.

	Command or Action	Purpose
Step 5		• • This type of mroute state limiter limits mroute state creation by accounting each time an mroute permitted by the ACL is created or deleted and limits mroute olist membership by accounting each time that an mroute olist member permitted by the ACL is added or removed. • Entering this form of the command (that is, with no optional keywords) is equivalent to specifying the **ip multicast limit rpf**and **ip multicast limit out** forms of the command.
Step 6		• Use the optional **connected** keyword to configure an mroute state limiter that limits mroute states created for an ACL-classified set of multicast traffic on an incoming (RPF) interface that is directly connected to a multicast source by accounting each time that an mroute permitted by the ACL is created or deleted. • Use the optional **out** keyword to configure an mroute state limiter that limits mroute olist membership on an outgoing interface for an ACL-classified set of multicast traffic by accounting each time that an mroute olist member permitted by the ACL is added or removed. • Use the optional **rpf** keyword to configure an mroute state limiter that limits the number of mroute states created for an ACL-classified set of multicast traffic on an incoming (RPF) interface by accounting each time an mroute permitted by the ACL is created or deleted. • For the required *access-list* argument, specify the ACL that defines the IP multicast traffic to be limited on an interface. • Standard ACLs can be used to define the (*, G) state to be limited on an interface. • Extended ACLs can be used to define the (S, G) state to be limited on an interface. Extended ACLs also can be used to define the (*, G) state to be limited on an interface, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list. • For the required *max-entries* argument, specify the maximum number of mroutes permitted by the per interface mroute state limiter. The range is from 0 to 2147483647.

	Command or Action	Purpose
Step 7	Repeat Step 4 if you want to configure additional mroute state limiters on the interface.	--
Step 8	Repeat Step 3 and Step 4 if you want to configure mroute state limiters on additional interfaces.	--
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode, and returns to global configuration mode.

Step 10	**ip multicast** [**vrf** *vrf-name*] **limit cost** *access-list cost-multiplier*
	Example:
	`Router(config)# ip multicast limit cost acl-MP2SD-channels 4000`
		Applies costs to per interface mroute state limiters.
		• Use the optional **vrf** keyword and *vrf-name*argument to specify that the cost be applied only to mroutes associated with MVRF specified for the *vrf-name* argument.
		• For the required *access-list* argument, specify the ACL that defines the IP multicast traffic for which to apply a cost.
		  • Standard ACLs can be used to define the (*, G) state.
		  • Extended ACLs can be used to define the (S, G) state. Extended ACLs also can be used to define the (*, G) state, by specifying 0.0.0.0 for the source address and source wildcard--referred to as (0, G)--in the permit or deny statements that compose the extended access list.
		• For the required *cost-multiplier* argument, specify the cost value to be applied to mroutes that match the ACL associated with the bandwidth-based multicast CAC policy. The range is 0 to 2147483647.

Step 11	Repeat Step 8 if you want to apply additional costs to mroutes.	--
Step 12	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode, and enters privileged EXEC mode.

What to Do Next

Proceed to the Monitoring Per Interface Mroute State Limiters and Bandwidth-Based Multicast CAC Policies task to monitor bandwidth-based multicast CAC policies.

Monitoring Per Interface Mroute State Limiters and Bandwidth-Based Multicast CAC Policies

Perform this optional task to monitor per interface mroute state limiters and bandwidth-based multicast CAC policies.

SUMMARY STEPS

1. **enable**
2. **debug ip mrouting limits** [*group-address*]
3. **show ip multicast limit** *type number*
4. **clear ip multicast limit** [*type number*]

DETAILED STEPS

Step 1 enable

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 debug ip mrouting limits [*group-address*]

Displays debugging information about configured per interface mroute state limiters and bandwidth-based multicast CAC policies.

The following output is from the **debug ip mrouting limits** command. The output displays the following events:

- An mroute state being created and the corresponding per interface mroute state limiter counter being increased by the default cost of 1 on incoming Ethernet interface 1/0.
- An mroute olist member being removed from the olist and the corresponding per interface mroute limiter being decreased by the default cost of 1 on outgoing Ethernet interface 1/0.
- An mroute being denied by the per interface mroute state limiter because the maximum number of mroute states has been reached.
- An mroute state being created and the corresponding per interface mroute state limiter counter being increased by the cost of 2 on incoming Ethernet interface 1/0.
- An mroute olist member being removed from the olist and the corresponding per interface mroute limiter being decreased by a cost of 2 on outgoing Ethernet interface 1/0.

Example:

```
device# debug ip mrouting limits

MRL(0): incr-ed acl 'rpf-list' to (13 < max 32), [n:0,p:0], (main) GigabitEthernet0/0, (10.41.0.41,
 225.30.200.60)
MRL(0): decr-ed acl 'out-list' to (10 < max 32), [n:0,p:0], (main) GigabitEthernet0/0, (*,
```

```
225.40.202.60)
MRL(0): Add mroute (10.43.0.43, 225.30.200.60) denied for GigabitEthernet0/2, acl std-list, (16 =
max 16)
MRL(0): incr-ed limit-acl `rpf-list' to (12 < max 32), cost-acl 'cost-list' cost 2, [n:0,p:0], (main)
 GigabitEthernet0/0, (10.41.0.41, 225.30.200.60)
MRL(0): decr-ed limit-acl `out-list' to (8 < max 32), cost-acl 'cost-list'' cost 2, [n:0,p:0], (main)
 GigabitEthernet0/0, (*, 225.40.202.60)
```

Step 3 **show ip multicast limit** *type number*

Displays counters related to mroute state limiters configured on the interfaces on the router.

For each per interface mroute state limiter shown in the output, the following information is displayed:

- The direction of traffic that the per mroute state limiter is limiting.
- The ACL referenced by the per interface mroute state limiter that defines the IP multicast traffic being limited.
- Statistics, enclosed in parenthesis, which track the current number of mroutes being limited less the configured limit. Each time the state for an mroute is created or deleted and each time an outgoing interface list (olist) member is added or removed, the counters for matching per interface mroute state limiters are increased or decreased accordingly.
- The exceeded counter, which tracks the total number of times that the limit configured for the per interface mroute state limiter has been exceeded. Each time an mroute is denied due to the configured limit being reached, the exceeded counter is increased by a value of 1.

The following is sample output from the **show ip multicast limit** command with the *type number* arguments. In this example, information about mroute state limiters configured on Gigabit Ethernet interface 0/0 is displayed.

Example:

```
Device# show ip multicast limit GigabitEthernet 0/0

Interface GigabitEthernet 0/0
 Multicast Access Limits
out acl out-list (1 < max 32) exceeded 0
rpf acl rpf-list (6 < max 32) exceeded 0
con acl conn-list (0 < max 32) exceeded 0
```

Step 4 **clear ip multicast limit** [*type number*]

Resets the exceeded counter for per interface mroute state limiters.

The following example shows how to reset exceeded counters for per interface mroute state limiters configured on Gigabit Ethernet interface 0/0:

Example:

```
Device# clear ip multicast limit interface GigabitEthernet 0/0
```

Configuration Examples for Configuring Multicast Admission Control

Configuring Global and Per MVRF Mroute State Limiters Example

The following example shows how to configure a global mroute state limiter. In this example, a global mroute state limiter is configured with an mroute limit of 1500 and an mroute threshold limit of 1460.

```
ip multicast route-limit 1500 1460
```

The following is a sample mroute threshold warning message. The output shows that the configured mroute threshold limit of 1460 has been exceeded by one mroute.

```
%MROUTE-4-ROUTELIMITWARNING : multicast route-limit warning 1461 threshold 1460
```

The following is a sample mroute exceeded warning message. The output shows that the configured mroute limit of 1500 has been exceeded by one mroute. States for mroutes that exceed the configured limit for the global mroute state limiter are not created on the router.

```
%MROUTE-4-ROUTELIMIT : 1501 routes exceeded multicast route-limit of 1500
```

Configuring MSDP SA Limiters Example

The following example shows how to configure an MSDP SA limiter. In this example, an MSDP SA limiter is configured that imposes a limit of 100 SA messages from the MSDP peer at 192.168.10.1.

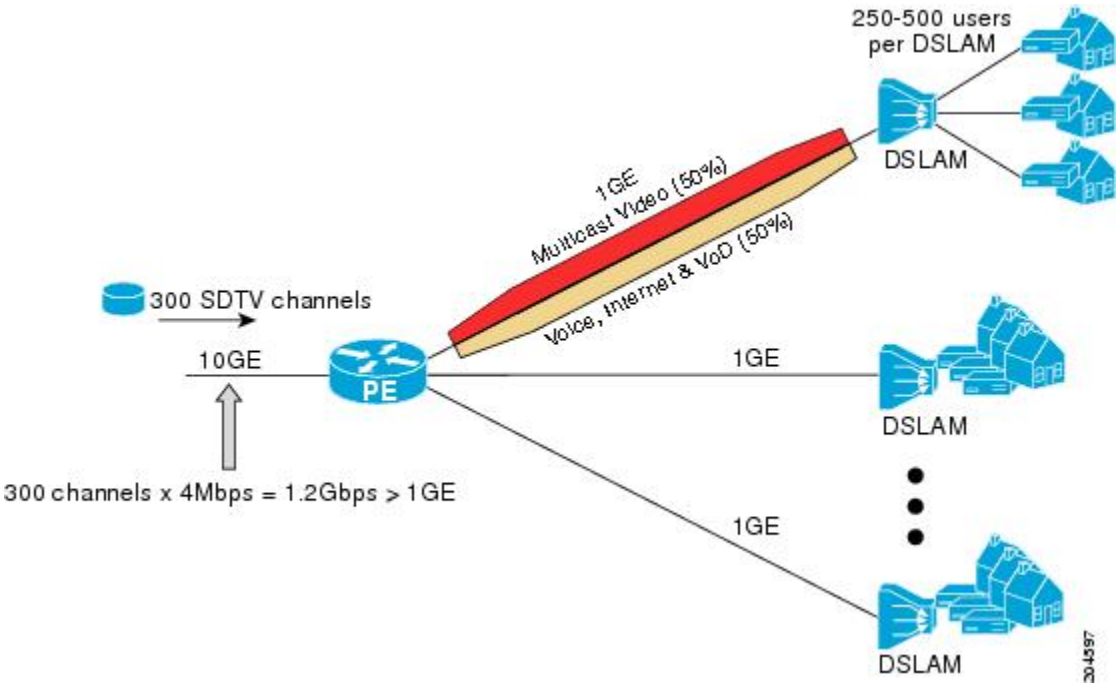
```
ip msdp sa-limit 192.168.10.1 100
```

Example: Configuring IGMP State Limiters

The following example shows how to configure IGMP state limiters to provide multicast CAC in a network environment where all the multicast flows roughly utilize the same amount of bandwidth.

This example uses the topology illustrated in the figure.

Figure 3: IGMP State Limit Example Topology



In this example, a service provider is offering 300 Standard Definition (SD) TV channels. Each SD channel utilizes approximately 4 Mbps.

The service provider must provision the Gigabit Ethernet interfaces on the PE device connected to the Digital Subscriber Line Access Multiplexers (DSLAMs) as follows: 50% of the link's bandwidth (500 Mbps) must be available to subscribers of the Internet, voice, and video on demand (VoD) service offerings while the remaining 50% (500 Mbps) of the link's bandwidth must be available to subscribers of the SD channel offerings.

Because each SD channel utilizes the same amount of bandwidth (4 Mbps), per interface IGMP state limiters can be used to provide the necessary CAC to provision the services being offered by the service provider. To determine the required CAC needed per interface, the total number of channels is divided by 4 (because each channel utilizes 4 Mbps of bandwidth). The required CAC needed per interface, therefore, is as follows:

500Mbps / 4Mbps = 125 mroutes

Once the required CAC is determined, the service provider uses the results to configure the per IGMP state limiters required to provision the Gigabit Ethernet interfaces on the PE device. Based on the network's CAC requirements, the service provider must limit the SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 125. Configuring a per interface IGMP state limit of 125 for the SD channels provisions the interface for 500 Mbps of bandwidth, the 50% of the link's bandwidth that must always be available (but never exceeded) for the SD channel offerings.

The following configuration shows how the service provider uses a per interface mroute state limiter to provision interface Gigabit Ethernet 0/0 for the SD channels and Internet, Voice, and VoD services being offered to subscribers:

```
interface GigabitEthernet0/0
description --- Interface towards the DSLAM ---
.
.
```

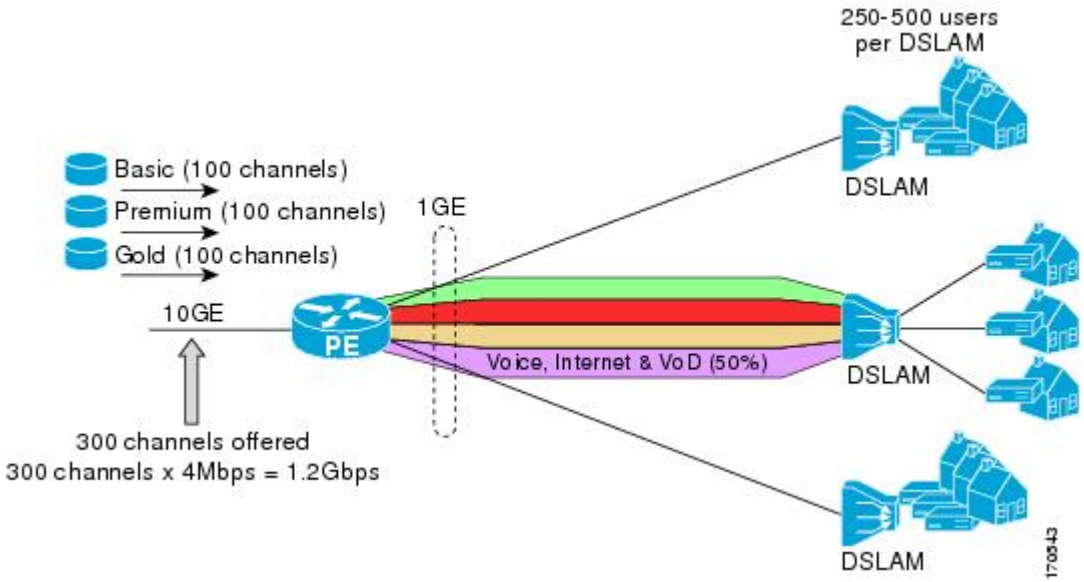
```
.
ip igmp limit 125
```

Example Configuring Per Interface Mroute State Limiters

The following example shows how to configure per interface mroute state limiters to provide multicast CAC in a network environment where all the multicast flows roughly utilize the same amount of bandwidth.

This example uses the topology illustrated in the figure.

Figure 4: Per Interface Mroute State Limit Example Topology



In this example, a service provider is offering 300 SD TV channels. The SD channels are being offered to customers in three service bundles (Basic, Premium, and Gold), which are available to customers on a subscription basis. Each bundle offers 100 channels to subscribers, and each channel utilizes approximately 4 Mbps of bandwidth.

The service provider must provision the Gigabit Ethernet interfaces on the PE device connected to DSLAMs as follows: 50% of the link's bandwidth (500 Mbps) must be available to subscribers of their Internet, voice, and VoD service offerings while the remaining 50% (500 Mbps) of the link's bandwidth must be available to subscribers of their SD channel bundle service offerings.

For the 500 Mbps of the link's bandwidth that must always be available to (but must never be exceeded by) the subscribers of the SD channel bundles, the interface must be further provisioned as follows:

- 60% of the bandwidth must be available to subscribers of the basic service (300 Mbps).
- 20% of the bandwidth must be available to subscribers of the premium service (100 Mbps).
- 20% of the bandwidth must be available to subscribers of the gold service (100 Mbps).

Because each SD channel utilizes the same amount of bandwidth (4 Mbps), per interface mroute state limiters can be used to provide the necessary CAC to provision the services being offered by the service provider. To determine the required CAC needed per interface, the number of channels for each bundle is divided by 4 (because each channel utilizes 4 Mbps of bandwidth). The required CAC needed per interface, therefore, is as follows:

- Basic Services: 300 / 4 = 75
- Premium Services: 100 / 4 = 25
- Gold Services: 100 / 4 = 25

Once the required CAC required per SD channel bundle is determined, the service provider uses the results to configure the mroute state limiters required to provision the Gigabit Ethernet interfaces on the PE device for the services being offered to subscribers behind the DSLAMs:

- For the Basic Services bundle, the service provider must limit the number of Basic Service SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 75. Configuring an mroute state limit of 75 for the SD channels offered in the Basic Service bundle provisions the interface for 300 Mbps of bandwidth (the 60% of the link's bandwidth that must always be available to [but never exceeded by] the subscribers of the Basic Services bundle).
- For the Premium Services bundle, the service provider must limit the number of Premium Service SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 25. Configuring an mroute state limit of 25 for the SD channels offered in the Premium Service bundle provisions the interface for 100 Mbps of bandwidth (the 20% of the link's bandwidth that must always be available to [but never exceeded by] the subscribers of the Premium Service bundle).
- For the Gold Services bundle, the service provider must limit the number of Gold Service SD channels that can be transmitted out a Gigabit Ethernet interface (at any given time) to 25. Configuring an mroute state limit of 25 for the SD channels offered in the Gold Service bundle provisions the interface for 100 Mbps of bandwidth (the 20% of the link's bandwidth that must always be available to [but never exceeded by] the subscribers of the Gold Service bundle).

The service provider then configures three ACLs to be applied to per interface mroute state limiters. Each ACL defines the SD channels for each SD channel bundle to be limited on an interface:

- acl-basic--The ACL that defines the SD channels offered in the basic service.
- acl-premium--The ACL that defines the SD channels offered in the premium service.
- acl-gold--The ACL that defines the SD channels offered in the gold service.

These ACLs are then applied to per interface mroute state limiters configured on the PE device's Gigabit Ethernet interfaces.

For this example, three per interface mroute state limiters are configured on Gigabit Ethernet interface 0/0 to provide the multicast CAC needed to provision the interface for the SD channel bundles being offered to subscribers:

- An mroute state limit of 75 for the SD channels that match acl-basic.
- An mroute state limit of 25 for the SD channels that match acl-premium.
- An mroute state limit of 25 for the SD channels that match acl-gold.

The following configuration shows how the service provider uses per interface mroute state limiters to provision Gigabit Ethernet interface 0/0 for the SD channel bundles and Internet, Voice, and VoD services being offered to subscribers:

```
interface GigabitEthernet0/0
description --- Interface towards the DSLAM ---
.
```

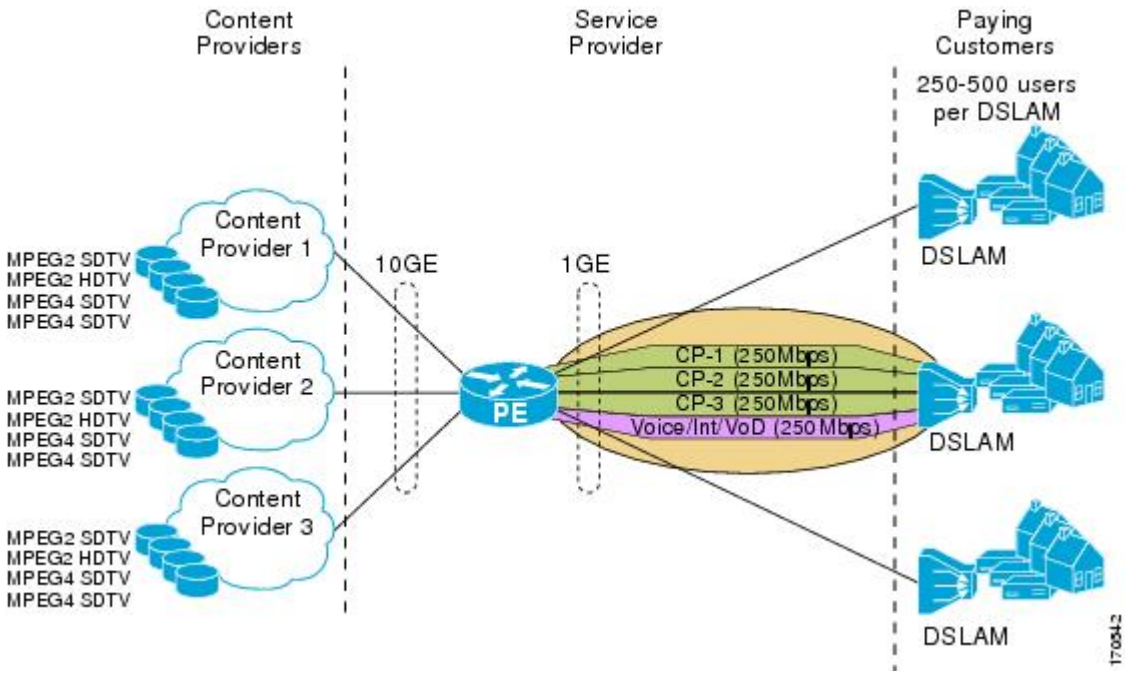
```
.
.
ip multicast limit out acl-basic 75
ip multicast limit out acl-premium 25
ip multicast limit out acl-gold 25
```

Example: Configuring Bandwidth-Based Multicast CAC Policies

The following example shows how to configure bandwidth-based multicast CAC policies to provide multicast CAC in a network environment where the multicast flows utilize different amounts of bandwidth.

This example uses the topology illustrated in the figure.

Figure 5: Bandwidth-Based CAC for IP Multicast Example Topology



In this example, three content providers are providing TV services across a service provider core. The content providers are broadcasting TV channels that utilize different amounts of bandwidth:

- MPEG-2 SDTV channels--4 Mbps per channel.
- MPEG-2 HDTV channels--18 Mbps per channel.
- MPEG-4 SDTV channels--1.6 Mbps per channel.
- MPEG-4 HDTV channels--6 Mbps per channel.

The service provider needs to provision the fair sharing of bandwidth between these three content providers to its subscribers across Gigabit Ethernet interfaces. The service provider, thus, determines that it needs to provision each Gigabit Ethernet interface on the PE device connected to the DSLAMs as follows:

- 250 Mbps per content provider.
- 250 Mbps for Internet, voice, and VoD services.

The service provider then configures three ACLs:

- acl-CP1-channels--The ACL that defines the channels being offered by the content provider CP1.
- acl-CP2-channels--The ACL that defines the channels being offered by the content provider CP2.
- acl-CP3-channels--The ACL that defines the channels being offered by the content provider CP3.

Because the content providers are broadcasting TV channels that utilize different amounts of bandwidth, the service provider needs to determine the values that need to be configured for the per interface mroute state limiters and bandwidth-based multicast CAC policies to provide the fair sharing of bandwidth required between the content providers.

Prior to the introduction of the Bandwidth-Based CAC for IP Multicast feature, per interface mroute state limiters were based strictly on the number of flows. The introduction of cost multipliers by the Bandwidth-Based CAC for IP Multicast feature expands how per interface mroute state limiters can be defined. Instead of defining the per interface mroute state limiters based on the number of multicast flows, the service provider looks for a common unit of measure and decides to represent the per interface mroute state limiters in kilobits per second (Kbps). The service provider then configures three per interface mroute state limiters, one per content provider. Because the link is a Gigabit, the service provider sets each limit to 250000 (because 250000 Kbps equals 250 Mbps, the number of bits that service provider needs to provision per content provider).

The service provider needs to further provision the fair sharing of bandwidth between the content providers, which can be achieved by configuring bandwidth-based multicast CAC policies. The service provider decides to create four bandwidth-based CAC policies, one policy per channel based on bandwidth. For these policies, the service provider configures the following ACLs:

- acl-MP2SD-channels--Defines all the MPEG-2 SD channels offered by the three content providers.
- acl-MP2HD-channels--Defines all the MPEG-2 HD channels offered by the three content providers.
- acl-MP4SD-channels--Defines all the MPEG-4 SD channels offered by the three content providers.
- acl-MP4HD-channels--Defines all the MPEG-4 HD channels offered by the three content providers.

For each policy, a cost multiplier (represented in Kbps) is defined for each ACL that is based on the bandwidth of the channels defined in the ACL:

- 4000--Represents the 4 Mbps MPEG-2 SD channels.
- 18000--Represents the 18 Mbps MPEG-2 HD channels.
- 1600--Represents the 1.6 Mbps MPEG-4 SD channels.
- 6000--Represents the 6 Mbps MPEG-4 HD channels.

The following configuration example shows how the service provider used per interface mroute state limiters with bandwidth-based multicast CAC policies to provision Gigabit Ethernet interface 0/0 for the fair sharing of bandwidth required between the three content providers:

```
!
ip multicast limit cost acl-MP2SD-channels 4000
ip multicast limit cost acl-MP2HD-channels 18000
ip multicast limit cost acl-MP4SD-channels 1600
ip multicast limit cost acl-MP4HD-channels 6000
!
.
.
```

```
.
!
interface GigabitEthernet0/0
 ip multicast limit out acl-CP1-channels 250000
 ip multicast limit out acl-CP2-channels 250000
 ip multicast limit out acl-CP3-channels 250000
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mcl/allreleasemcl/all-book.html

Standards and RFCs

Standard/RFC	Title
No specific Standards and RFCs are supported by the features in this document.	—

MIBs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

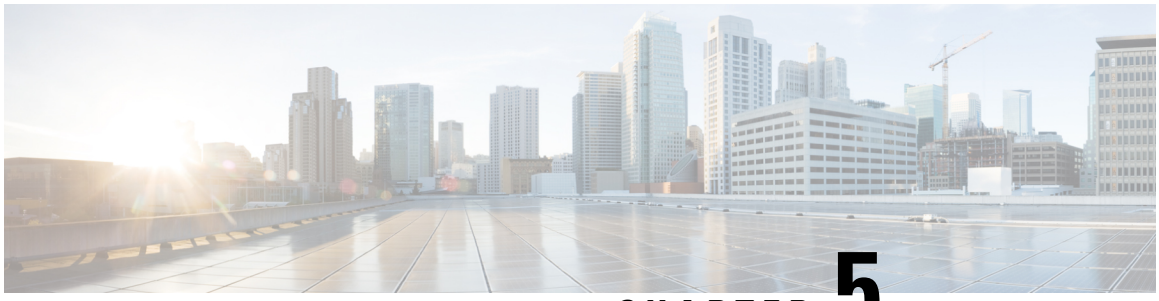
Feature Information for Multicast Admission Control

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

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

Table 4: Feature Information for Multicast Admission Control

Feature Name	Releases	Feature Information
Multicast Admission Control	Cisco IOS XE Release 3.14.0S	This feature was introduced on the Cisco ASR 920 Series Aggregation Services Router (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D, ASR-920-10SZ-PD, ASR-920-24SZ-IM, ASR-920-24SZ-M, ASR-920-24TZ-M) .



CHAPTER 5

Configuring Multicast VPN

The Multicast VPN (MVPN) feature provides the ability to support multicast over a Layer 3 VPN. As enterprises extend the reach of their multicast applications, service providers can accommodate these enterprises over their Multiprotocol Label Switching (MPLS) core network. IP multicast is used to stream video, voice, and data to an MPLS VPN network core.

Historically, point-to-point tunnels were the only way to connect through a service provider network. Although such tunneled networks tend to have scalability issues, they represented the only means of passing IP multicast traffic through a VPN.

Because Layer 3 VPNs support only unicast traffic connectivity, deploying MPLS in conjunction with a Layer 3 VPN allows service providers to offer both unicast and multicast connectivity to Layer 3 VPN customers.

Prerequisites for Configuring Multicast VPN

Enable IP multicast and configure the PIM interfaces using the tasks described in the “Configuring Basic IP Multicast” module.

Restrictions for Configuring Multicast VPN

- The update source interface for the Border Gateway Protocol (BGP) peerings must be the same for all BGP peerings configured on the router in order for the default multicast distribution tree (MDT) to be configured properly. If you use a loopback address for BGP peering, then PIM sparse mode must be enabled on the loopback address.
- MVPN does not support multiple BGP peering update sources.
- Multiple BGP update sources are not supported and configuring them can break MVPN reverse path forwarding (RPF) checking. The source IP address of the MVPN tunnels is determined by the highest IP address used for the BGP peering update source. If this IP address is not the IP address used as the BGP peering address with the remote provider edge (PE) router, MVPN will not function properly.

- PIM Dense mode is *not* supported on core network.
- Extra traffic is noticed when the router acts as MVPN PE without any receivers attached. It is recommended to create an ACL and attach it to VRF to drop extra forwarded packets.
- A maximum of 20 multicast VRFs are supported.
- A maximum of 255 OIFs are supported.
- MVPN GRE is supported only in video template.
- Tx and Rx SPAN are not supported on PE routers configured with MVPN GRE.
- You can only use single encapsulation per core/WAN interface.
- Effective Cisco IOS XE 16.5.1, MVPN GRE with BDI interfaces in core will be is supported.
- In case of SM in VRF, RP should be in ENCAP PE.
- IPv6 is not supported MVPN GRE.
- mLDP and MVPN GRE cannot be configured in the same VRF.

Information About Configuring Multicast VPN

Multicast VPN Operation

MVPN IP allows a service provider to configure and support multicast traffic in an MPLS VPN environment. This feature supports routing and forwarding of multicast packets for each individual VRF instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone.

A VPN is network connectivity across a shared infrastructure, such as an ISP. Its function is to provide the same policies and performance as a private network, at a reduced cost of ownership, thus creating many opportunities for cost savings through operations and infrastructure.

An MVPN allows an enterprise to transparently interconnect its private network across the network backbone of a service provider. The use of an MVPN to interconnect an enterprise network in this way does not change the way that enterprise network is administered, nor does it change general enterprise connectivity.

Benefits of Multicast VPN

- Provides a scalable method to dynamically send information to multiple locations.
- Provides high-speed information delivery.
- Provides connectivity through a shared infrastructure.

Multicast VPN Routing and Forwarding and Multicast Domains

MVPN introduces multicast routing information to the VPN routing and forwarding table. When a provider edge (PE) device receives multicast data or control packets from a customer edge (CE) router, forwarding is

performed according to the information in the Multicast VPN routing and forwarding instance (MVRF). MVPN does not use label switching.

A set of MVRFs that can send multicast traffic to each other constitutes a multicast domain. For example, the multicast domain for a customer that wanted to send certain types of multicast traffic to all global employees would consist of all CE routers associated with that enterprise.

Multicast Distribution Trees

MVPN establishes a static default MDT for each multicast domain. The default MDT defines the path used by PE routers to send multicast data and control messages to every other PE router in the multicast domain.

If Source Specific Multicast (SSM) is used as the core multicast routing protocol, then the multicast IP addresses used for the default and data multicast distribution tree (MDT) must be configured within the SSM range on all PE routers.

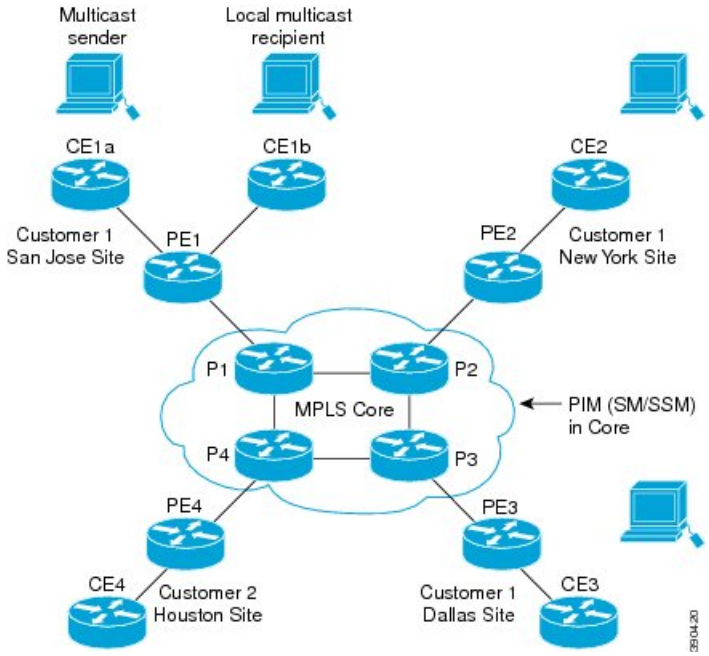
MVPN also supports the dynamic creation of MDTs for high-bandwidth transmission. Data MDTs are a feature unique to Cisco IOS software. Data MDTs are intended for high-bandwidth sources such as full-motion video inside the VPN to ensure optimal traffic forwarding in the MPLS VPN core. The threshold at which the data MDT is created can be configured on a per-router or a per-VRF basis. When the multicast transmission exceeds the defined threshold, the sending PE router creates the data MDT and sends a User Datagram Protocol (UDP) message, which contains information about the data MDT to all routers on the default MDT. The statistics to determine whether a multicast stream has exceeded the data MDT threshold are examined once every second. After a PE router sends the UDP message, it waits 3 more seconds before switching over; 13 seconds is the worst case switchover time and 3 seconds is the best case.

Data MDTs are created only for (S, G) multicast route entries within the VRF multicast routing table. They are not created for (*, G) entries regardless of the value of the individual source data rate.

In the following example, a service provider has a multicast customer with offices in San Jose, New York, and Dallas. A one-way multicast presentation is occurring in San Jose. The service provider network supports all three sites associated with this customer, in addition to the Houston site of a different enterprise customer.

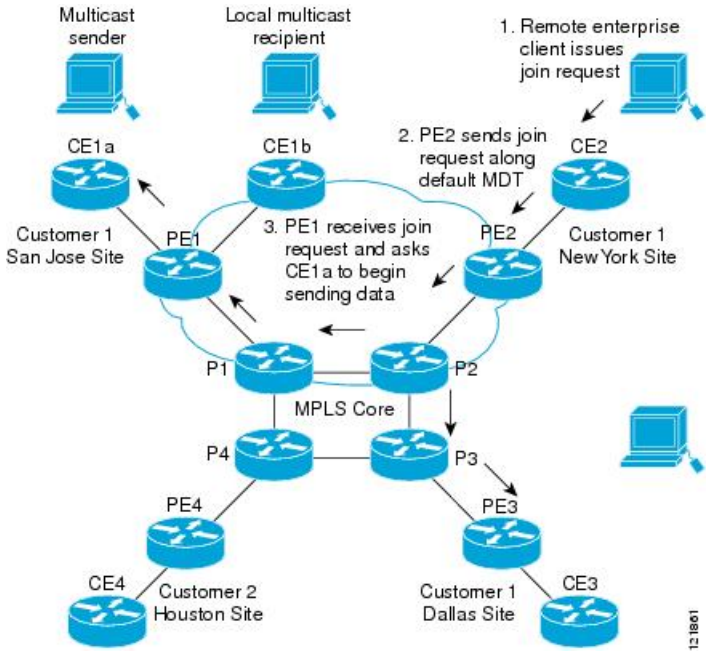
The default MDT for the enterprise customer consists of provider routers P1, P2, and P3 and their associated PE routers. PE4 is not part of the default MDT, because it is associated with a different customer. The figure shows that no data flows along the default MDT, because no one outside of San Jose has joined the multicast.

Figure 6: Default Multicast Distribution Tree Overview



An employee in New York joins the multicast session. The PE router associated with the New York site sends a join request that flows across the default MDT for the multicast domain of the customer. PE1, the PE router associated with the multicast session source, receives the request. The figure depicts that the PE router forwards the request to the CE router associated with the multicast source (CE1a).

Figure 7: Initializing the Data MDT



The CE router (CE1a) begins to send the multicast data to the associated PE router (PE1), which sends the multicast data along the default MDT. Through default MDT, traffic from CE1a is sent to all the PEs. Immediately sending the multicast data, PE1 recognizes that the multicast data exceeds the bandwidth threshold for which a data MDT should be created. Therefore, PE1 creates a data MDT, sends a message to all routers using the default MDT that contains information about the data MDT, and, three seconds later, begins sending the multicast data for that particular stream using the data MDT. Only PE2 has interested receivers for this source, so only PE2 will join the data MDT and receive traffic on it.

The other PE routers also receive traffic from the default MDT.

PE routers maintain a PIM relationship with other PE routers over the default MDT and a PIM relationship with its directly attached PE routers.

Multicast Tunnel Interface

An MVRF, which is created per multicast domain, requires the device to create a tunnel interface from which all MVRF traffic is sourced. A multicast tunnel interface is an interface that the MVRF uses to access the multicast domain. It can be thought of as a conduit that connects an MVRF and the global MVRF. One tunnel interface is created per MVRF.

MDT Address Family in BGP for Multicast VPN

The **mdt** keyword has been added to the **address-family ipv4** command to configure an MDT address-family session. MDT address-family sessions are used to pass the source PE address and MDT group address to PIM using Border Gateway Protocol (BGP) MDT Subaddress Family Identifier (SAFI) updates.

BGP Advertisement Methods for Multicast VPN Support

In a single autonomous system, if the default MDT for an MVPN is using PIM sparse mode (PIM-SM) with a rendezvous point (RP), then PIM is able to establish adjacencies over the Multicast Tunnel Interface (MTI) because the source PE and receiver PE discover each other through the RP. In this scenario, the local PE (the source PE) sends register messages to the RP, which then builds a shortest-path tree (SPT) toward the source PE. The remote PE, which acts as a receiver for the MDT multicast group, then sends (*, G) joins toward the RP and joins the distribution tree for that group.

However, if the default MDT group is configured in a PIM Source Specific Multicast (PIM-SSM) environment rather than a PIM-SM environment, the receiver PE needs information about the source PE and the default MDT group. This information is used to send (S, G) joins toward the source PE to build a distribution tree from the source PE (without the need for an RP). The source PE address and default MDT group address are sent using BGP.

BGP Extended Community

When BGP extended communities are used, the PE loopback (source address) information is sent as a VPNv4 prefix using Route Distinguisher (RD) Type 2 (to distinguish it from unicast VPNv4 prefixes). The MDT group address is carried in a BGP extended community. Using a combination of the embedded source in the VPNv4 address and the group in the extended community, PE routers in the same MVRF instance can establish SSM trees to each other.

Note Prior to the introduction of MDT SAFI support, the BGP extended community attribute was used as an interim solution to advertise the IP address of the source PE and default MDT group before IETF standardization. A BGP extended community attribute in an MVPN environment, however, has certain limitations: it cannot be used in inter-AS scenarios (because the attribute is nontransitive), and it uses RD Type 2, which is not a supported standard and not supported effective with Cisco IOS Release 15.5(1)T and Cisco IOS Release 15.4(3)S.

BGP MDT SAFI

Cisco software releases that support the MDT SAFI, the source PE address and the MDT group address are passed to PIM using BGP MDT SAFI updates. The RD type has changed to RD type 0, and BGP determines the best path for the MDT updates before passing the information to PIM.

Note To prevent backward-compatibility issues, BGP allows the communication of the older style updates with peers that are unable to understand the MDT SAFI address family.

Cisco software releases that support the MDT SAFI, the MDT SAFI address family needs to be explicitly configured for BGP neighbors using the **address-family ipv4 mdt** command. Neighbors that do not support the MDT SAFI still need to be enabled for the MDT SAFI in the local BGP configuration. Prior to the introduction of the MDT SAFI, additional BGP configuration from the VPNv4 unicast configuration was not needed to support MVPN.

Because the new MDT SAFI does not use BGP route-target extended communities, the regular extended community methods to filter these updates no longer apply. As a result, the **match mdt-group** route-map configuration command has been added to filter on the MDT group address using access control lists (ACLs). These route maps can be applied—inbound or outbound—to the IPv4 MDT address-family neighbor configuration.

Automigration to the MDT SAFI

When migrating a Cisco IOS release to the MDT SAFI, existing VPNv4 neighbors will be automatically configured for the MDT SAFI upon bootup based on the presence of an existing default MDT configuration (that is, pre-MDT SAFI configurations will be automatically converted to an MDT SAFI configuration upon bootup). In addition, when a default MDT configuration exists and a VPNv4 neighbor in BGP is configured, a similar neighbor in the IPv4 MDT address family will be automatically configured.

Note Because there is no VRF configuration on route reflectors (RRs), automigration to the MDT SAFI will not be triggered on RRs. The MDT SAFI configuration, thus, will need to be manually configured on RRs. Having a uniform MDT transmission method will reduce processing time on the routers (because MDT SAFI conversion is not necessary).

Guidelines for Configuring the MDT SAFI

- We recommend that you configure the MDT SAFI on all routers that participate in the MVPN. Even though the benefits of the MDT SAFI are for SSM tree building, the MDT SAFI must also be configured when using MVPN with the default MDT group for PIM-SM. From the multicast point of view, the

MDT SAFI is not required for MVPN to work within a PIM-SM core. However, in certain scenarios, the new address family must be configured in order to create the MTI. Without this notification, the MTI would not be created and MVPN would not function (even with PIM-SM).

- For backward compatible sessions, extended communities must be enabled on all MDT SAFI peers. In a pure MDT SAFI environment, there is no need to configure extended communities explicitly for MVPN. However, extended communities will be needed for VPNv4 interior BGP (iBGP) sessions to relay the route-target. In a hybrid (MDT SAFI and pre-MDT SAFI) environment, extended communities must be configured to send the embedded source in the VPNv4 address and the MDT group address to MDT SAFI neighbors.

Guidelines for Upgrading a Network to Support the MDT SAFI

When moving from a pre-MDT SAFI to an MDT SAFI environment, the upmost care should be taken to minimize the impact to the MVPN service. The unicast service will not be affected, other than the outage due to the reload and recovery. To upgrade a network to support the MDT SAFI, we recommend that you perform the following steps:

1. Upgrade the PEs in the MVPN to a Cisco IOS release that supports the MDT SAFI. Upon bootup, the PE configurations will be automigrated to the MDT SAFI. For more information about the automigration to the MDT SAFI functionality, see Automigration to the MDT SAFI, on page 84 section.
2. After the PEs have been upgraded, upgrade the RRs and enable the MDT SAFI for all peers providing MVPN service. Enabling or disabling the MDT SAFI will reset the BGP peer relationship for all address families; thus, a loss of routing information may occur.

Note A multihomed BGP RR scenario, one of the RRs must be upgraded and configured last. The upgraded PEs will use this RR to relay MDT advertisements while the other RRs are being upgraded.

Supported Policy

The following policy configuration parameters are supported under the MDT SAFI:

- Mandatory attributes and well-known attributes, such as the AS-path, multiexit discriminator (MED), BGP local-pref, and next-hop attributes.
- Standard communities, community lists, and route maps.

How to Configure Multicast VPN

Configuring a Default MDT Group for a VRF

Perform this task to configure a default MDT group for a VRF.

The default MDT group must be the same group configured on all devices that belong to the same VPN. The source IP address will be the address used to source the BGP sessions.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing distributed**
4. **ip multicast-routing vrf** *vrf-name* **distributed**
5. **ip vrf** *vrf-name*
6. **mdt default** *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	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing distributed** **Example:** `Device(config)# ip multicast-routing distributed`	Enables multicast routing.
Step 4	**ip multicast-routing vrf** *vrf-name* **distributed** **Example:** `Device(config)# ip multicast-routing vrf vrf1 distributed`	Supports the MVPN VRF instance.
Step 5	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vrf1`	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name. See the Example: Configuring the MDT Address Family in BGP for Multicast VPN, on page 96 section for an alternate command.
Step 6	**mdt default** *group-address* **Example:** `Device(config-vrf)# mdt default 232.0.0.1`	Configures the multicast group address for the default MDT for a VRF. • A tunnel interface is created as a result of this command. • By default, the destination address of the tunnel header is the *group-address* value.

Configuring the MDT Address Family in BGP for Multicast VPN

Perform this task to configure an MDT address family session on PE devices to establish MDT peering sessions for MVPN.

Before you begin

Before MVPN peering can be established through an MDT address family, MPLS and Cisco Express Forwarding (CEF) must be configured in the BGP network and multiprotocol BGP on PE devices that provide VPN services to CE devices.

Note The following policy configuration parameters are not supported:

- Route-originator attribute
- Network Layer Reachability Information (NLRI) prefix filtering (prefix lists, distribute lists)
- Extended community attributes (route target and site of origin)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4 mdt**
5. **neighbor** *neighbor-address* **activate**
6. **neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**]
7. **exit**
8. **address-family vpnv4**
9. **neighbor** *neighbor-address* **activate**
10. **neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**]
11. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65535`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4 mdt** **Example:** `Device(config-router)# address-family ipv4 mdt`	Enters address family configuration mode to create an IP MDT address family session.
Step 5	**neighbor** *neighbor-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 activate`	Enables the MDT address family for this neighbor.
Step 6	**neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Device(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables community and (or) extended community exchange with the specified neighbor.
Step 7	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.
Step 8	**address-family vpnv4** **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode to create a VPNv4 address family session.
Step 9	**neighbor** *neighbor-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 activate`	Enables the VPNv4 address family for this neighbor.
Step 10	**neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Device(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables community and (or) extended community exchange with the specified neighbor.
Step 11	**end** **Example:**	Exits address family configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-router-af)# end`	

Configuring the Data Multicast Group

A data MDT group can include a maximum of 256 multicast groups per VPN per VRF per PE device. Multicast groups used to create the data MDT group are dynamically chosen from a pool of configured IP addresses.

Before you begin

- Before configuring a default MDT group, the VPN must be configured for multicast routing as described in the "Configuring a Default MDT Group for a VRF" section.
- All access lists needed when using the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the “Creating an IP Access List and Applying It to an Interface” module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **mdt data** *group-address-range wildcard-bits* [**threshold** *kbps*] [**list** *access-list*]
5. **mdt log-reuse**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vrf1`	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name. See the Example: Configuring MVPN and SSM, on page 95 section for an alternate command.
Step 4	**mdt data** *group-address-range wildcard-bits* [**threshold** *kbps*] [**list** *access-list*] **Example:**	Specifies a range of addresses to be used in the data MDT pool.

	Command or Action	Purpose
	`Device(config-vrf)# mdt data 239.192.20.32 0.0.0.15 threshold 1`	• For the *group-address-range* and *wildcard-bits* arguments, specify a a multicast group address range. The range is from 224.0.0.1 to 239.255.255.255. Because the range of addresses used in the data MDT pool are multicast group addresses (Class D addresses), there is no concept of a subnet; therefore, you can use all addresses in the mask (wildcard) range that you specify for the *wildcard-bits* argument. • The threshold is in *kbps*. The range is from 1 through 4294967. • Use the optional **list** keyword and *access-list* argument to define the (S, G) MVPN entries to be used in a data MDT pool, which would further limit the creation of a data MDT pool to the particular (S, G) MVPN entries defined in the access list specified for the *access-list* argument
Step 5	**mdt log-reuse** **Example:** `Device(config-vrf)# mdt log-reuse`	(Optional) Enables the recording of data MDT reuse and generates a syslog message when a data MDT has been reused.
Step 6	**end** **Example:** `Device(config-vrf)# end`	Returns to privileged EXEC mode.

Configuring Multicast Routes and Information

Perform this task to limit the number of multicast routes that can be added in a device.

Before you begin

- Before configuring a default MDT group, the VPN must be configured for multicast routing as described in the "Configuring a Default MDT Group for a VRF" section.
- All access lists needed when using the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the " Creating an IP Access List and Applying It to an Interface " module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast vrf** *vrf-name* **route-limit** *limit* [*threshold*]
4. **ip multicast mrinfo-filter** *access-list*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast vrf** *vrf-name* **route-limit** *limit* [*threshold*] **Example:** `Device(config)# ip multicast vrf cisco route-limit 500 50`	Sets the mroute limit and the threshold parameters.
Step 4	**ip multicast mrinfo-filter** *access-list* **Example:** `Device(config)# ip multicast mrinfo-filter 4`	Filters the multicast device information request packets for all sources specified in the access list.

Verifying Information for the MDT Default Group

SUMMARY STEPS

1. **enable**
2. **show ip pim mdt bgp**
3. **show ip pim vrf** *vrf-name* **mdt history interval** *minutes*

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ip pim mdt bgp**

Example:

```
Device# show ip pim mdt bgp

MDT (Route Distinguisher + IPv4)              Router ID         Next Hop
```

```
  MDT group 238.2.2.0
   2:200:50.0.0.4                          0.0.0.0          0.0.0.0
  MDT group 239.1.1.1
   2:200:50.0.0.4                          50.0.0.4         50.0.0.4
```

Displays information about the BGP advertisement of the RD for the MDT default group.

Step 3 **show ip pim vrf** *vrf-name* **mdt history interval** *minutes*

Example:

```
Device# show ip pim vrf vrf1 mdt history interval 20

   MDT-data send history for VRF - vrf1 for the past 20 minutes
MDT-data group        Number of reuse
     10.9.9.8           3
     10.9.9.9           2
```

Displays the data MDTs that have been reused during the past configured interval.

Troubleshoot

The following are some troubleshooting tips for MVPN:

- Use the **show ip pim vrf neighbor** command to check that PE routers established a PIM neighbor relationship through the dynamic tunnel interface. If they did, then the Default MDT operates properly.
- If the Default MDT does not function, use the **show ip pim mdt bgp** command to check that loopbacks of remote PE routers participating in MVPN are known by the local router. If they are not, verify that PIM is enabled on interfaces used as a source of MP BGP sessions.

Verifying Information for the Data Multicast Group

SUMMARY STEPS

1. **show ip pim** [**vrf** *vrf-name*] **mdt send**
2. **show ip pim** [**vrf** *vrf-name*] **mdt receive**

DETAILED STEPS

Step 1 **show ip pim** [**vrf** *vrf-name*] **mdt send**

Example:

```
Device# show ip pim vrf VPN_A mdt send

MDT-data send list for VRF: VPN_A
  (source, group)                    MDT-data group/num   ref_count
  (80.0.0.10, 232.1.1.1)             238.2.2.0            1
```

Displays detailed information about the MDT data group including MDT advertisements that the specified device has made.

Step 2 **show ip pim** [**vrf** *vrf-name*] **mdt receive**

Example:

```
Device# show ip pim vrf VPN_A mdt receive
Joined MDT-data [group/mdt number : source]  uptime/expires for VRF: VPN_A
 [238.2.2.0 : 50.0.0.4]  00:51:27/00:02:32
```

Displays detailed information about the MDT data group joined.

Verifying Information for the Multicast Routes

SUMMARY STEPS

1. **show ip mroute**
2. **show ip mroute vrf** *vrf name*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**show ip mroute**
	Example:
	`Device# show ip mroute`
	`IP Multicast Routing Table`
	`Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,`
	`L - Local, P - Pruned, R - RP-bit set, F - Register flag,`
	`T - SPT-bit set, J - Join SPT, M - MSDP created entry, 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`
	`Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join`
	`Timers: Uptime/Expires`
	`Interface state: Interface, Next-Hop or VCD, State/Mode`
	
	`(*, 239.1.1.1), 05:06:08/stopped, RP 0.0.0.0, flags: DCZ`
	`Incoming interface: Null, RPF nbr 0.0.0.0`
	`Outgoing interface list:`
	`TenGigabitEthernet0/2/0, Forward/Sparse-Dense,`
		Displays the contents of the IP multicast routing table in the provider's core.

	Command or Action	Purpose
	01:08:04/stopped MVRF VPN_A, Forward/Sparse-Dense, 05:06:08/stopped (50.0.0.4, 239.1.1.1), 01:04:27/00:01:31, flags: TZ Incoming interface: TenGigabitEthernet0/2/0, RPF nbr 60.0.0.3 Outgoing interface list: MVRF VPN_A, Forward/Sparse-Dense, 01:04:27/stopped (50.0.0.2, 239.1.1.1), 05:06:07/00:02:42, flags: T Incoming interface: Loopback50, RPF nbr 0.0.0.0 Outgoing interface list: TenGigabitEthernet0/2/0, Forward/Sparse-Dense, 01:08:04/stopped (*, 238.2.2.0), 00:52:26/stopped, RP 0.0.0.0, flags: DCZ Incoming interface: Null, RPF nbr 0.0.0.0 Outgoing interface list: TenGigabitEthernet0/2/0, Forward/Sparse-Dense, 00:52:26/stopped MVRF VPN_A, Forward/Sparse-Dense, 00:52:26/stopped (*, 224.0.1.40), 05:09:15/00:02:47, RP 0.0.0.0, flags: DCL Incoming interface: Null, RPF nbr 0.0.0.0 Outgoing interface list: TenGigabitEthernet0/2/0, Forward/Sparse-Dense, 01:08:04/stopped Loopback50, Forward/Sparse-Dense, 05:09:15/stopped	
Step 2	**show ip mroute vrf** *vrf name* **Example:** Device# **show ip mroute vrf VPN_A** IP Multicast Routing Table Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected, L - Local, P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet, X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement, U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel, z - MDT-data group sender, Y - Joined MDT-data group, y - Sending to MDT-data group, G - Received BGP C-Mroute, g - Sent BGP C-Mroute, N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed, Q - Received BGP S-A Route, q - Sent BGP S-A Route,	Displays the multicast routing table in the client's VRF.

	Command or Action	Purpose
	V - RD & Vector, v - Vector, p - PIM Joins on route, x - VxLAN group Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join Timers: Uptime/Expires Interface state: Interface, Next-Hop or VCD, State/Mode (80.0.0.10, 232.1.1.1), 00:53:04/00:02:59, flags: sTIY Incoming interface: Tunnel0, RPF nbr 50.0.0.4, MDT:238.2.2.0/00:02:55 Outgoing interface list: BDI1101, Forward/Sparse-Dense, 00:53:04/00:02:59 (*, 224.0.1.40), 05:06:46/00:02:15, RP 0.0.0.0, flags: DCL Incoming interface: Null, RPF nbr 0.0.0.0 Outgoing interface list: Tunnel0, Forward/Sparse-Dense, 05:06:46/stopped	

Configuration Examples for Multicast VPN

Example: Configuring MVPN and SSM

In the following example, PIM-SSM is configured in the backbone. Therefore, the default and data MDT groups are configured within the SSM range of IP addresses. Inside the VPN, PIM-SM is configured and only Auto-RP announcements are accepted.

```
ip vrf vrf1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 mdt default 232.0.0.1
 mdt data 232.0.1.0 0.0.0.255 threshold 500 list 101
!
ip pim ssm default
ip pim vrf vrf1 accept-rp auto-rp
```

In the following example, an alternate command is used to configure MVPN and SSM:

```
vrf definition vrf1
rd 101:1
route-target export 101:1
route-target import 101:1
!
address-family ipv4
mdt default 232.1.1.1
mdt data 232.5.1.1 0.0.0.255 threshold 500 list 101
exit-address-family
!
ip pim ssm default
ip pim vrf vrf1 accept-rp auto-rp
```

Example: Enabling a VPN for Multicast Routing

In the following example, multicast routing is enabled with a VPN routing instance named vrf1:

```
ip multicast-routing vrf vrf1 distributed
```

Example: Configuring the MDT Address Family in BGP for Multicast VPN

In the following example, an MDT address family session is configured on a PE router to establish MDT peering sessions for MVPN.

```
!
ip vrf test
 rd 55:2222
 route-target export 55:2222
 route-target import 55:2222
 mdt default 232.0.0.1
!
ip multicast-routing distributed
ip multicast-routing vrf test distributed
!
router bgp 55
.
.
.
!
 address-family vpnv4
 neighbor 192.168.1.1 activate
 neighbor 192.168.1.1 send-community both
!
 address-family ipv4 mdt
 neighbor 192.168.1.1 activate
 neighbor 192.168.1.1 send-community both
!
```

In the following example, an alternate command is used to configure an MDT address family session on a PE router to establish MDT peering sessions for MVPN:

```
vrf definition vrf1
rd 101:1
route-target export 101:1
route-target import 101:1
!
address-family ipv4
mdt default 232.1.1.1
.
.
.

ip multicast-routing distributed
ip multicast-routing vrf test distributed
!
router bgp 55
.
.
.
!
address-family vpnv4
neighbor 192.168.1.1 activate
neighbor 192.168.1.1 send-community both
```

```
!
address-family ipv4 mdt
neighbor 192.168.1.1 activate
neighbor 192.168.1.1 send-community both
```

Example: Configuring the Multicast Group Address Range for Data MDT Groups

In the following example, the VPN routing instance is assigned a VRF named VPN_A. The MDT default group for a VPN VRF is 239.1.1.1, and the multicast group address range for MDT groups is 239.2.2.0 with wildcard bits of 0.0.0.255:

```
ip vrf VPN_A
rd 2:200
route-target export 2:200
route-target import 2:200
mdt default 239.1.1.1
mdt data 239.2.2.0 0.0.0.255
```

The following is an alternate command to assign VRF to the VPN routing instance.

```
vrf definition VPN_A
rd 101:1
route-target export 101:1
route-target import 101:1
!
address-family ipv4
mdt default 232.1.1.1
mdt data 232.5.1.1 0.0.0.255 threshold 500 list 101
```

Example: Limiting the Number of Multicast Routes

In the following example, the number of multicast routes that can be added to a multicast routing table is set to 500 and the threshold value of the number of mroutes that will cause a warning message to occur is set to 50:

```
ip multicast route-limit 500 50
ip multicast vrf VPN_A route-limit 500 50
no mpls traffic-eng auto-bw timers
!
```

Example: Configuring MVPN on VRF

```
ip multicast-routing distributed
ip multicast-routing vrf VPN_A distributed
!

ip pim ssm default
ip pim vrf VPN_A ssm default
!

interface loopback50
ip address 50.0.0.2 255.255.255.255
ip pim sparse-dense-mode
ip ospf 1 area 0
exit
!
```

Example: Configuring Access-Interface

```
interface GigabitEthernet0/1/0
no ip address
negotiation auto
service instance 1101 ethernet
encapsulation dot1q 1101
rewrite ingress tag pop 1 symmetric
bridge-domain 1101
!
interface BDI1101
ip vrf forwarding VPN_A
ip address 40.0.0.2 255.255.255.0
ip pim sparse-dense-mode
ip igmp version 3
ip ospf 2 area 0
end
```

The following is an alternate method to configure a physical interface or BDI under specified VRF:

```
interface GigabitEthernet0/1/0
no ip address
negotiation auto
service instance 1101 ethernet
encapsulation dot1q 1101
rewrite ingress tag pop 1 symmetric
bridge-domain 1101
!
interface BDI1101
vrf forwarding VPN_A
ip address 40.0.0.2 255.255.255.0
ip pim sparse-dense-mode
ip igmp version 3
ip ospf 2 area 0
end
```

Example: Configuring Core Interfaces

```
interface ten 0/2/0
ip address 60.0.0.2 255.255.255.0
ip pim sparse-dense-mode
ip igmp version 3
ip ospf 1 area 0
end
```

Example: Configuring BGP

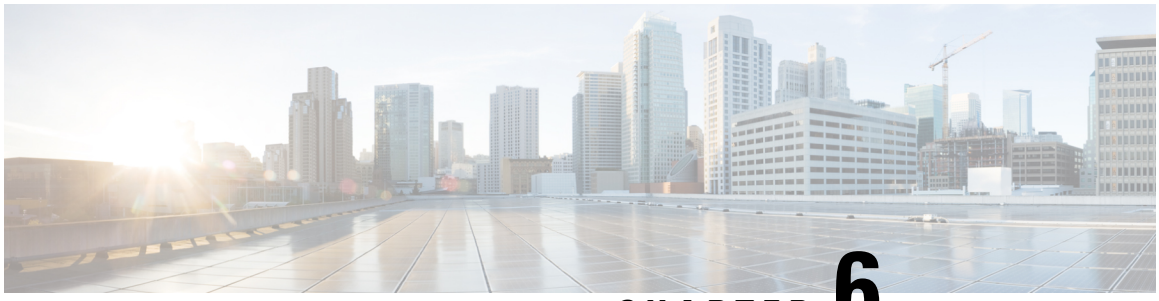
```
bgp log-neighbor-changes
no bgp default ipv4-unicast
neighbor 50.0.0.4 remote-as 100
neighbor 50.0.0.4 update-source Loopback50
!
address-family ipv4
neighbor 50.0.0.4 activate
exit-address-family

address-family vpnv4
neighbor 50.0.0.4 activate
neighbor 50.0.0.4 send-community extended
exit-address-family
```

```
address-family ipv4 mdt
neighbor 50.0.0.4 activate
neighbor 50.0.0.4 send-community extended
exit-address-family

address-family ipv4 vrf VPN_A
redistribute connected
redistribute static
redistribute ospf 2
exit-address-family
!
!
router ospf 2 vrf VPN_A
redistribute connected subnets
redistribute static subnets
redistribute bgp 100 subnets
exit
```

-



CHAPTER 6

Configuring Multicast VPN Extranet Support

The Multicast VPN Extranet Support feature (sometimes referred to as the MVPN Extranet Support feature) enables service providers to distribute IP multicast content originated from one enterprise site to other enterprise sites. This feature enables service providers to offer the next generation of flexible extranet services, helping to enable business partnerships between different enterprise VPN customers.

This module describes the concepts and the tasks related to configuring Multicast VPN Extranet Support.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring Multicast VPN Extranet Support

- You are familiar with IP multicast concepts and configuration tasks.
- You are familiar with Multicast VPN (MVPN) concepts and configuration tasks.
- You are familiar with Multiprotocol Label Switching (MPLS) Layer 3 Virtual Private Network (VPN) concepts and configuration tasks.

Restrictions for Configuring Multicast VPN Extranet Support

- The Multicast VPN Extranet Support feature supports only Protocol Independent Multicast (PIM) sparse mode (PIM-SM) and Source Specific Multicast (SSM) traffic; PIM dense mode (PIM-DM) and bidirectional PIM (bidir-PIM) traffic are not supported.
- When configuring extranet MVPNs in a PIM-SM environment, the source and the rendezvous point (RP) must reside in the same site of the MVPN behind the same provider edge (PE) router.
- It is required to configure either all the Receiver MVRF(s) in Source PE or Source MVRF in all the Receiver PE(s) to deliver the Extranet content.
- IPV6 based MVPN Extranet is *not* supported.
- Only Routed interfaces and Routed interfaces on Port channels are supported towards the core. BDI towards core is *not* supported.
- The scale data for MVPN extranet is as follows:
  - Maximum number of mVRFs supported is 20
  - Maximum number of mroutes supported (Intranet + extranet) is 1000 in case of default template and 2000 in case of Video template.
- PIM-SM and PIM-SSM are supported.
- PIM-DM and bidir-PIM are *not* supported.
- RP must be configured behind the PE router and the source is in the same intranet-MVPN and behind the CE router.
- Static mroute with fallback-lookup option is supported for RPF lookup[2].
- Configuring the Receiver mVRF on the Source PE only to implement MVPN Extranet support is *not* supported.

Information About Multicast VPN Extranet Support

Overview of MVPN Extranet Support

An extranet can be viewed as part of a company's intranet that is extended to users outside the company. It has also been described as a "state of mind" in which a VPN is used as a way to do business with other companies as well as to sell products and content to customers and companies. An extranet is a VPN connecting the corporate site or sites to external business partners or suppliers to securely share part of a business's information or operations among them.

MPLS VPNs inherently provide security, ensuring that users access only appropriate information. MPLS VPN extranet services offer extranet users unicast connectivity without compromising the integrity of their corporate data. The Multicast VPN Extranet Support feature extends this offer to include multicast connectivity to the extranet community of interest.

The Multicast VPN Extranet Support feature enables service providers to distribute IP multicast content originated from one enterprise site to other enterprise sites. This feature enables service providers to offer the next generation of flexible extranet services, helping to enable business partnerships between different enterprise VPN customers. Using this feature, service providers can offer multicast extranet contracts to meet various business partnership requirements, including short-term, annual, and rolling contracts.

Benefits of MVPN Extranet Support

The Multicast VPN Extranet Support feature can be used to solve such business problems as:

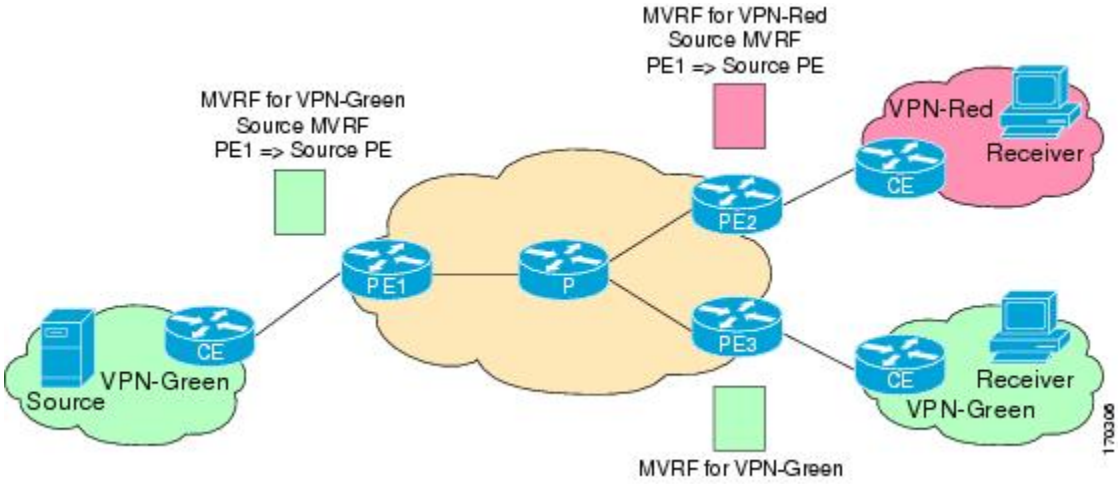
- Efficient content distribution between enterprises
- Efficient content distribution from service providers or content providers to their different enterprise VPN customers

Components of an Extranet MVPN

The figure below illustrates the components that constitute an extranet MVPN.

- **MVRF** --Multicast VPN routing and forwarding (VRF) instance. An MVRF is a multicast-enabled VRF. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a provider edge (PE) router.
- **Source MVRF** --An MVRF that can reach the source through a directly connected customer edge (CE) router.
- **Receiver MVRF** --An MVRF to which receivers are connected through one or more CE devices.
- **Source PE** --A PE router that has a multicast source behind a directly connected CE router.
- **Receiver PE** --A PE router that has one or more interested receivers behind a directly connected CE router.

Figure 8: Components of an Extranet MVPN



Solution for MVPN Extranet Support

For unicast, there is no difference between an intranet or extranet from a routing perspective; that is, when a VRF imports a prefix, that prefix is reachable through a label-switched path (LSP). If the enterprise owns the prefix, the prefix is considered a part of the corporate intranet; otherwise, the prefix is considered a part of an extranet. For multicast, however, the reachability of a prefix (especially through an LSP) is not sufficient to build a multicast distribution tree (MDT).

In order to provide support for extranet MVPN services, the same default MDT group must be configured in the source and receiver MVRF. Prior to the introduction of the Multicast VPN Extranet Support feature, there were challenges that prevented service providers from providing extranet MVPN services:

- The source MVRF may not have been configured with a default MDT group, or it may have been configured with a different MDT group as compared to the receiver MVRF. In the former case there was no way for the source MVRF to forward multicast streams to extranet sites, and in the latter case, there was no way for the separate MVRFs to be linked.
- It was not possible to maintain a forwarding table in cases where the RPF interface and outgoing interfaces belong to different VRFs.

The Multicast VPN Extranet Support feature solves these challenges as follows:

- The receiver and source MVRF multicast route (mroute) entries are linked.
- The Reverse Path Forwarding (RPF) check relies on unicast routing information to determine the interface through which the source is reachable. This interface is used as the RPF interface.

Configuration Guidelines for MVPN Extranet Support

Two configuration options are available to provide extranet MVPN services:

- Option 1: Source Side Chaining (SSC)--Configure the receiver MVRF on the source PE router.
- Option 2: Receiver Side Chaining (RSC)--Configure the source MVRF on the receiver PE router.

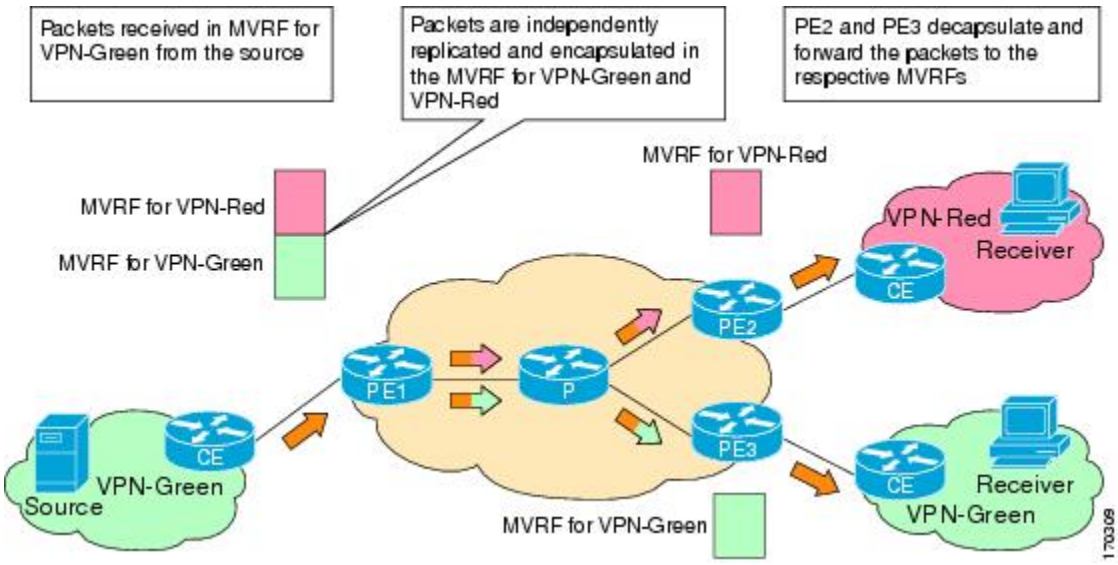
MVPN Extranet Support Configuration Guidelines for Option 1

To provide extranet MVPN services to enterprise VPN customers by configuring the receiver MVRF on the source PE router (Option 1), you would complete the following procedure:

- For each extranet site, you would configure an additional MVRF on the source PE router, that has the same default MDT group as the receiver MVRF, if the MVRF is not configured on the source PE.
- In the receiver MVRF configuration, you would configure the same unicast routing policy on the source and receiver PE routers to import routes from the source MVRF to the receiver MVRF.

The figure illustrates the flow of multicast traffic in an extranet MVPN topology where a receiver MVRF is configured on the source PE router (Option 1). In the topology, an MVRF is configured for VPN-Green and VPN-Red on PE1, the source PE router. A multicast source behind PE1 is sending out a multicast stream to the MVRF for VPN-Green, and there are interested receivers behind PE2 and PE3, the receiver PE routers for VPN-Red and VPN-Green, respectively. After PE1 receives the packets from the source in the MVRF for VPN-Green, it independently replicates and encapsulates the packets in the MVRF for VPN-Green and VPN-Red and forwards the packets. After receiving the packets from this source, PE2 and PE3 decapsulate and forward the packets to the respective MVRFs.

Figure 9: Packet Flow for MVPN Extranet Support Configuration Option 1



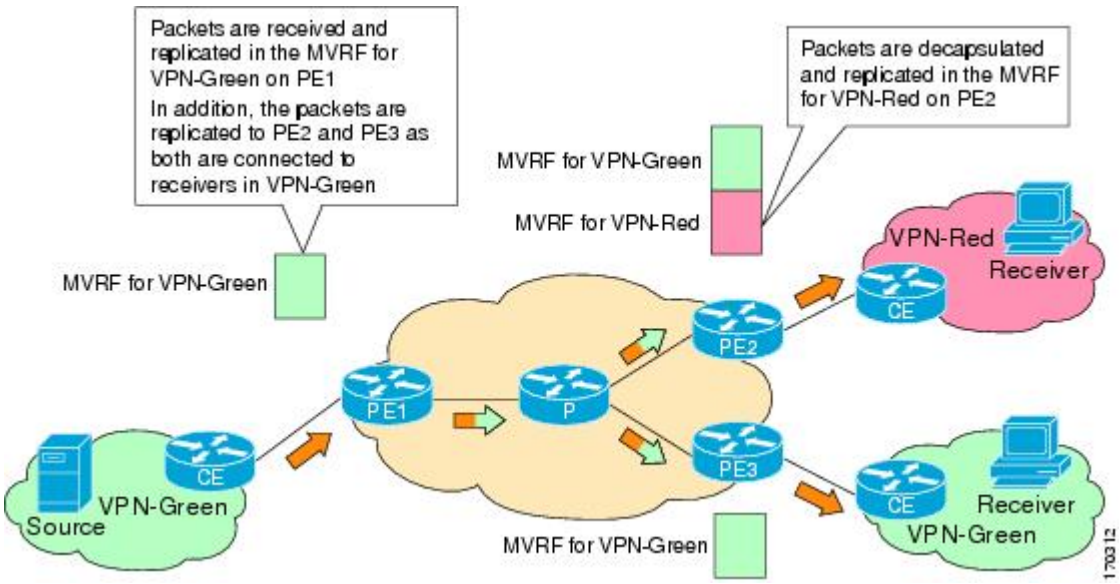
MVPN Extranet Support Configuration Guidelines for Option 2

To provide extranet MVPN services to enterprise VPN customers by configuring a source MVRF on a receiver PE router (Option 2), you would complete the following procedure:

- On a receiver PE router that has one or more interested receivers in a extranet site behind a directly connected CE router, configure an additional MVRF that has the same default MDT group as the site connected to the multicast source, if the MVRF is not configured.
- On the receiver PE router, you would configure the same unicast routing policy to import routes from the source MVRF to the receiver MVRF.

The figure illustrates the flow of multicast traffic in an extranet MVPN topology where the source MVRF is configured on a receiver PE router (Option 2). In the topology, an MVRF is configured for VPN-Green and VPN-Red on PE2, a receiver PE router. A multicast source behind PE1, the source PE router, is sending out a multicast stream to the MVRF for VPN-Green, and there are interested receivers behind PE2, the receiver PE router for VPN-Red, and behind PE3, the receiver PE router for VPN-Green. After PE1 receives the packets from the source in the MVRF for VPN-Green, it replicates and forwards the packets to PE2 and PE3, because both routers are connected to receivers in VPN-Green. The packets that originated from VPN-Green are then replicated on PE2 and forwarded to the interested receivers in VPN-Red and are replicated on PE3 and forwarded to the interested receivers in VPN-Green.

Figure 10: Packet Flow for MVPN Extranet Support Configuration Option 2



RPF for MVPN Extranet Support Using Imported Routes

You must configure either the receiver MVRF on the source PE router (Option 1) or the source MVRF on the receiver PE router (Option 2) for extranet links to be created. Once configured, RPF relies on unicast routing information to determine the interface through which the source is reachable. This interface is used as the RPF interface. No additional configuration is required for RPF resolution. The Multicast VPN Extranet Support feature supports RPF from one VRF to another VRF, from a VRF to the global routing table, and from the global routing table to a VRF.

RPF for MVPN Extranet Support Using Static Mroutes

Note This capability is not supported for MVPNv6 extranet.

By default, an extranet MVPN relies on unicast routing policies to determine the RPF interface. When the RPF lookup originates in a receiver MVRF, and it finds that the RPF interface does not lie in the same MVRF, the router uses the information in the Border Gateway Protocol (BGP) imported route to determine the source MVRF. The RPF lookup then continues and resolves in the source MVRF. In cases where the multicast and unicast topologies are incongruent, you can override the default behavior by configuring a static mroute in the receiver MVRF to explicitly specify the source MVRF using the **ip mroute** command with the **fallback-lookup** keyword and **vrf** *vrf-name* keyword and argument.

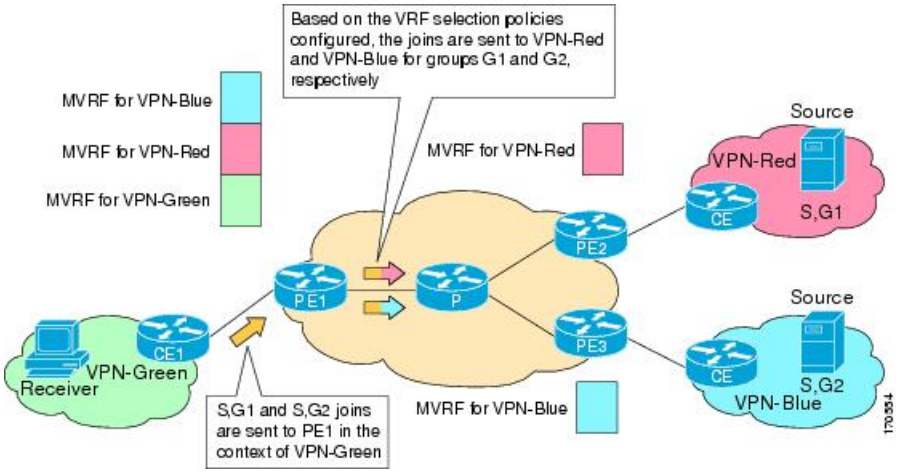
Static mroutes can also be configured to support RPF for extranet MVPN in the case where the source is present in an MVRF and the receiver is in the global table. In this case, because BGP does not allow VPNv4 routes to be imported into the IPv4 routing table, unicast cannot obtain the source MVRF information needed to resolve the RPF lookup. To enable the RPF lookup to be resolved in this case, a static mroute can be configured to explicitly specify the source MVRF using the **ip mroute** command with the **fallback-lookup** keyword and the **global** keyword.

Multicast VPN Extranet VRF Select

The Multicast VPN VRF Select feature is configured by creating group-based VRF selection policies. Group-based VRF selection policies are configured using the **ip multicast rpf select** command. The **ip multicast rpf select**command is used to configure RPF lookups originating in a receiver MVRF or in the global routing table to be resolved in a source MVRF or in the global routing table based on group address. Access Control Lists (ACLs) are used to define the groups to be applied to group-based VRF selection policies.

The figure illustrates an extranet MVPN topology with the Multicast VPN VRF Select feature configured. In this topology, (S, G1) and (S, G2) PIM joins originating from VPN-Green, the receiver VRF, are forwarded to PE1, the receiver PE. Based on the group-based VRF selection policies configured, PE1 sends the PIM joins to VPN-Red and VPN-Blue for groups G1 and G2, respectively.

Figure 11: RPF Lookups Using Group-Based VRF Selection Policies



How to Configure Multicast VPN Extranet Support

Configuring MVPN Support

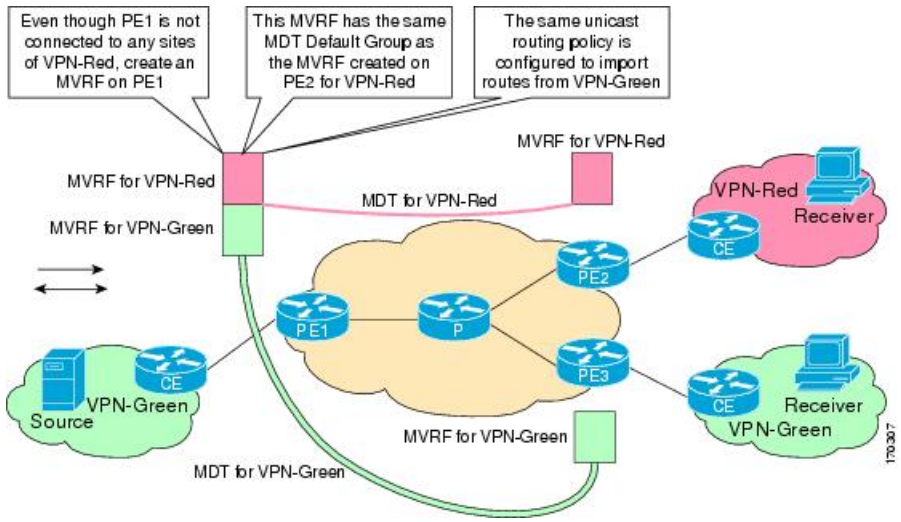
Perform one of the following tasks to provide extranet MVPN capabilities in an IPv4 core network:

Configuring the Receiver MVRF on the Source PE - Option 1 (SSC)

Perform this task to configure the receiver MVRF on the source PE router (Option 1) and provide support for extranet MVPN services.

In the following figure, the source PE router is PE1. To provide extranet MVPN services from one enterprise VPN site (VPN-Green) to another enterprise VPN site (VPN-Red) using Option 1, configure the receiver MVRF on the source PE router. In the receiver MVRF configuration, the default MDT group must be the same on both the source and receiver PE routers. In addition, you must configure the same unicast routing policy to import routes from the source MVRF (he MVRF for VPN-Green) to the receiver MVRF (the MVRF for VPN-Red).

Figure 12: Topology for MVPN Extranet Support Configuration Option 1



Before you begin

Intranet VPN in the source and receiver VPNs must be already configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **vpn id** *oui:vpn-index*
6. **route-target** { **export** | **import** | **both** } *route-target-ext-community*
7. **mdt default mpls mldp** *PE/P router_loopback-ip*
8. **mdt data mpls mldp** *number of data mdt*
9. **mdt data threshold** *threshold value*
10. **end**
11. **show ip mroute** [**vrf** *vrf-name*] *group-address*

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf VPN-Red`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.

Step 4	**rd** *route-distinguisher*
	Example:
	`Router(config-vrf)# rd 55:2222`
		Creates routing and forwarding tables.
		• Specify the *route-distinguisher* argument to add an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats:
		  • 16-bit autonomous system number: your 32-bit number, for example, 101:3
		  • 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1

Step 5	**vpn id** *oui:vpn-index* **Example:** `Router(config-vrf)# vpn id 1:2`	Assigns the VPN ID to the VRF oui. An organizationally unique identifier. The OUI is restricted to three octets. The vpn-index—This value identifies the VPN within the company. This VPN index is restricted to four octets.
Step 6	**route-target** { **export** | **import** | **both** } *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 55:1111`	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y). **Note** This command works only if BGP is running.
Step 7	**mdt default mpls mldp** *PE/P router_loopback-ip* **Example:** `Router(config-vrf)# mdt default mpls mldp 20.100.0.4`	The default MDT defines the path used by PE routers to send multicast data and control messages to every other PE router in the multicast domain. Loop-back address of any PE routers can be configured.
Step 8	**mdt data mpls mldp** *number of data mdt* **Example:** `Router(config-vrf)# mdt default mpls mldp 10.1.1.10`	Creating data mdt by mentioning the number of data mdts to be created. The range is from 1 to 5000.
Step 9	**mdt data threshold** *threshold value* **Example:** `Router(config-vrf)# mdt default threshold 1`	The threshold is in kbps. The range is from 1 through 4294967.

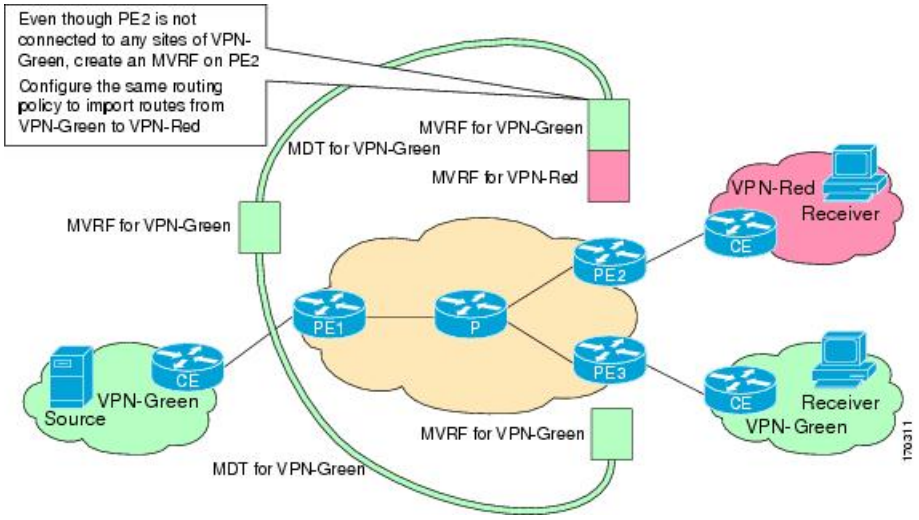
	Command or Action	Purpose
Step 10	**end** **Example:** `Router(config-vrf)# end`	Exits VRF configuration mode and returns to privileged EXEC mode.
Step 11	**show ip mroute** [**vrf** *vrf-name*] *group-address* **Example:** `Router# show ip mroute 232.3.3.3`	(Optional) Displays the contents of the IP multicast mroute table for a specific group address.

Configuring the Source MVRF on the Receiver PE - Option 2 (RSC)

Perform this task to configure the source MVRF on the receiver PE router (Option 2) and provide support for extranet MVPN services.

In the following figure, the receiver PE router is PE2. To provide support for extranet MVPN services from one enterprise VPN site (VPN-Green) to another enterprise VPN site (VPN-Red) using Option 2, configure the source MVRF on the receiver PE router. The MDT group configuration of the source MVRF must be the same on both the source and receiver PE routers. In addition, you must configure the same unicast routing policy to import routes from the source MVRF (the MVRF for VPN-Green) to the receiver MVRF (the MVRF for VPN-Red).

Figure 13: Topology for MVPN Extranet Support Configuration Option 2



Before you begin

Intranet VPN in the source and receiver VPNs must be already configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*

4. **rd** *route-distinguisher*
5. **vpn id** *oui:vpn-index*
6. **route-target** { **export** | **import** | **both** } *route-target-ext-community*
7. **mdt default mpls mldp** *PE/P router_loopback-ip*
8. **mdt data mpls mldp** *number of data mdt*
9. **mdt data threshold** *threshold value*
10. **end**
11. **show ip mroute** [**vrf** *vrf-name*] [*group-address*] verbose
12. **show mls ip multicast group** *group-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf VPN-Red`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.

Step 4	**rd** *route-distinguisher*
	Example:
	`Router(config-vrf)# rd 55:1111`
		Creates routing and forwarding tables.
		• The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats:
		  • 16-bit autonomous system number: your 32-bit number, for example, 101:3
		  • 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1

Step 5	**vpn id** *oui:vpn-index* **Example:** `Router(config-vrf)# vpn id 1:2`	Assigns the VPN ID to the VRF oui. An organizationally unique identifier. The OUI is restricted to three octets. The vpn-index—This value identifies the VPN within the company. This VPN index is restricted to four octets.
Step 6	**route-target** { **export**	**import**	**both** } *route-target-ext-community* **Example:**	Creates a list of import, export, or import and export route target communities for the specified VRF.

	Command or Action	Purpose
	`Router(config-vrf)# route-target import 55:1111`	Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y). **Note** This command works only if BGP is running.
Step 7	**mdt default mpls mldp** *PE/P router_loopback-ip* **Example:** `Router(config-vrf)# mdt default mpls mldp 20.100.0.4`	The default MDT defines the path used by PE routers to send multicast data and control messages to every other PE router in the multicast domain. Loop-back address of any PE routers can be configured.
Step 8	**mdt data mpls mldp** *number of data mdt* **Example:** `Router(config-vrf)# mdt default mpls mldp 10.1.1.10`	Creating data mdt by mentioning the number of data mdts to be created. The range is from 1 to 5000.
Step 9	**mdt data threshold** *threshold value* **Example:** `Router(config-vrf)# mdt default threshold 1`	The threshold is in kbps. The range is from 1 through 4294967.
Step 10	**end** **Example:** `Router(config-vrf)# end`	Exits VRF configuration mode and returns to privileged EXEC mode.
Step 11	**show ip mroute** [**vrf** *vrf-name*] [*group-address*] verbose **Example:** `Router# show ip mroute vrf vpn_a 224.1.1.1 verbose`	To display the detailed contents of the multicast routing (mroute) table for the multicast-group address, use the show ip mroute command in user EXEC or privileged EXEC mode.
Step 12	**show mls ip multicast group** *group-address* **Example:** `Router# show mls ip multicast group 232.3.3.3`	(Optional) Displays MLS information related to a specific multicast group.

Configuring RPF for MVPN Extranet Support Using Static Mroutes

Note This task is not supported for MVPNv6 extranet.

Before you begin

You must configure support for extranet MVPN services prior to performing this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip mroute vrf** *vrf-name source-address mask* **fallback-lookup** {**global** | **vrf** *vrf-name*} [*distance*]
4. **end**
5. **show ip mroute** [**vrf** *vrf-name*] *group-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip mroute vrf** *vrf-name source-address mask* **fallback-lookup** {**global** | **vrf** *vrf-name*} [*distance*] **Example:** `Router(config)# ip mroute vrf VPN-Red 224.100.0.5 255.255.255.255 fallback-lookup vrf VPN-Green`	(For IPv4 only) Configures the RPF lookup originating in a receiver MVRF to continue and be resolved in a source MVRF or in the global routing table using a static mroute. • The **global** keyword is used to specify that the source MVRF is in the global routing table. • The **vrf** keywordand *vrf-name* argument are used to explicitly specify a VRF as the source MVRF.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and enters privileged EXEC mode.
Step 5	**show ip mroute** [**vrf** *vrf-name*] *group-address* **Example:** `Router# show ip mroute 224.100.0.5`	(Optional) Displays the contents of the IP multicast mroute table for a specific group address.

Configuring Group-Based VRF Selection Policies with MVPN

Perform this task to configure group-based VRF selection policies with MVPN .

This task enables RPF lookups to be performed to the same source address in different VRFs using the group address as the VRF selector. This feature enhances extranet MVPNs by enabling service providers to distribute content streams coming in from different MVPNs and redistributing them from there.

Before you begin

- You must configure support for extranet MVPN services prior to performing this task.
- ACLs are used to define the groups to be applied to group-based VRF selection policies. This task assumes that you have configured the ACLs to be applied to group-based VRF selection policies.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Use the following commands:
   - **ip multicast** [**vrf** *receiver-vrf-name*] **rpf select** {**global** | **vrf** *source-vrf-name*} **group-list** *access-list*
4. Repeat step 3 to create additional group-based VRF selection policies.
5. **end**
6. Use the following commands:
   - **show ip**} **rpf** [**vrf** *vrf-name*] **select**
7. Use one of the following commands:
   - **show ip rpf** [**vrf** *vrf-name*] *source-address* [*group-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Use the following commands: • **ip multicast** [**vrf** *receiver-vrf-name*] **rpf select** {**global** | **vrf** *source-vrf-name*} **group-list** *access-list* **Example:** `Router(config)# ip multicast vrf VPN-Green rpf select vrf VPN-Red group-list 1`	• (For IPv4 only) Configures RPF lookups originating in a receiver MVRF or in the global routing table to be resolved in a source MVRF or in the global routing table based on group address.
Step 4	Repeat step 3 to create additional group-based VRF selection policies.	--

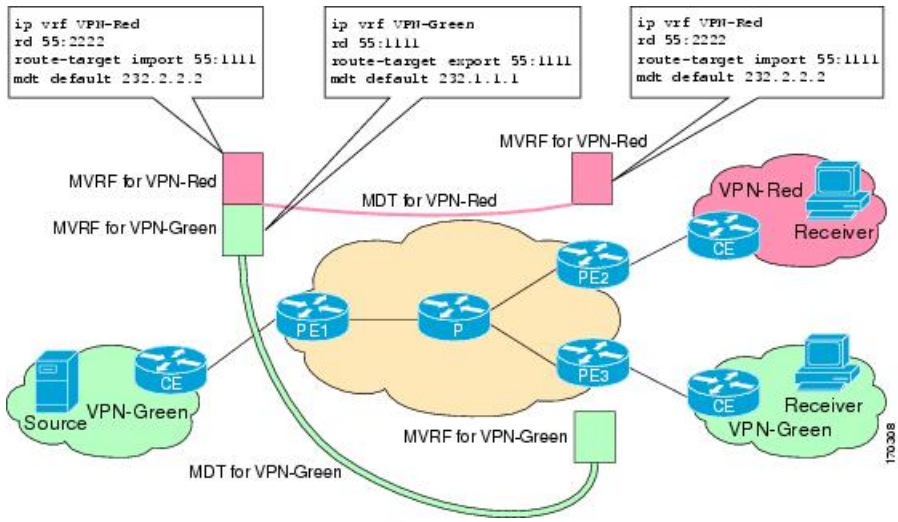
	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and enters privileged EXEC mode.
Step 6	Use the following commands: • **show ip**} **rpf** [**vrf** *vrf-name*] **select** **Example:** `Router# show ip rpf select`	Displays group-to-VRF mapping information.
Step 7	Use one of the following commands: • **show ip rpf** [**vrf** *vrf-name*] *source-address* [*group-address*] **Example:** `Router# show ip rpf 172.16.10.13`	Displays information about how IP multicast routing does RPF. • Use this command after configuring group-based VRF selection policies to confirm that RPF lookups are being performed based on the group address, and to display the VRF where the RPF lookup is being performed.

Configuration Examples for Multicast VPN Extranet Support

Example Configuring the Receiver VRF on the Source PE Router - Option 1 (SSC)

The following example shows the configurations for PE1, the source PE router, and PE2, the receiver PE router, in the figure. In this example, extranet MVPN services are supported between VPN-Green and VPN-Red by configuring the receiver MVRF for VPN-Red on PE1, the source PE router. The MVRF configuration for VPN-Red is configured to import routes from the MVRF for VPN-Green to the MVRF for VPN-Red.

Figure 14: Topology for MVPN Extranet Support Option 1 Configuration Example



PE1 Configuration

```
ip cef
!ip vrf VPN-Green
rd 55:1111
route-target export 55:1111
route-target import 55:1111
mdt default mpls mldp 20.100.0.4
!
ip vrf VPN-Red
rd 55:2222
route-target export 55:2222
route-target import 55:2222
route-target import 55:1111
mdt default mpls mldp 20.100.0.4
!
!
ip multicast-routing
ip multicast-routing vrf VPN-Green
ip multicast-routing vrf VPN-Red
!
interface Loopback0
ip address 10.1.0.1 255.255.255.0
ip pim sparse-dense-mode
!
.
.
.
!
router bgp 55
no synchronization
bgp log-neighbor-changes
neighbor 10.2.0.2 remote-as 55
neighbor 10.2.0.2 update-source Loopback0

!
address-family ipv4 mdt
neighbor 10.2.0.2 activate
neighbor 10.2.0.2 send-community extended
```

```
!
address-family vpnv4
neighbor 10.2.0.2 activate
neighbor 10.2.0.2 send-community extended
!
 address-family ipv4
  neighbor 10.2.0.2 activate
  neighbor 10.2.0.2 send-community both
 exit-address-family
 !
 address-family ipv4 vrf VPN-Green
  redistribute connected
  redistribute static
  neighbor 10.2.0.2 remote-as 100
  neighbor 10.2.0.2 activate
 exit-address-family
!
 address-family ipv4 vrf VPN-Red
  redistribute connected
  redistribute static
  neighbor 10.2.0.2 remote-as 100
  neighbor 10.2.0.2 activate
 exit-address-family
!
```

PE2 Configuration

```
!
ip vrf VPN-Red
rd 55:2222
route-target export 55:2222
route-target import 55:2222
route-target import 55:1111
mdt default mpls mldp 20.100.0.4
!
ip multicast-routing
ip multicast-routing vrf VPN-Red
!
interface Loopback0
ip address 10.2.0.2 255.255.255.0
ip pim sparse-dense-mode
!
.
.
.
!
router bgp 55
no synchronization
bgp log-neighbor-changes
neighbor 10.1.0.1 remote-as 55
neighbor 10.1.0.1 update-source Loopback0
!
address-family ipv4 mdt
neighbor 10.1.0.1 activate
neighbor 10.1.0.1 send-community extended
!
address-family vpnv4
neighbor 10.1.0.1 activate
neighbor 10.1.0.1 send-community extended
!
!
 address-family ipv4
  neighbor 10.1.0.1 activate
```

```
  neighbor 10.1.0.1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf VPN-Red
  redistribute connected
  redistribute static
  neighbor 10.1.0.1 remote-as 100
  neighbor 10.1.0.1 activate
 exit-address-family
!
ip pim vrf blue1 rp-address 55.55.55.55
ip pim vrf red1 rp-address 55.55.55.55
ip mroute vrf red1 40.0.0.0 255.255.255.0 fallback-lookup vrf blue1
ip mroute vrf red1 55.55.55.55 255.255.255.255 fallback-lookup vrf blue1
```

States in the Global Table on PE1 and PE2 for the MDT Default Group 232.3.3.3

The following are sample outputs from the **show ip mroute** command on PE1 and PE2. The sample outputs show the global table for the MDT default group 232.3.3.3 on PE1 and PE2.

```
PE1# show ip mroute 232.3.3.3
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.1.0.1, 232.3.3.3), 00:46:27/00:03:27, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse-Dense, 00:45:17/00:02:44
(10.2.0.2, 232.3.3.3), 00:45:17/00:02:57, flags: sTIZ
  Incoming interface: Ethernet0/0, RPF nbr 224.0.1.4
  Outgoing interface list:
    MVRF VPN-Red, Forward/Sparse-Dense, 00:45:17/00:01:09
PE2# show ip mroute 232.3.3.3
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.1.0.1, 232.3.3.3), 00:45:08/00:02:37, flags: sTIZ
  Incoming interface: Ethernet1/0, RPF nbr 224.0.2.4
  Outgoing interface list:
    MVRF VPN-Red, Forward/Sparse-Dense, 00:45:08/00:01:27
(10.2.0.2, 232.3.3.3), 00:46:19/00:03:07, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:45:08/00:02:49
```

States in the Global Table on PE1 and PE2 for the MDT Default Group 232.3.3.3 When PE1 and PE2 are Configured for MVPN Extranet Support

The following are sample outputs from the **show ip mroute** on PE1 and PE2, when PE1 and PE2 are configured to support extranet MVPN services. The sample output from the **show ip mroute** command shows the global table for the MDT default group 232.3.3.3 on PE1 and PE2. In the output, the "RPF-MFD" flag indicates that a multicast flow is completely hardware switched and "H" flag indicates that the flow is being hardware switched on an outgoing interface.

```
PE1# show ip mroute 232.3.3.3
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.1.0.1, 232.3.3.3), 00:46:27/00:03:27, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0, RPF-MFD
  Outgoing interface list:
    GigabitEthernet2/16, Forward/Sparse-Dense, 00:45:17/00:02:44, H
(10.2.0.2, 232.3.3.3), 00:45:17/00:02:57, flags: sTIZ
  Incoming interface: GigabitEthernet2/16, RPF nbr 224.0.1.4, RPF-MFD
  Outgoing interface list:
    MVRF VPN-Red, Forward/Sparse-Dense, 00:45:17/00:01:09, H


PE2# show ip mroute 232.3.3.3
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.1.0.1, 232.3.3.3), 00:45:08/00:02:37, flags: sTIZ
  Incoming interface: GigabitEthernet4/1, RPF nbr 224.0.2.4, RPF-MFD
  Outgoing interface list:
    MVRF VPN-Red, Forward/Sparse-Dense, 00:45:08/00:01:27, H
(10.2.0.2, 232.3.3.3), 00:46:19/00:03:07, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0, RPF-MFD
  Outgoing interface list:
    GigabitEthernet4/1, Forward/Sparse-Dense, 00:45:08/00:02:49, H
```

States in the VRF Table for VPN-Green on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE1. The sample output shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8. The output indicates that extranet receivers in VPN-Red are receiving content from a source in VPN-Green that is sending to multicast group 228.8.8.8. The "E" flag in the output indicates that a (*, G) or (S, G) entry in the VRF routing table is a source VRF entry and has extranet receiver MVRF mroute entries linked to it.

```
PE1# show ip mroute vrf VPN-Green 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:38/stopped, RP 10.100.0.5, flags: SE
  Incoming interface: Ethernet3/0, RPF nbr 10.1.1.5
  Outgoing interface list: Null
  Extranet receivers in vrf VPN-Red:
(*, 228.8.8.8), 00:01:38/stopped, RP 10.100.0.5, OIF count: 1, flags: S
(10.1.1.200, 228.8.8.8), 00:00:05/00:02:54, flags: TE
  Incoming interface: Ethernet3/0, RPF nbr 10.1.1.5
  Outgoing interface list: Null
  Extranet receivers in vrf VPN-Red:
  (10.1.1.200, 228.8.8.8), 00:00:05/stopped, OIF count: 1, flags:
```

States in the VRF Table for VPN-Green on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8 When PE1 is configured for MVPN Extranet Support

The following are sample outputs from the **show ip mroute** on PE1, when PE1 is configured to support extranet MVPN services. The sample output from the **show ip mroute** command shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8. The sample output indicate that extranet receivers in VPN-Red are receiving content from a source in VPN-Green that is sending to multicast group 228.8.8.8.

```
PE1# show ip mroute vrf VPN-Green 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:38/stopped, RP 10.100.0.5, flags: SE
  Incoming interface: GigabitEthernet3/1, RPF nbr 10.1.1.5, RPF-MFD
  Outgoing interface list: Null
  Extranet receivers in vrf VPN-Red:
(*, 228.8.8.8), 00:01:38/stopped, RP 10.100.0.5, OIF count: 1, flags: S
(10.1.1.200, 228.8.8.8), 00:00:05/00:02:54, flags: TE
  Incoming interface: GigabitEthernet3/1, RPF nbr 10.1.1.5, RPF-MFD
  Outgoing interface list: Null
  Extranet receivers in vrf VPN-Red:
  (10.1.1.200, 228.8.8.8), 00:00:05/stopped, OIF count: 1, flags:
```

States in the VRF Table for VPN-Red on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE1. The sample output shows the state of the VRF table for VPN-Red on PE1 when receivers join the multicast group 228.8.8.8. The "using vrf VPN-Green" field indicates that VPN-Red is using unicast routing information from VPN-Green to determine the RPF interface through which the source is reachable.

```
PE1# show ip mroute vrf VPN-Red 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:45/stopped, RP 10.100.0.5, flags: S
  Incoming interface: Ethernet3/0, RPF nbr 10.1.1.5, using vrf VPN-Green
  Outgoing interface list:
    Tunnel2, Forward/Sparse-Dense, 00:01:45/00:02:49
(10.1.1.200, 228.8.8.8), 00:00:12/00:03:27, flags:
  Incoming interface: Ethernet3/0, RPF nbr 10.1.1.5, using vrf VPN-Green
  Outgoing interface list:
    Tunnel2, Forward/Sparse-Dense, 00:00:12/00:03:18
```

States in the VRF Table for VPN-Red on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8 When PE1 is configured for MVPN Extranet Support

The following is a sample output from the **show ip mroute** command on PE1, when PE1 configured to support extranet MVPN services. The sample output from the **show ip mroute** command shows the state of the VRF table for VPN-Red on PE1 when receivers join the multicast group 228.8.8.8. The "using vrf VPN-Green" field indicates that VPN-Red is using unicast routing information from VPN-Green to determine the RPF interface through which the source is reachable.

```
PE1# show ip mroute vrf VPN-Red 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:45/stopped, RP 10.100.0.5, flags: S
  Incoming interface: GigabitEthernet3/1, RPF nbr 10.1.1.5, using vrf VPN-Green, RPF-MFD
  Outgoing interface list:
    Tunnel2, Forward/Sparse-Dense, 00:01:45/00:02:49, H
(10.1.1.200, 228.8.8.8), 00:00:12/00:03:27, flags:
  Incoming interface: GigabitEthernet3/1, RPF nbr 10.1.1.5, using vrf VPN-Green, RPF-MFD
  Outgoing interface list:
    Tunnel2, Forward/Sparse-Dense, 00:00:12/00:03:18, H
```

States in the VRF Table for VPN-Red on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE2. The sample output shows the VRF table for VPN-Red on PE2 when receivers join the multicast group 228.8.8.8.

```
PE2# show ip mroute vrf VPN-Red 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:00:28/stopped, RP 10.100.0.5, flags: S
  Incoming interface: Tunnel1, RPF nbr 10.1.0.1
  Outgoing interface list:
    Ethernet9/0, Forward/Sparse-Dense, 00:00:28/00:03:02
(10.1.1.200, 228.8.8.8), 00:00:00/00:03:29, flags:
  Incoming interface: Tunnel1, RPF nbr 10.1.0.1
  Outgoing interface list:
    Ethernet9/0, Forward/Sparse-Dense, 00:00:00/00:03:29
```

States in the VRF Table for VPN-Red on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8 When PE2 is configured for MVPN Extranet Support

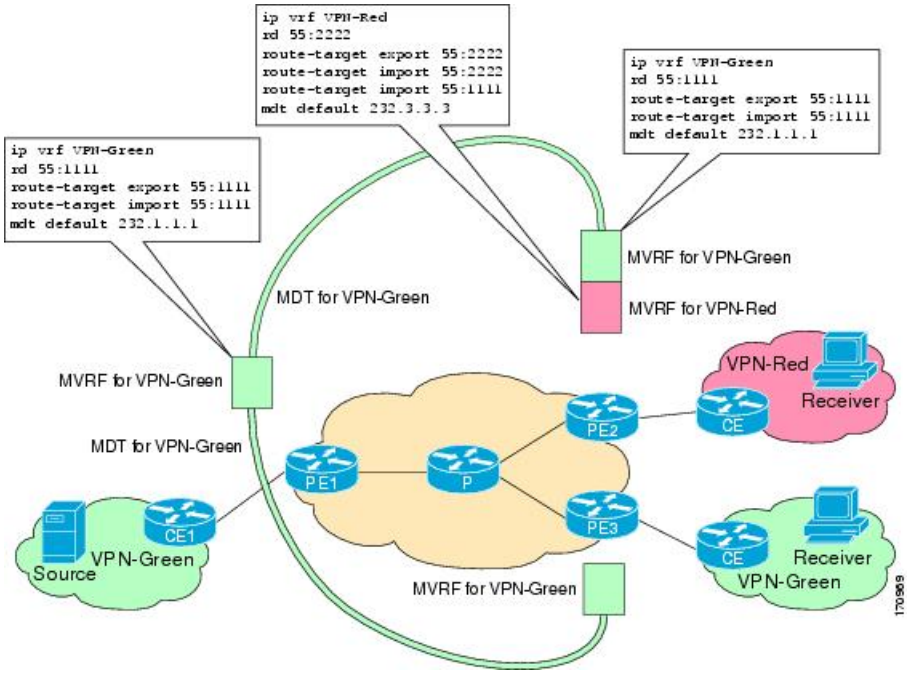
The following are sample output from the **show ip mroute** command on PE2, when PE2 is configured to support extranet MVPN services. The sample output from the **show ip mroute** command shows the VRF table for VPN-Red on PE2 when receivers join the multicast group 228.8.8.8.

```
PE2# show ip mroute vrf VPN-Red 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:00:28/stopped, RP 10.100.0.5, flags: S
  Incoming interface: Tunnel1, RPF nbr 10.1.0.1, RPF-MFD
  Outgoing interface list:
    GigabitEthernet9/1, Forward/Sparse-Dense, 00:00:28/00:03:02, H
(10.1.1.200, 228.8.8.8), 00:00:00/00:03:29, flags:
  Incoming interface: Tunnel1, RPF nbr 10.1.0.1, RPF-MFD
  Outgoing interface list:
    GigabitEthernet9/1, Forward/Sparse-Dense, 00:00:00/00:03:29, H
```

Example Configuring the Source VRF on the Receiver PE - Option 2 (RSC)

The following configuration example is based on the extranet MVPN topology illustrated in the figure. This example shows the configurations for PE2, the receiver PE router, and PE1, the source PE router. In this example, extranet MVPN services are supported between VPN-Green and VPN-Red by configuring the source MVRF for VPN-Green on PE2. The same unicast routing policy is configured to import routes from VPN-Green to VPN-Red.

Figure 15: Topology for MVPN Extranet Support Option 2 Configuration Example



PE2 Configuration

```
!
ip vrf VPN-Green
rd 55:1111
route-target export 55:1111
route-target import 55:1111
mdt default mpls mldp 20.100.0.4
!
ip vrf VPN-Red
rd 55:2222
route-target export 55:2222
route-target import 55:2222
route-target import 55:1111
mdt default mpls mldp 20.100.0.4
!
!
ip multicast-routing
ip multicast-routing vrf VPN-Green
ip multicast-routing vrf VPN-Red
!
interface Loopback0
ip address 10.1.0.1 255.255.255.0
ip pim sparse-dense-mode
```

```
!
.
.
.
!
router bgp 55
no synchronization
bgp log-neighbor-changes
neighbor 10.2.0.2 remote-as 55
neighbor 10.2.0.2 update-source Loopback0

!
address-family ipv4 mdt
neighbor 10.2.0.2 activate
neighbor 10.2.0.2 send-community extended
!
address-family vpnv4
neighbor 10.2.0.2 activate
neighbor 10.2.0.2 send-community extended
!
 address-family ipv4
  neighbor 10.2.0.2 activate
  neighbor 10.2.0.2 send-community both
 exit-address-family
 !
 address-family ipv4 vrf VPN-Green
  redistribute connected
  redistribute static
  neighbor 10.2.0.2 remote-as 100
  neighbor 10.2.0.2 activate
 exit-address-family
!
 address-family ipv4 vrf VPN-Red
  redistribute connected
  redistribute static
  neighbor 10.2.0.2 remote-as 100
  neighbor 10.2.0.2 activate
 exit-address-family
!
ip mroute vrf red1 40.0.0.0 255.255.255.0 fallback-lookup vrf VPN-Green
ip mroute vrf red1 55.55.55.55 255.255.255.255 fallback-lookup vrf VPN-Green
```

PE1 Configuration

```
!
ip vrf VPN-Red
rd 55:2222
route-target export 55:2222
route-target import 55:2222
route-target import 55:1111
mdt default mpls mldp 20.100.0.4
!
ip multicast-routing
ip multicast-routing vrf VPN-Red
!
interface Loopback0
ip address 10.2.0.2 255.255.255.0
ip pim sparse-dense-mode
!
.
.
.
!
```

```
router bgp 55
no synchronization
bgp log-neighbor-changes
neighbor 10.1.0.1 remote-as 55
neighbor 10.1.0.1 update-source Loopback0
!
address-family ipv4 mdt
neighbor 10.1.0.1 activate
neighbor 10.1.0.1 send-community extended
!
address-family vpnv4
neighbor 10.1.0.1 activate
neighbor 10.1.0.1 send-community extended
!
!
 address-family ipv4
  neighbor 10.1.0.1 activate
  neighbor 10.1.0.1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf VPN-Red
  redistribute connected
  redistribute static
  neighbor 10.1.0.1 remote-as 100
  neighbor 10.1.0.1 activate
 exit-address-family
!
ip pim vrf red1 rp-address 55.55.55.55
```

States in the Global Table on PE1 and PE2 for the MDT Default Group 232.1.1.1

The following are sample outputs from the **show ip mroute** command on PE1 and PE2. The sample outputs show the global table for the MDT default group 232.1.1.1 on PE1 and PE2.

```
PE1# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.2.0.2, 232.1.1.1), 00:01:19/00:02:42, flags: sTIZ
  Incoming interface: Ethernet0/0, RPF nbr 10.0.1.4
  Outgoing interface list:
    MVRF VPN-Green, Forward/Sparse-Dense, 00:01:19/00:02:07
(10.1.0.1, 232.1.1.1), 00:02:19/00:03:11, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse-Dense, 00:02:00/00:02:36
PE2# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
```

```
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.1.0.1, 232.1.1.1), 00:02:04/00:02:38, flags: sTIZ
  Incoming interface: Ethernet1/0, RPF nbr 10.0.2.4
  Outgoing interface list:
    MVRF VPN-Green, Forward/Sparse-Dense, 00:02:04/00:02:09
(10.2.0.2, 232.1.1.1), 00:02:04/00:03:09, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/0, Forward/Sparse-Dense, 00:01:22/00:03:09
```

States in the Global Table on PE1 and PE2 for the MDT Default Group 232.1.1.1 When PE1 and PE2 for MVPN Extranet Support

The following are sample output from the **show ip mroute** command on PE1 and PE2, when PE1 and PE2 are configured to support extranet MVPN services. The sample output from the **show ip mroute** command shows the global table for the MDT default group 232.1.1.1 on PE1 and PE2. In the output, the “RPF-MFD” flag indicates that a multicast flow is completely hardware switched and “H” flag indicates that the flow is being hardware switched on an outgoing interface.

```
PE1# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.2.0.2, 232.1.1.1), 00:01:19/00:02:42, flags: sTIZ
  Incoming interface: GigabitEthernet2/16, RPF nbr 10.0.1.4, RPF-MFD
  Outgoing interface list:
    MVRF VPN-Green, Forward/Sparse-Dense, 00:01:19/00:02:07, H
(10.1.0.1, 232.1.1.1), 00:02:19/00:03:11, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0, RPF-MFD
  Outgoing interface list:
    GigabitEthernet2/16, Forward/Sparse-Dense, 00:02:00/00:02:36, H
PE2# show ip mroute 232.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(10.1.0.1, 232.1.1.1), 00:02:04/00:02:38, flags: sTIZ
  Incoming interface: GigabitEthernet4/1, RPF nbr 10.0.2.4, RPF-MFD
  Outgoing interface list:
    MVRF VPN-Green, Forward/Sparse-Dense, 00:02:04/00:02:09, H
(10.2.0.2, 232.1.1.1), 00:02:04/00:03:09, flags: sT
```

```
  Incoming interface: Loopback0, RPF nbr 0.0.0.0, RPF-MFD
  Outgoing interface list:
    GigabitEthernet4/1, Forward/Sparse-Dense, 00:01:22/00:03:09, H
```

States in the VRF Table for VPN-Green on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE1. The sample output shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8.

```
PE1# show ip mroute vrf VPN-Green 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:43/00:02:52, RP 10.100.0.5, flags: S
  Incoming interface: Ethernet3/0, RPF nbr 10.1.1.5
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:01:43/00:02:52
(10.1.1.200, 228.8.8.8), 00:01:15/00:03:26, flags: T
  Incoming interface: Ethernet3/0, RPF nbr 10.1.1.5
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:01:15/00:03:19
```

States in the VRF Table for VPN-Green on PE1 After Receivers in VPN-Red Join Multicast Group 228.8.8.8 When PE1 Configured for MVPN Extranet Support

The following are sample output from the **show ip mroute** commands on PE1, when PE1 is configured to support extranet MVPN services. The sample output from the **show ip mroute** command shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8.

```
PE1# show ip mroute vrf VPN-Green 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:43/00:02:52, RP 10.100.0.5, flags: S
  Incoming interface: GigabitEthernet3/1, RPF nbr 10.1.1.5, RPF-MFD
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:01:43/00:02:52, H
(10.1.1.200, 228.8.8.8), 00:01:15/00:03:26, flags: T
  Incoming interface: GigabitEthernet3/1, RPF nbr 10.1.1.5, RPF-MFD
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:01:15/00:03:19, H
```

States in the VRF Table for VPN-Green on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE2. The output shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8. The output indicates that extranet receivers in VPN-Red are receiving content from the source in VPN-Green that is sending to multicast group 228.8.8.8. The “E” flag indicates that a (*, G) or (S, G) entry in the VRF routing table is a source VRF entry and has extranet receiver MVRF mroute entries linked to it.

```
PE2# show ip mroute vrf VPN-Green 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:59/stopped, RP 10.100.0.5, flags: SE
  Incoming interface: Tunnel0, RPF nbr 10.1.0.1
  Outgoing interface list: Null
  Extranet receivers in vrf VPN-Red:
(*, 228.8.8.8), 00:01:59/stopped, RP 10.100.0.5, OIF count: 1, flags: S
(10.1.1.200, 228.8.8.8), 00:01:31/00:02:59, flags: TE
  Incoming interface: Tunnel0, RPF nbr 10.1.0.1
  Outgoing interface list: Null
  Extranet receivers in vrf VPN-Red:
  (10.1.1.200, 228.8.8.8), 00:01:31/00:03:29, OIF count: 1, flags:
```

States in the VRF Table for VPN-Green on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8 When PE2 is Configured for MVPN Extranet Support

The following are sample output from the **show ip mroute** command on PE2, when PE2 is configured to support extranet MVPN services. The sample output from the **show ip mroute** command shows the state of the VRF table for VPN-Green on PE1 when receivers join the multicast group 228.8.8.8. The sample output indicates that extranet receivers in VPN-Red are receiving content from the source in VPN-Green that is sending to multicast group 228.8.8.8. The “E” flag indicates that a (*, G) or (S, G) entry in the VRF routing table is a source VRF entry and has extranet receiver MVRF mroute entries linked to it.

```
PE2# show ip mroute vrf VPN-Green 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:01:59/stopped, RP 10.100.0.5, flags: SE
  Incoming interface: Tunnel0, RPF nbr 10.1.0.1, RPF-MFD
  Outgoing interface list: Null
  Extranet receivers in vrf VPN-Red:
(*, 228.8.8.8), 00:01:59/stopped, RP 10.100.0.5, OIF count: 1, flags: S
```

```
(10.1.1.200, 228.8.8.8), 00:01:31/00:02:59, flags: TE
  Incoming interface: Tunnel0, RPF nbr 10.1.0.1, RPF-MFD
  Outgoing interface list: Null
  Extranet receivers in vrf VPN-Red:
  (10.1.1.200, 228.8.8.8), 00:01:31/00:03:29, OIF count: 1, flags:
```

States in the VRF Table for VPN-Red on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8

The following is sample output from the **show ip mroute** command on PE2. The sample output shows the state of the VRF table for VPN-Red on PE2 when receivers join the multicast group 228.8.8.8. The "using vrf VPN-Green" field indicates that VPN-Red is using unicast routing information from VPN-Green to determine the RPF interface through which the source is reachable.

```
PE2# show ip mroute vrf VPN-Red 228.8.8.8

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:02:00/stopped, RP 10.100.0.5, flags: S
  Incoming interface: Tunnel0, RPF nbr 10.1.0.1, using vrf VPN-Green
  Outgoing interface list:
    Ethernet9/0, Forward/Sparse-Dense, 00:02:00/00:02:34
(10.1.1.200, 228.8.8.8), 00:01:32/00:03:28, flags:
  Incoming interface: Tunnel0, RPF nbr 10.1.0.1, using vrf VPN-Green
  Outgoing interface list:
    Ethernet9/0, Forward/Sparse-Dense, 00:01:32/00:03:01
```

States in the VRF Table for VPN-Red on PE2 After Receivers in VPN-Red Join Multicast Group 228.8.8.8 When PE2 is configured for MVPN Extranet Support

The following are sample output from the **show ip mroute** command on PE2, when PE2 is configured to support extranet MVPN services. The sample output from the **show ip mroute** command shows the state of the VRF table for VPN-Red on PE2 when receivers join the multicast group 228.8.8.8. The "using vrf VPN-Green" field indicates that VPN-Red is using unicast routing information from VPN-Green to determine the RPF interface through which the source is reachable.

```
PE2# show ip mroute vrf VPN-Red 228.8.8.8
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.8.8.8), 00:02:00/stopped, RP 10.100.0.5, flags: S
  Incoming interface: Tunnel0, RPF nbr 10.1.0.1, using vrf VPN-Green, RPF-MFD
```

```
  Outgoing interface list:
    GigabitEthernet9/1, Forward/Sparse-Dense, 00:02:00/00:02:34, H
(10.1.1.200, 228.8.8.8), 00:01:32/00:03:28, flags:
  Incoming interface: Tunnel0, RPF nbr 10.1.0.1, using vrf VPN-Green, RPF-MFD
  Outgoing interface list:
    GigabitEthernet9/1, Forward/Sparse-Dense, 00:01:32/00:03:01, H
```

Example: Displaying Statistics for MVPN Extranet Support

This example is a stand alone example and does not refer to any other technologies.

The MFIB-based implementation of IP multicast updates counters in source MVRF mroute entries for extranet MVPN. Counters in the source MVRF can be displayed using Cisco IOS commands. Counters in the receiver MVRF mroute entries will remain zero.

Use the **show ip mroute** command to determine the source and receiver MVRFs. The following sample output shows that VRF blue is the source MVRF and VRF red is the receiver MVRF:

```
PE1# show ip mroute vrf blue 228.1.1.1

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.1.1.1), 00:05:48/stopped, RP 202.100.0.5, flags: SE
  Incoming interface: Ethernet3/0, RPF nbr 200.1.1.5
  Outgoing interface list: Null
  Extranet receivers in vrf red:
(*, 228.1.1.1), 00:05:48/stopped, RP 202.100.0.5, OIF count: 1, flags: S
(220.1.1.200, 228.1.1.1), 00:02:42/00:02:09, flags: TE
  Incoming interface: Ethernet3/0, RPF nbr 200.1.1.5
  Outgoing interface list: Null
  Extranet receivers in vrf red:
  (220.1.1.200, 228.1.1.1), 00:02:42/stopped, OIF count: 1, flags: T


PE1# show ip mroute vrf red 228.1.1.1

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.1.1.1), 00:05:55/stopped, RP 202.100.0.5, flags: S
  Incoming interface: Ethernet3/0, RPF nbr 200.1.1.5, using vrf blue
  Outgoing interface list:
    Tunnel16, Forward/Sparse-Dense, 00:05:55/00:03:26
```

```
(220.1.1.200, 228.1.1.1), 00:02:49/stopped, flags: T
  Incoming interface: Ethernet3/0, RPF nbr 200.1.1.5, using vrf blue
  Outgoing interface list:
    Tunnel16, Forward/Sparse-Dense, 00:02:49/00:03:26
```

Use the **show ip mfib vrf** *vrf-name* command, with the source MVRF for the *vrf-name* argument, to display statistics.

The following example shows statistics for the source MVRF blue. Inspect the output to ensure that the forwarding statistics in the source MVRF MFIB are correct and that the A and F flags are set in the source MVRF. Notice that there is no indication of extranet forwarding in the MFIB.

```
PE1# show ip mfib vrf blue 228.1.1.1

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
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
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 blue
(*,228.1.1.1) Flags: C
  SW Forwarding: 1/0/100/0, Other: 0/0/0
  Ethernet3/0 Flags: A
  Tunnel16, MDT/239.3.3.3 Flags: F
    Pkts: 1/0
(220.1.1.200,228.1.1.1) Flags:
  SW Forwarding: 37/0/100/0, Other: 0/0/0
  Ethernet3/0 Flags: A NS
  Tunnel16, MDT/239.3.3.3 Flags: F
    Pkts: 37/0
```

The following example shows the following information for the receiver MVRF red:

- There are no forwarding statistics in the receiver MVRF MFIB because these statistics are collected in the source MVRF.
- The A and F flags are not set because these flags are only set in the source MVRF for MVPN extranet.
- There is no indication of extranet forwarding in the MFIB.

Note The NS flag in the output is present for the purpose of receiving PIM control traffic in the receiver MVRF.

```
PE1# show ip mfib vrf red 228.1.1.1

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
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
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 red
(*,228.1.1.1) Flags: C
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel16, MDT/239.3.3.3 Flags: NS
(220.1.1.200,228.1.1.1) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnel16, MDT/239.3.3.3 Flags: NS
```

You can also use the **show ip mroute count** command to display the extranet MVPN statistics. However, we recommend that you use the **show ip mfib** command instead. If you use the **show ip mroute count**command to display statistics, inspect the output to ensure that the forwarding statistics in the source MVRF are correct and that there are no forwarding statistics in the receiver MVRF.

The following sample output from the **show ip mroute count** command shows statistics for the source MVRF blue:

```
PE1# show ip mroute vrf blue 228.1.1.1 count

Use "show ip mfib count" to get better response time for a large number of
mroutes.

IP Multicast Statistics
3 routes using 1354 bytes of memory
2 groups, 0.50 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: 228.1.1.1, Source count: 1, Packets forwarded: 38, Packets received: 38
  RP-tree: Forwarding: 1/0/100/0, Other: 1/0/0
  Source: 220.1.1.200/32, Forwarding: 37/0/100/0, Other: 37/0/0
```

The following sample output from the **show ip mroute count** command is for the receiver MVRF red:

```
PE1# show ip mroute vrf red 228.1.1.1 count

Use "show ip mfib count" to get better response time for a large number of
mroutes.

IP Multicast Statistics
3 routes using 1672 bytes of memory
2 groups, 0.50 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: 228.1.1.1, Source count: 1, Packets forwarded: 0, Packets received: 0
  RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0
  Source: 220.1.1.200/32, Forwarding: 0/0/0/0, Other: 0/0/0
```

Example Configuring RPF for MVPN Extranet Support Using Static Mroutes

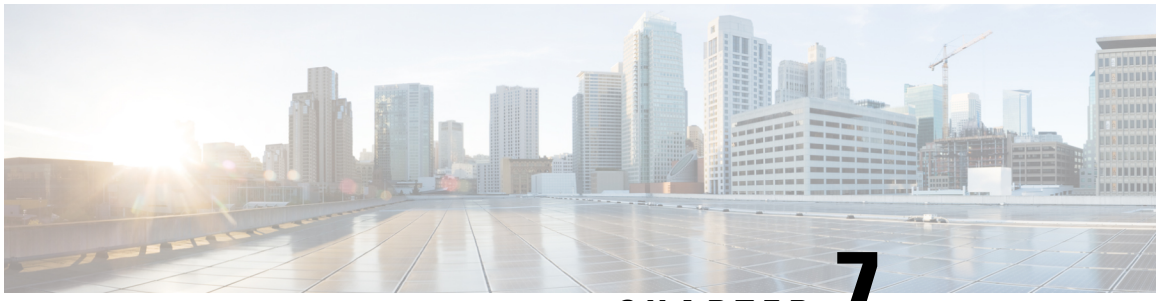
The following example shows how to configure the RPF lookup originating in VPN-Red to be resolved in VPN-Green using the static mroute 192.168.1.1:

```
ip mroute vrf VPN-Red 192.168.1.1 255.255.255.255 fallback-lookup vrf VPN-Green
```

Example Configuring Group-Based VRF Selection Policies with MVPN Extranet Support

The following example shows how to use group-based VRF selection policies to configure RPF lookups originating in VPN-Green to be performed in VPN-Red for group addresses that match ACL 1 and to be performed in VPN-Blue for group addresses that match ACL 2.

```
ip multicast vrf VPN-Green rpf select vrf VPN-Red group-list 1
ip multicast vrf VPN-Green rpf select vrf VPN-Blue group-list 2
!
.
.
.
!
access-list 1 permit 239.0.0.0 0.255.255.255
access-list 2 permit 238.0.0.0 0.255.255.255
!
```



CHAPTER 7

IPv6 Multicast PIM

This module describes how to configure IPv6 Multicast PIM features.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for IPv6 Multicast

The following are the prerequisites for IPv6 PIM source-specific multicast (SSM):

- Multicast Listener Discovery (MLD) version 2 is required for source-specific multicast (SSM) to operate.
- Before configuring SSM with MLD, SSM must be supported by the Cisco IPv6 device, the host where the application is running, and the application itself.

Restrictions for IPv6 Multicast

The supported IPv6 scale for the default templates for multicast routes is 500.

The router does *not* support :

- IPv6 Multicast Listener Discovery (MLD) snooping
- Bootstrap Router (BSR)
- Multiprotocol BGP Extensions for multicast
- Equal Cost Multipath (ECMP) for multicast
- Multicast Virtual Private Network (MVPN)
- Multicast QoS
- Dynamic Domain Name System (DNS) PIM Source Specific Multicast (SSM) mapping for multicast
- Multicast Source Discovery Protocol (MSDP)

Note The Multicast control packets are *not* processed when the system memory utilization is more than 90 %. The following message is displayed on the console.

```
*Sep 18 18:21:07.287: %SYS-2-NOMEMORY: No memory available for multicast control packets,
dropping multicast control packets.
  Memory usage percentage:  91
```

The system memory utilization may increase when the number of multicast sources and IGMP reports join rate is increased. When 90% of the system memory is used, the IGMP reports are *not* processed and multicast may not function as expected. For the multicast reports to be processed again, decrease the join rate.

Information About IPv6 Multicast

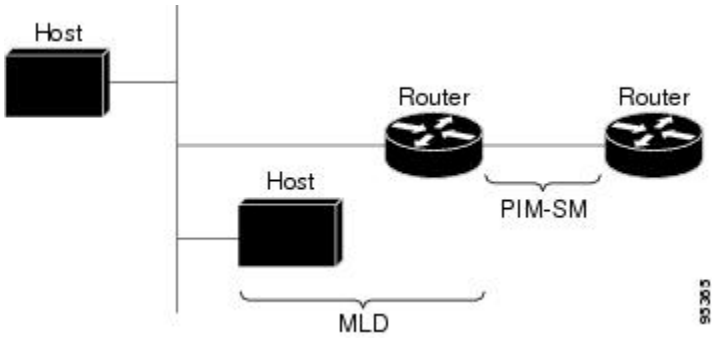
IPv6 Multicast Routing Implementation

Cisco software supports the following protocols to implement IPv6 multicast routing:

- MLD is used by IPv6 devices to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. There are two versions of MLD:
  - MLD version 1 is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4.
  - MLD version 2 is based on version 3 of the IGMP for IPv4.
- IPv6 multicast for Cisco software uses both MLD version 2 and MLD version 1. MLD version 2 is fully backward-compatible with MLD version 1 (described in RFC 2710). Hosts that support only MLD version 1 will interoperate with a device running MLD version 2. Mixed LANs with both MLD version 1 and MLD version 2 hosts are likewise supported.
- PIM-SM is used between devices so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- PIM in Source Specific Multicast (PIM-SSM) is similar to PIM-SM with the additional ability to report interest in receiving packets from specific source addresses (or from all but the specific source addresses) to an IP multicast address.

The figure below shows where MLD and PIM-SM operate within the IPv6 multicast environment.

Figure 16: IPv6 Multicast Routing Protocols Supported for IPv6



Protocol Independent Multicast

Protocol Independent Multicast (PIM) is used between devices so that they can track which multicast packets to forward to each other and to their directly connected LANs. PIM works independently of the unicast routing protocol to perform send or receive multicast route updates like other protocols. Regardless of which unicast routing protocols are being used in the LAN to populate the unicast routing table, Cisco IOS PIM uses the existing unicast table content to perform the Reverse Path Forwarding (RPF) check instead of building and maintaining its own separate routing table.

You can configure IPv6 multicast to use either a PIM- Sparse Mode (SM) or PIM-Source Specific Multicast (SSM) operation, or you can use both PIM-SM and PIM-SSM together in your network.

PIM-Sparse Mode

IPv6 multicast provides support for intradomain multicast routing using PIM-SM. PIM-SM uses unicast routing to provide reverse-path information for multicast tree building, but it is not dependent on any particular unicast routing protocol.

PIM-SM is used in a multicast network when relatively few devices are involved in each multicast and these devices do not forward multicast packets for a group, unless there is an explicit request for the traffic. PIM-SM distributes information about active sources by forwarding data packets on the shared tree. PIM-SM initially uses shared trees, which requires the use of an RP.

Requests are accomplished via PIM joins, which are sent hop by hop toward the root node of the tree. The root node of a tree in PIM-SM is the RP in the case of a shared tree or the first-hop device that is directly connected to the multicast source in the case of a shortest path tree (SPT). The RP keeps track of multicast groups and the hosts that send multicast packets are registered with the RP by that host's first-hop device.

As a PIM join travels up the tree, devices along the path set up multicast forwarding state so that the requested multicast traffic will be forwarded back down the tree. When multicast traffic is no longer needed, a device sends a PIM prune up the tree toward the root node to prune (or remove) the unnecessary traffic. As this PIM prune travels hop by hop up the tree, each device updates its forwarding state appropriately. Ultimately, the forwarding state associated with a multicast group or source is removed.

A multicast data sender sends data destined for a multicast group. The designated router (DR) of the sender takes those data packets, unicast-encapsulates them, and sends them directly to the RP. The RP receives these encapsulated data packets, de-encapsulates them, and forwards them onto the shared tree. The packets then follow the (*, G) multicast tree state in the devices on the RP tree, being replicated wherever the RP tree

branches, and eventually reaching all the receivers for that multicast group. The process of encapsulating data packets to the RP is called registering, and the encapsulation packets are called PIM register packets.

Designated Router

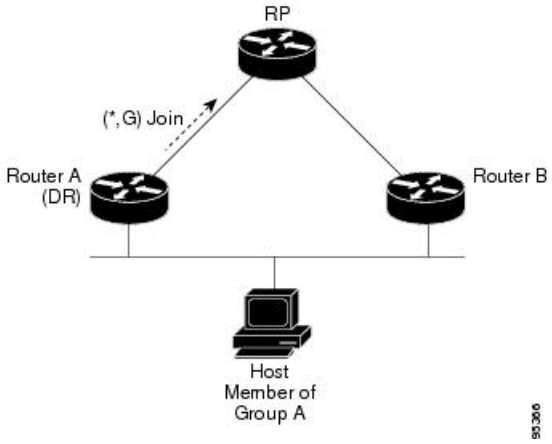
Cisco devices use PIM-SM to forward multicast traffic and follow an election process to select a designated device when there is more than one device on a LAN segment.

The designated router (DR) is responsible for sending PIM register and PIM join and prune messages toward the RP to inform it about active sources and host group membership.

If there are multiple PIM-SM devices on a LAN, a DR must be elected to avoid duplicating multicast traffic for connected hosts. The PIM device with the highest IPv6 address becomes the DR for the LAN unless you choose to force the DR election by use of the **ipv6 pim dr-priority** command. This command allows you to specify the DR priority of each device on the LAN segment (default priority = 1) so that the device with the highest priority will be elected as the DR. If all devices on the LAN segment have the same priority, then the highest IPv6 address is again used as the tiebreaker.

The figure below illustrates what happens on a multiaccess segment. Device A and Device B are connected to a common multiaccess Ethernet segment with Host A as an active receiver for Group A. Only Device A, operating as the DR, sends joins to the RP to construct the shared tree for Group A. If Device B was also permitted to send (*, G) joins to the RP, parallel paths would be created and Host A would receive duplicate multicast traffic. Once Host A begins to source multicast traffic to the group, the DR's responsibility is to send register messages to the RP. If both devices were assigned the responsibility, the RP would receive duplicate multicast packets and result in wastage of bandwidth.

Figure 17: Designated Router Election on a Multiaccess Segment



If the DR should fail, the PIM-SM provides a way to detect the failure of Device A and elect a failover DR. If the DR (Device A) became inoperable, Device B would detect this situation when its neighbor adjacency with Device A timed out. Because Device B has been hearing MLD membership reports from Host A, it already has MLD state for Group A on this interface and would immediately send a join to the RP when it became the new DR. This step reestablishes traffic flow down a new branch of the shared tree via Device B. Additionally, if Host A were sourcing traffic, Device B would initiate a new register process immediately after receiving the next multicast packet from Host A. This action would trigger the RP to join the SPT to Host A via a new branch through Device B.

Tip Two PIM devices are neighbors if there is a direct connection between them. To display your PIM neighbors, use the **show ipv6 pim neighbor** command in privileged EXEC mode.

Note The DR election process is required only on multiaccess LANs.

Rendezvous Point

IPv6 PIM provides embedded RP support. Embedded RP support allows the device to learn RP information using the multicast group destination address instead of the statically configured RP. For devices that are the RP, the device must be statically configured as the RP.

The device searches for embedded RP group addresses in MLD reports or PIM messages and data packets. On finding such an address, the device learns the RP for the group from the address itself. It then uses this learned RP for all protocol activity for the group. For devices that are the RP, the device is advertised as an embedded RP must be configured as the RP.

To select a static RP over an embedded RP, the specific embedded RP group range or mask must be configured in the access list of the static RP. When PIM is configured in sparse mode, you must also choose one or more devices to operate as an RP. An RP is a single common root placed at a chosen point of a shared distribution tree and is configured statically in each box.

PIM DRs forward data from directly connected multicast sources to the RP for distribution down the shared tree. Data is forwarded to the RP in one of two ways:

- Data is encapsulated in register packets and unicast directly to the RP by the first-hop device operating as the DR.
- If the RP has itself joined the source tree, it is multicast-forwarded per the RPF forwarding algorithm described in the PIM-Sparse Mode section.

The RP address is used by first-hop devices to send PIM register messages on behalf of a host sending a packet to the group. The RP address is also used by last-hop devices to send PIM join and prune messages to the RP to inform it about group membership. You must configure the RP address on all devices (including the RP device).

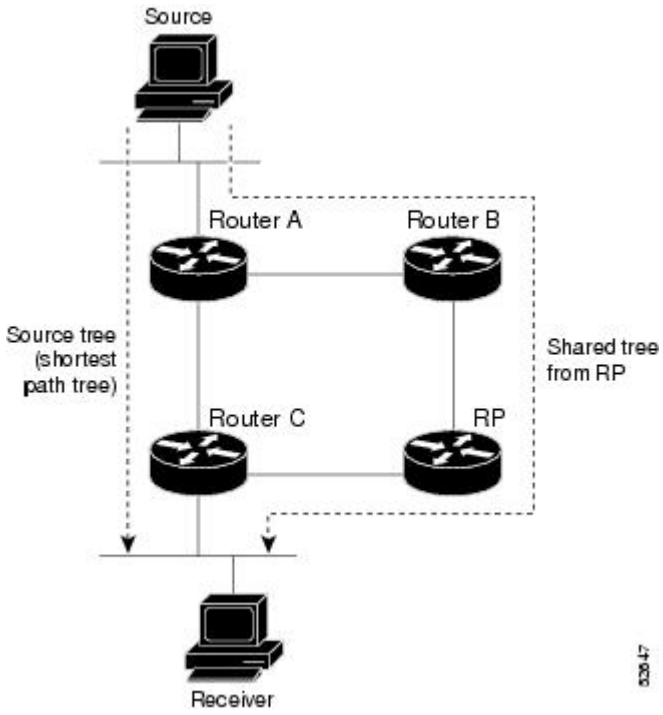
A PIM device can be an RP for more than one group. Only one RP address can be used at a time within a PIM domain for a certain group. The conditions specified by the access list determine for which groups the device is an RP.

IPv6 multicast supports the PIM accept register feature, which is the ability to perform PIM-SM register message filtering at the RP. The user can match an access list or compare the AS path for the registered source with the AS path specified in a route map.

PIM Shared Tree and Source Tree (Shortest-Path Tree)

By default, members of a group receive data from senders to the group across a single data distribution tree rooted at the RP. This type of distribution tree is called shared tree or rendezvous point tree (RPT), as illustrated in the figure below. Data from senders is delivered to the RP for distribution to group members joined to the shared tree.

Figure 18: Shared Tree and Source Tree (Shortest Path Tree)



If the data threshold warrants, leaf devices on the shared tree may initiate a switch to the data distribution tree rooted at the source. This type of distribution tree is called a shortest path tree or source tree. By default, the software switches to a source tree upon receiving the first data packet from a source.

The following process details the move from shared tree to source tree:

1. Receiver joins a group; leaf Device C sends a join message toward the RP.
2. RP puts the link to Device C in its outgoing interface list.
3. Source sends the data; Device A encapsulates the data in the register and sends it to the RP.
4. RP forwards the data down the shared tree to Device C and sends a join message toward the source. At this point, data may arrive twice at Device C, once encapsulated and once natively.
5. When data arrives natively (unencapsulated) at the RP, the RP sends a register-stop message to Device A.
6. By default, receipt of the first data packet prompts Device C to send a join message toward the source.
7. When Device C receives data on (S, G), it sends a prune message for the source up the shared tree.
8. RP deletes the link to Device C from the outgoing interface of (S, G).
9. RP triggers a prune message toward the source.

Join and prune messages are sent for sources and RPs. They are sent hop-by-hop and are processed by each PIM device along the path to the source or RP. Register and register-stop messages are not sent hop-by-hop. They are sent by the designated router (DR) that is directly connected to a source and are received by the RP for the group.

Reverse Path Forwarding

Reverse-path forwarding is used for forwarding multicast datagrams. It functions as follows:

- If a device receives a datagram on an interface it uses to send unicast packets to the source, the packet has arrived on the RPF interface.
- If the packet arrives on the RPF interface, a device forwards the packet out the interfaces present in the outgoing interface list of a multicast routing table entry.
- If the packet does not arrive on the RPF interface, the packet is silently discarded to prevent loops.

PIM uses both source trees and RP-rooted shared trees to forward datagrams; the RPF check is performed differently for each, as follows:

- If a PIM device has source-tree state (that is, an (S, G) entry is present in the multicast routing table), the device performs the RPF check against the IPv6 address of the source of the multicast packet.
- If a PIM device has shared-tree state (and no explicit source-tree state), it performs the RPF check on the RP's address (which is known when members join the group).

Sparse-mode PIM uses the RPF lookup function to determine where it needs to send joins and prunes. (S, G) joins (which are source-tree states) are sent toward the source. (*, G) joins (which are shared-tree states) are sent toward the RP.

Note To do a RPF check, use the **show ipv6 rpf hostname** or **show ipv6 rpf vrf vrf_name hostname** command.

IPv6 Multicast: PIM Sparse Mode

IPv6 multicast provides support for intradomain multicast routing using PIM sparse mode (PIM-SM). PIM-SM uses unicast routing to provide reverse-path information for multicast tree building, but it is not dependent on any particular unicast routing protocol.

IPv6 PIM Passive Mode

A device configured with PIM will always send out PIM hello messages to all interfaces enabled for IPv6 multicast routing, even if the device is configured not to accept PIM messages from any neighbor on the LAN.

IPv6 Multicast: PIM Source-Specific Multicast

The PIM source-specific multicast (SSM) routing protocol supports SSM implementation and is derived from PIM-SM. However, unlike PIM-SM data from all multicast sources are sent when there is a PIM join, the SSM feature forwards datagram traffic to receivers from only those multicast sources that the receivers have explicitly joined, thus optimizing bandwidth utilization and denying unwanted Internet broadcast traffic.

IPv6 Source Specific Multicast Mapping

SSM mapping for IPv6 supports both static and dynamic Domain Name System (DNS) mapping for MLD version 1 receivers. This feature allows deployment of IPv6 SSM with hosts that are incapable of providing

MLD version 2 support in their TCP/IP host stack and their IP multicast receiving application. SSM mapping allows the device to look up the source of a multicast MLD version 1 report either in the running configuration of the device or from a DNS server. The device can then initiate an (S, G) join toward the source.

Selective Packet Discard

Selective Packet Discard (SPD) is a mechanism to manage the process level input queues on the Route Processor (RP). When the router receives bursty traffic, it cannot handle all packets. The input queue is maintained by the hardware interfaces and shared amongst all the sub interfaces. The default input queue size is 75 and is configured per interface using the **hold-queue** *length* {**in** | **out**] command in interface configuration mode. If SPD is not configured, all packets are dropped if the input queue is full when the packet is received.

All packets processed by the router CPU through process switching are queued in the interface input queue. Packets are queued until the queue reaches its maximum length and then additional packets are indiscriminately tail dropped. SPD provides intelligent handling of packets in this queue.

SPD provides preferential treatment during congestion of this interface-to-process-switching queue. As MLD packets fall in the low priority they get dropped during the congestion. To avoid the drops in MLD control packets, increase the below threshold values using the following commands.

- **ipv6 spd queue max-threshold**
- **ipv6 spd queue min-threshold**

Use the **hold-queue** *length* **in** command on the interfaces to receive the MLD packets.

How to Configure IPv6 Multicast

Enabling IPv6 Multicast Routing

IPv6 multicast uses MLD version 2. This version of MLD is fully backward-compatible with MLD version 1 (described in *RFC 2710*). Hosts that support only MLD version 1 will interoperate with a device running MLD version 2. Mixed LANs with both MLD version 1 and MLD version 2 hosts are likewise supported.

Before you begin

You must first enable IPv6 unicast routing on all interfaces of the device on which you want to enable IPv6 multicast routing .

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 multicast-routing** [**vrf** *vrf-name*] **Example:** `Device(config)# ipv6 multicast-routing`	Enables multicast routing on all IPv6-enabled interfaces and enables multicast forwarding for PIM and MLD on all enabled interfaces of the device. • IPv6 multicast routing is disabled by default when IPv6 unicast routing is enabled. IPv6 multicast-routing needs to be enabled for IPv6 multicast routing to function.

Configuring PIM Options

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim** [**vrf** *vrf-name*] **spt-threshold infinity** [**group-list** *access-list-name*]
4. **ipv6 pim** [**vrf** *vrf-name*] **accept-register** {**list** *access-list* | **route-map** *map-name*}
5. **interface** *type number*
6. **ipv6 pim dr-priority** *value*
7. **ipv6 pim hello-interval** *seconds*
8. **ipv6 pim join-prune-interval** *seconds*
9. **exit**
10. **show ipv6 pim** [**vrf** *vrf-name*] **join-prune statistic** [*interface-type*]

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ipv6 pim** [**vrf** *vrf-name*] **spt-threshold infinity** [**group-list** *access-list-name*] **Example:** `Device(config)# ipv6 pim spt-threshold infinity group-list acc-grp-1`	Configures when a PIM leaf device joins the SPT for the specified groups.
Step 4	**ipv6 pim** [**vrf** *vrf-name*] **accept-register** {**list** *access-list* | **route-map** *map-name*} **Example:** `Device(config)# ipv6 pim accept-register route-map reg-filter`	Accepts or rejects registers at the RP.
Step 5	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 6	**ipv6 pim dr-priority** *value* **Example:** `Device(config-if)# ipv6 pim dr-priority 3`	Configures the DR priority on a PIM device.
Step 7	**ipv6 pim hello-interval** *seconds* **Example:** `Device(config-if)# ipv6 pim hello-interval 45`	Configures the frequency of PIM hello messages on an interface.
Step 8	**ipv6 pim join-prune-interval** *seconds* **Example:** `Device(config-if)# ipv6 pim join-prune-interval 75`	Configures periodic join and prune announcement intervals for a specified interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 10	**show ipv6 pim** [**vrf** *vrf-name*] **join-prune statistic** [*interface-type*] **Example:** `Device# show ipv6 pim join-prune statistic`	Displays the average join-prune aggregation for the most recently aggregated packets for each interface.

Configuring PIM-SM and Displaying PIM-SM Information for a Group Range

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim** [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-access-list*] [**bidir**]
4. **end**
5. **show ipv6 pim** [**vrf** *vrf-name*] **interface** [**state-on**] [**state-off**] [*type number*]
6. **show ipv6 pim** [**vrf** *vrf-name*] **group-map** [*group-name* | *group-address*] | [*group-range* | *group-mask*] [**info-source** {**bsr** | **default** | **embedded-rp** | **static**}]
7. **show ipv6 pim** [**vrf** *vrf-name*] **neighbor** [**detail**] [*interface-type interface-number* | **count**]
8. **show ipv6 pim** [**vrf** *vrf-name*] **range-list**[**config**] [*rp-address* | *rp-name*]
9. **show ipv6 pim** [**vrf** *vrf-name*] **tunnel** [*interface-type interface-number*]
10. **debug ipv6 pim** [*group-name* | *group-address* | **interface** *interface-type* | **bsr** | **group** | **mvpn** | **neighbor**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim** [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-access-list*] [**bidir**] **Example:** `Device(config)# ipv6 pim rp-address 2001:DB8::01:800:200E:8C6C acc-grp-1`	Configures the address of a PIM RP for a particular group range.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 5	**show ipv6 pim** [**vrf** *vrf-name*] **interface** [**state-on**] [**state-off**] [*type number*] **Example:** `Device# show ipv6 pim interface`	Displays information about interfaces configured for PIM.

	Command or Action	Purpose
Step 6	**show ipv6 pim** [**vrf** *vrf-name*] **group-map** [*group-name* | *group-address*] | [*group-range* | *group-mask*] [**info-source** {**bsr** | **default** | **embedded-rp** | **static**}] **Example:** `Device# show ipv6 pim group-map`	Displays an IPv6 multicast group mapping table.
Step 7	**show ipv6 pim** [**vrf** *vrf-name*] **neighbor** [**detail**] [*interface-type interface-number* | **count**] **Example:** `Device# show ipv6 pim neighbor`	Displays the PIM neighbors discovered by the Cisco IOS software.
Step 8	**show ipv6 pim** [**vrf** *vrf-name*] **range-list**[**config**] [*rp-address* | *rp-name*] **Example:** `Device# show ipv6 pim range-list`	Displays information about IPv6 multicast range lists.
Step 9	**show ipv6 pim** [**vrf** *vrf-name*] **tunnel** [*interface-type interface-number*] **Example:** `Device# show ipv6 pim tunnel`	Displays information about the PIM register encapsulation and de-encapsulation tunnels on an interface.
Step 10	**debug ipv6 pim** [*group-name* | *group-address* | **interface** *interface-type* | **bsr** | **group** | **mvpn** | **neighbor**] **Example:** `Device# debug ipv6 pim`	Enables debugging on PIM protocol activity.

Resetting the PIM Traffic Counters

If PIM malfunctions, or in order to verify that the expected number of PIM packets are received and sent, clear PIM traffic counters. Once the traffic counters are cleared, you can verify that PIM is functioning correctly and that PIM packets are being received and sent correctly.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 pim** [**vrf** *vrf-name*] **traffic**
3. **show ipv6 pim** [**vrf** *vrf-name*] **traffic**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 pim** [**vrf** *vrf-name*] **traffic** **Example:** `Device# clear ipv6 pim traffic`	Resets the PIM traffic counters.
Step 3	**show ipv6 pim** [**vrf** *vrf-name*] **traffic** **Example:** `Device# show ipv6 pim traffic`	Displays the PIM traffic counters.

Clearing the PIM Topology Table to Reset the MRIB Connection

No configuration is necessary to use the MRIB. However, users may in certain situations want to clear the PIM topology table in order to reset the MRIB connection and verify MRIB information.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 pim** [**vrf** *vrf-name*] **topology** [*group-name* | *group-address*]
3. **show ipv6 mrib** [**vrf** *vrf-name*] **client** [**filter**] [**name** {*client-name* | *client-name* **:** *client-id*}]
4. **show ipv6 mrib** [**vrf** *vrf-name*] **route** [**link-local**| **summary** | [*sourceaddress-or-name* | *****] [*groupname-or-address* [*prefix-length*]]]
5. **show ipv6 pim** [**vrf** *vrf-name*] **topology** [*groupname-or-address* [*sourcename-or-address*] | **link-local** | **route-count** [**detail**]]
6. **debug ipv6 mrib** [**vrf** *vrf-name*] **client**
7. **debug ipv6 mrib** [**vrf** *vrf-name*] **io**
8. **debug ipv6 mrib proxy**
9. **debug ipv6 mrib** [**vrf** *vrf-name*] **route** [*group-name* | *group-address*]
10. **debug ipv6 mrib** [**vrf** *vrf-name*] **table**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**clear ipv6 pim** [**vrf** *vrf-name*] **topology** [*group-name* | *group-address*] **Example:** `Device# clear ipv6 pim topology FF04::10`	Clears the PIM topology table.
Step 3	**show ipv6 mrib** [**vrf** *vrf-name*] **client** [**filter**] [**name** {*client-name* | *client-name* **:** *client-id*}] **Example:** `Device# show ipv6 mrib client`	Displays multicast-related information about an interface.
Step 4	**show ipv6 mrib** [**vrf** *vrf-name*] **route** [**link-local**| **summary** | [*sourceaddress-or-name* | *****] [*groupname-or-address* [*prefix-length*]]] **Example:** `Device# show ipv6 mrib route`	Displays the MRIB route information.
Step 5	**show ipv6 pim** [**vrf** *vrf-name*] **topology** [*groupname-or-address* [*sourcename-or-address*] | **link-local** | **route-count** [**detail**]] **Example:** `Device# show ipv6 pim topology`	Displays PIM topology table information for a specific group or all groups.
Step 6	**debug ipv6 mrib** [**vrf** *vrf-name*] **client** **Example:** `Device# debug ipv6 mrib client`	Enables debugging on MRIB client management activity.
Step 7	**debug ipv6 mrib** [**vrf** *vrf-name*] **io** **Example:** `Device# debug ipv6 mrib io`	Enables debugging on MRIB I/O events.
Step 8	**debug ipv6 mrib proxy** **Example:** `Device# debug ipv6 mrib proxy`	Enables debugging on MRIB proxy activity between the route processor and line cards on distributed router platforms.
Step 9	**debug ipv6 mrib** [**vrf** *vrf-name*] **route** [*group-name* | *group-address*] **Example:** `Device# debug ipv6 mrib route`	Displays information about MRIB routing entry-related activity.

	Command or Action	Purpose
Step 10	**debug ipv6 mrib** [**vrf** *vrf-name*] **table** **Example:** `Device# debug ipv6 mrib table`	Enables debugging on MRIB table management activity.

Turning Off IPv6 PIM on a Specified Interface

A user might want only specified interfaces to perform IPv6 multicast and will therefore want to turn off PIM on a specified interface.

Note Though IOS supports disabling PIM on an interface, this is not possible on RSP3 platform due to caveat. Ipv6 multicast packets will still get punted to CPU even if PIM is turned off on the interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ipv6 pim**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**no ipv6 pim** **Example:** `Device(config-if)# no ipv6 pim`	Turns off IPv6 PIM on a specified interface.

Disabling Embedded RP Support in IPv6 PIM

A user might want to disable embedded RP support on an interface if all of the devices in the domain do not support embedded RP.

Note This task disables PIM completely, not just embedded RP support in IPv6 PIM.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ipv6 pim** [**vrf** *vrf-name*] **rp embedded**
4. **interface** *type number*
5. **no ipv6 pim**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ipv6 pim** [**vrf** *vrf-name*] **rp embedded** **Example:** `Device(config)# no ipv6 pim rp embedded`	Disables embedded RP support in IPv6 PIM.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 5	**no ipv6 pim** **Example:** `Device(config-if)# no ipv6 pim`	Turns off IPv6 PIM on a specified interface.

Configuring IPv6 SSM

When the SSM mapping feature is enabled, DNS-based SSM mapping is automatically enabled, which means that the device will look up the source of a multicast MLD version 1 report from a DNS server.

You can configure either DNS-based or static SSM mapping, depending on your device configuration. If you choose to use static SSM mapping, you can configure multiple static SSM mappings. If multiple static SSM mappings are configured, the source addresses of all matching access lists will be used.

Before you begin

Note To use DNS-based SSM mapping, the device needs to find at least one correctly configured DNS server to which the device can be directly attached.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 mld** [**vrf** *vrf-name*] **ssm-map enable**
4. **no ipv6 mld** [**vrf** *vrf-name*] **ssm-map query dns**
5. **ipv6 mld** [**vrf** *vrf-name*] **ssm-map static** *access-list source-address*
6. **end**
7. **show ipv6 mld** [**vrf** *vrf-name*] **ssm-map** [*source-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 mld** [**vrf** *vrf-name*] **ssm-map enable** **Example:** `Device(config)# ipv6 mld ssm-map enable`	Enables the SSM mapping feature for groups in the configured SSM range.
Step 4	**no ipv6 mld** [**vrf** *vrf-name*] **ssm-map query dns** **Example:** `Device(config)# no ipv6 mld ssm-map query dns`	Disables DNS-based SSM mapping.

	Command or Action	Purpose
Step 5	**ipv6 mld** [**vrf** *vrf-name*] **ssm-map static** *access-list source-address* **Example:** `Device(config)# ipv6 mld ssm-map static SSM_MAP_ACL_2 2001:DB8:1::1`	Configures static SSM mappings.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show ipv6 mld** [**vrf** *vrf-name*] **ssm-map** [*source-address*] **Example:** `Device# show ipv6 mld ssm-map`	Displays SSM mapping information.

Configuring PIM Passive/MLD State Limit

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast pim-passive-enable**
4. **ipv6 mld state-limit** *number*
5. **interface** *type number*
6. **ipv6 pim passive**
7. **ipv6 mld limit** *number*
8. **no ipv6 mld router**
9. **show ipv6 mld** *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.

	Command or Action	Purpose
Step 3	**ipv6 multicast pim-passive-enable** **Example:** `Device(config)# ipv6 multicast pim-passive-enable`	Enables the PIM passive feature on an IPv6 device.
Step 4	**ipv6 mld state-limit** *number* **Example:** `Device(config)# ipv6 mld state-limit 100`	(Optional) Specifies maximum number of dynamic MLD groups allowed on a router.
Step 5	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 6	**ipv6 pim passive** **Example:** `Device(config-if)# ipv6 pim passive`	Enables the PIM passive feature on a specific interface.
Step 7	**ipv6 mld limit** *number* **Example:** `Device(config-if)# ipv6 mld limit 300`	(Optional) Configure the per-interface MLD state limit. You can use this command to limit the dynamic MLD groups joined.
Step 8	**no ipv6 mld router** **Example:** `Device(config-if)# no ipv6 mld router`	(Optional) Prevents the interface from processing MLD v1/v2 joins sent through it or to prune from a group it has already joined. To enable the interface to start receiving MLD reports again, use **ipv6 mld router** command.
Step 9	**show ipv6 mld** *interface* **Example:** `Device(config-if)# show ipv6 mld interface 1/0/0`	(Optional) Displays MLD information about the interface. You can use this command to determine which interface acts as a querier.

Configuration Examples for IPv6 Multicast

Example: Enabling IPv6 Multicast Routing

The following example enables multicast routing on all interfaces and also enables multicast forwarding for PIM and MLD on all enabled interfaces of the device.

```
Device> enable
```

```
Device# configure terminal
Device(config)# ipv6 multicast-routing
```

Example: Configuring PIM Options

The following example sets the DR priority, the PIM hello interval, and the periodic join on the interface.

```
Device(config)# interface gigabitethernet 1/0/0
Device(config)# ipv6 pim hello-interval 60
Device(config)# ipv6 pim dr-priority 3
```

Example: Configuring PIM

The following example shows how to configure a device to use PIM-SM using 2001:DB8::1 as the RP. It sets the SPT threshold to infinity to prevent switchover to the source tree when a source starts sending traffic and sets a filter on all sources that do not have a local multicast BGP prefix.

```
Device(config)# ipv6 multicast-routing
Device(config)# ipv6 pim rp-address 2001:DB8::1
Device(config)# ipv6 pim spt-threshold infinity
Device(config)# ipv6 pim accept-register route-map reg-filter
```

Example: Displaying PIM-SM Information for a Group Range

This example displays information about interfaces configured for PIM:

```
Device# show ipv6 pim interface state-on

Interface          PIM   Nbr   Hello  DR
                         Count Intvl  Prior

Gi0/1/2            on    0     30     1
    Address: FE80::D2C2:82FF:FE17:F392
    DR     : this system
Gi0/1/5            on    1     30     1
    Address: FE80::D2C2:82FF:FE17:F395
    DR     : FE80::D2C2:82FF:FE17:FAA5
Loopback0          on    0     30     1
    Address: FE80::D2C2:82FF:FE17:F380
    DR     : this system
```

This example displays an IPv6 multicast group mapping table:

```
Device# show ipv6 pim group-map

FF33::/32*
      SSM
      Info source:Static
      Uptime:00:08:32, Groups:0
  FF34::/32*
      SSM
      Info source:Static
      Uptime:00:09:42, Groups:0
```

This example displays information about IPv6 multicast range lists:

```
Device# show ipv6 pim range-list

config SSM Exp:never Learnt from :::
 FF33::/32 Up:00:26:33
 FF34::/32 Up:00:26:33
 FF35::/32 Up:00:26:33
 FF36::/32 Up:00:26:33
 FF37::/32 Up:00:26:33
 FF38::/32 Up:00:26:33
 FF39::/32 Up:00:26:33
 FF3A::/32 Up:00:26:33
 FF3B::/32 Up:00:26:33
 FF3C::/32 Up:00:26:33
 FF3D::/32 Up:00:26:33
 FF3E::/32 Up:00:26:33
 FF3F::/32 Up:00:26:33
config SM RP:40::1:1:1 Exp:never Learnt from :::
 FF13::/64 Up:00:03:50
config SM RP:40::1:1:3 Exp:never Learnt from :::
 FF09::/64 Up:00:03:50
```

Example: Displaying IPv6 PIM Topology Information

```
Device# show ipv6 pim topology

IP PIM Multicast Topology Table
Entry state: (*/S,G)[RPT/SPT] Protocol Uptime Info Upstream Mode
Entry flags: KAT - Keep Alive Timer, AA - Assume Alive, PA - Probe Alive,
    RA - Really Alive, LH - Last Hop, DSS - Don't Signal Sources,
    RR - Register Received, SR - Sending Registers, E - MSDP External,
    DCC - Don't Check Connected, Y - Joined MDT-data group,
    y - Sending to MDT-data group
    BGS - BGP Signal Sent, !BGS - BGP signal suppressed
    SAS - BGP Src-Act Sent, SAR - BGP Src-Act Received
Interface state: Name, Uptime, Fwd, Info
Interface flags: LI - Local Interest, LD - Local Disinterest,
    II - Internal Interest, ID - Internal Disinterest,
    LH - Last Hop, AS - Assert, AB - Admin Boundary, BS - BGP Signal,
    BP - BGP Shared-Tree Prune, BPT - BGP Prune Time

(*,FF08::1)
SM UP: 00:04:36 JP: Join(00:00:28) Flags:
RP: 8001::1*
RPF: Tunnel1,8001::1*
  Gi0/1/5            00:04:36  fwd     Join(00:03:01)

(3001::5,FF08::1)
SM SPT UP: 00:04:57 JP: Join(never) Flags: KAT(00:02:12) RA
RPF: GigabitEthernet0/1/2,3001::5*
  Gi0/1/5            00:04:36  fwd     Join(00:03:01)
```

Property Type	Property Value	Property Description

-

Example: Displaying Information About PIM Traffic

```
Device# show ipv6 pim traffic

PIM Traffic Counters
Elapsed time since counters cleared:00:05:29

                              Received     Sent
Valid PIM Packets               22          22
Hello                           22          22
Join-Prune                      0           0
Register                        0           0
Register Stop                   0           0
Assert                          0           0
Bidir DF Election               0           0

Errors:
Malformed Packets                           0
Bad Checksums                               0
Send Errors                                 0
Packet Sent on Loopback Errors              0
Packets Received on PIM-disabled Interface  0
Packets Received with Unknown PIM Version   0
```

Example: Disabling Embedded RP Support in IPv6 PIM

The following example disables embedded RP support on IPv6 PIM:

```
Device(config)# ipv6 multicast-routing
Device(config)# no ipv6 pim rp embedded
```

Example: IPv6 SSM Mapping

```
Device# show ipv6 mld ssm-map 2001:DB8::1

 Group address  : 2001:DB8::1
 Group mode ssm : TRUE
 Database       : STATIC
 Source list    : 2001:DB8::2
                  2001:DB8::3

Device# show ipv6 mld ssm-map 2001:DB8::2

 Group address  : 2001:DB8::2
 Group mode ssm : TRUE
 Database       : DNS
 Source list    : 2001:DB8::3
                  2001:DB8::1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mcl/allreleasemcl/all-book.html

Standards and RFCs

Standard/RFC	Title
No specific Standards and RFCs are supported by the features in this document.	—

MIBs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

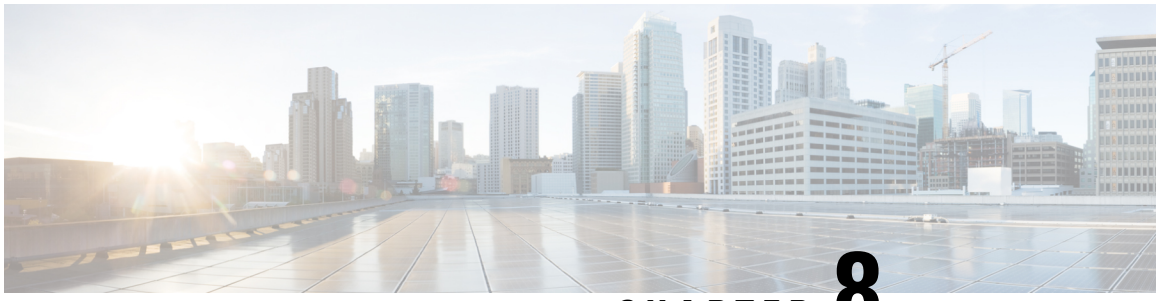
Feature Information for IPv6 Multicast PIM

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

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

Table 5: Feature Information for IPv6 Multicast PIM

Feature Name	Releases	Feature Information
IPv6 Multicast PIM	Cisco IOS XE Release 3.14.0S	This feature was introduced on the Cisco ASR 920 Series Aggregation Services Router (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D, ASR-920-10SZ-PD, ASR-920-24SZ-IM, ASR-920-24SZ-M, ASR-920-24TZ-M).



CHAPTER 8

IGMP Snooping

This module describes how to enable and configure the Ethernet Virtual Connection (EVC)-based IP Multicast Internet Group Management Protocol (IGMP) Snooping feature both globally and on bridge domains.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for IGMP Snooping

- IGMP snooping is implemented based on layer 2 multicast frames.
- Basic IGMP v3 snooping support (BISS) is supported.
- POP operation for all vlan tags should be configured on EFP.
- Bridge domain (BD) interfaces from 1 to 4094 support IGMP snooping.
- IGMP static joins are *not* supported.

Autogenerated Files and Directories

Caution Any autogenerated file in the bootflash: directory should not be deleted, renamed, moved, or altered in any way unless directed by customer support; altering these files can have unpredictable consequences for system performance.

Table 6: Autogenerated Files

File or Directory	Description
crashinfo files	A crashinfo file may appear in the bootflash: file system. Crashinfo files are useful for tuning and troubleshooting, but are not related to router operations: you can erase them without impacting the router's performance.
core files	The bootflash/core directory is the storage area for .core files. **Warning** Do not erase or move the core directory.
lost+found directory	This directory is created on bootup if a system check is performed. Its appearance is completely normal and does not indicate any issues with the router.
tracelogs files	The storage area for trace files is bootflash/tracelogs. Trace files are useful for troubleshooting; you can access trace files using diagnostic mode to gather information related to the IOS failure. **Warning** Do not erase or move the tracelog directory.

Restrictions for IGMP Snooping

- IGMP snooping is *not* supported on Bridge Domain (BD) interfaces greater than 4094.
- Static mrouter configuration is *not* supported.
- IGMP snooping is *not* supported for pseudowires.
- IGMP snooping is supported only on the EFP, Trunk EFPs, port-channel EFP, and port-channel Trunk EFPs.
- Layer2 multicast is not suported with IGMP snooping when static joins are configured in EFP or TEFP. However, Layer2 multicast with IGMP snooping is supported for dynamic joins configured on the EFP or TEFP.
- Starting with Cisco IOS Release 3.13, for Protocol Independent Multicast (PIM) Source Specific Multicast (SSM), with Bridge Domain Interface (BDI) as Incoming Interface (IIF), IGMP Snooping is *not* supported on the corresponding Bridge Domain (BD).

- Starting with Cisco IOS Release 3.13, for Protocol Independent Multicast Sparse Mode (PIM-SM), with Bridge Domain Interface BDI as Incoming Interface (IIF), IGMP Snooping is *not* supported on the corresponding Bridge Domain (BD) in non Designated Router (DR) node.

Information About IGMP Snooping

IGMP Snooping

IP Multicast Internet Group Management Protocol (IGMP), which runs at Layer 3 on a multicast device, generates Layer 3 IGMP queries in subnets where the multicast traffic must be routed. IGMP (on a device) sends out periodic general IGMP queries.

IGMP Snooping is an Ethernet Virtual Circuit (EVC)-based feature set. EVC decouples the concept of VLAN and broadcast domain. An EVC is an end-to-end representation of a single instance of a Layer 2 service being offered by a provider. In the Cisco EVC framework, bridge domains are made up of one or more Layer 2 interfaces known as service instances. A service instance is the instantiation of an EVC on a given port on a given device. A service instance is associated with a bridge domain based on the configuration.

When you enable EVC-based IGMP snooping on a bridge domain, the bridge domain interface responds at Layer 2 to the IGMP queries with only one IGMP join request per Layer 2 multicast group. Each bridge domain represents a Layer 2 broadcast domain. The bridge domain interface creates one entry per subnet in the Layer 2 forwarding table for each Layer 2 multicast group from which it receives an IGMP join request. All hosts interested in this multicast traffic send IGMP join requests and are added to the forwarding table entry. During a Layer 2 lookup on a bridge domain to which the bridge domain interface belongs, the bridge domain forwards the packets to the correct EFP. When the bridge domain interface hears the IGMP Leave group message from a host, it removes the table entry of the host.

IGMP snooping is supported on Metro IP and Metro Aggregate licenses on the Cisco ASR 920 Series Routers. IGMP snooping is supported with MSTP, REP, and G.8032. IGMP snooping is also supported on the port-channel interfaces.

How to Configure IGMP Snooping

Enabling IGMP Snooping

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping**
4. **bridge-domain** *bridge-id*
5. **ip igmp snooping**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping** **Example:** `Device(config)# ip igmp snooping`	Globally enables IGMP snooping after it has been disabled.
Step 4	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 100`	(Optional) Enters bridge domain configuration mode.
Step 5	**ip igmp snooping** **Example:** `Device(config-bdomain)# ip igmp snooping`	(Optional) Enables IGMP snooping on the bridge domain interface being configured. • Required only if IGMP snooping was previously explicitly disabled on the specified bridge domain.
Step 6	**end** **Example:** `Device(config-bdomain)# end`	Returns to privileged EXEC mode.

Configuring IGMP Snooping Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping robustness-variable** *variable*
4. **ip igmp snooping report-suppression**
5. **ip igmp snooping last-member-query-count** *count*
6. **ip igmp snooping last-member-query-interval** *interval*
7. **ip igmp snooping check ttl**
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.
Step 3	**ip igmp snooping robustness-variable** *variable* **Example:** `Device(config)# ip igmp snooping robustness-variable 3`	Configures the IGMP defined robustness variable .
Step 4	**ip igmp snooping report-suppression** **Example:** `Device(config)# ip igmp snooping report-suppression`	Enables report suppression for IGMP snooping.
Step 5	**ip igmp snooping last-member-query-count** *count* **Example:** `Device(config)# ip igmp snooping last-member-query-count 5`	Configures how often IGMP snooping sends query messages in response to receiving an IGMP leave message. The default is 2.
Step 6	**ip igmp snooping last-member-query-interval** *interval* **Example:** `Device(config)# ip igmp snooping last-member-query-interval 200`	Configures the length of time after which the group record is deleted if no reports are received. The default is 1000 milliseconds.
Step 7	**ip igmp snooping check ttl** **Example:** `Device(config)# ip igmp snooping check ttl`	Enforces IGMP snooping check.
Step 8	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring IGMP Snooping on a Bridge Domain

Before you begin

- The bridge domain must be created. See the Ethernet Virtual Connections Configuration for configuration information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge-domain** *bridge-id*
4. **ip igmp snooping immediate-leave**
5. **ip igmp snooping last-member-query-count** *count*
6. **ip igmp snooping last-member-query-interval** *interval*
7. **ip igmp snooping robustness-variable** *variable*
8. **ip igmp snooping report-suppression**
9. **ip igmp snooping check ttl**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 100`	Enters bridge domain configuration mode.
Step 4	**ip igmp snooping immediate-leave** **Example:** `Device(config-bdomain)# ip igmp snooping immediate-leave`	Enables IGMPv2 immediate-leave processing. **Note** When both immediate-leave processing and the query count are configured, fast-leave processing takes precedence.

	Command or Action	Purpose
Step 5	**ip igmp snooping last-member-query-count** *count* **Example:** `Device(config-bdomain)# ip igmp snooping last-member-query-count 5`	Sets the count for last member query messages sent in response to receiving an IGMP leave message. The valid range is 1 to 7. The default is 2 milliseconds. **Note** When both immediate-leave processing and the query count are configured, fast-leave processing takes precedence.
Step 6	**ip igmp snooping last-member-query-interval** *interval* **Example:** `Device(config-bdomain)# ip igmp snooping last-member-query-interval 2000`	Sets the last member query interval of the bridge domain. The valid range is from 100 to 32767. The default is 1000 milliseconds.
Step 7	**ip igmp snooping robustness-variable** *variable* **Example:** `Device(config-bdomain)# ip igmp snooping robustness-variable 3`	Configures the IGMP snooping robustness variable. The default is 2.
Step 8	**ip igmp snooping report-suppression** **Example:** `Device(config-bdomain)# ip igmp snooping report-suppression`	Enables report suppression for all hosts on the bridge domain.
Step 9	**ip igmp snooping check ttl** **Example:** `Device(config-bdomain)# ip igmp snooping check ttl`	Enforces IGMP snooping check.
Step 10	**end** **Example:** `Device(config-bdomain)# end`	Returns to privileged EXEC mode.

Disabling IGMP Snooping Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip igmp snooping**
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	**no ip igmp snooping** **Example:** `Device(config)# no ip igmp snooping`	Disables IGMP snooping on the router.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Disabling IGMP Snooping on a Bridge Domain

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge-domain** *bridge-id*
4. **no ip igmp snooping**
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	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 4000`	Enters bridge domain configuration mode.

	Command or Action	Purpose
Step 4	**no ip igmp snooping** **Example:** `Device(config-bdomain)# no ip igmp snooping`	Disables IGMP snooping on the bridge domain.
Step 5	**end** **Example:** `Device(config-bdomain)# end`	Returns to privileged EXEC mode.

Verifying IGMP Snooping

Use these commands to verify IGMP Snooping on the router.

- **show ip igmp snooping**

  This command displays the IGMP snooping configuration globally on the router. The following is a sample output from the command:

```
Router# show ip igmp snooping

Global IGMP Snooping configuration:
-------------------------------------------
IGMP snooping Oper State     : Enabled
IGMPv3 snooping (minimal)    : Enabled
Report suppression           : Enabled
TCN solicit query            : Enabled
Robustness variable          : 3
Last member query count      : 2
Last member query interval   : 200
Check TTL=1                  : Yes
Check Router-Alert-Option    : No

Vlan 1:
--------
IGMP snooping Admin State           : Enabled
IGMP snooping Oper State            : Enabled
IGMPv2 immediate leave              : Disabled
Report suppression                  : Enabled
Robustness variable                 : 3
Last member query count             : 2
Last member query interval          : 200
Check TTL=1                         : Yes
Check Router-Alert-Option           : Yes
.
.
.
```

- **show ip igmp snooping** [**bd** *bd-id*]

  This command displays configuration for IGMP snooping by bridge domain. The following is a sample output from the command:

```
Router# show ip igmp snooping bd 100

Global IGMP Snooping configuration:
-------------------------------------------
IGMP snooping Oper State     : Enabled
```

```
IGMPv3 snooping (minimal)    : Enabled
Report suppression           : Enabled
TCN solicit query            : Enabled
Robustness variable          : 3
Last member query count      : 2
Last member query interval   : 200
Check TTL=1                  : Yes
Check Router-Alert-Option    : No

Vlan 100:
--------
IGMP snooping Admin State           : Enabled
IGMP snooping Oper State            : Enabled
IGMPv2 immediate leave              : Disabled
Report suppression                  : Enabled
Robustness variable                 : 3
Last member query count             : 2
Last member query interval          : 200
Check TTL=1                         : Yes
Check Router-Alert-Option           : Yes
Query Interval                      : 0
Max Response Time                   : 10000
```

- **show ip igmp snooping groups bd** *bd-id* **count**

  This command displays snooping information for groups by bridge domain. This is a sample output from the command:

```
Router# show ip igmp snooping group bd 4000 count

Total number of groups in Vlan 4000:   2
Total number of  (S,G) in Vlan 4000:   0
```

- **show ip igmp snooping groups count**

  This command displays snooping information for groups. This is a sample output from the command:

```
Router# show ip igmp snooping groups count

Total number of groups:   4
Total number of  (S,G):   0
```

- **show ip igmp snooping counters** [**bd** *bd-id*]

  This command displays IGMP snooping counters, globally or by bridge domain. This is the sample output from this command where Ovr and Und represent oversize and undersize respectively:

```
Router# show ip igmp snooping counters

Counters of group "IGMP snooping  counters                          " overall there
are 15   counters
 Type                                                | Value      | Ovr | Und
-----------------------------------------------------+------------+-----+----
RX processed Query Count                             | 0          |     |
RX processed Group Specific Query                    | 0          |     |
RX processed Join                                    | 0          |     |
RX processed Leave                                   | 0          |     |
RX processed Total Valid Packets                     | 0          |     |
RX processed Other Packets                           | 0          |     |
RX Packets dropped for sanity errors                 | 0          |     |
RX Packets dropped for checksum errors               | 0          |     |
RX Packets dropped for header length errors          | 0          |     |
```

```
RX Packets dropped for other errors                | 0          |     |
RX processed Topology change notification          | 0          |     |
TX processed Query Count                           | 0          |     |
TX processed Group Specific Query                  | 0          |     |
TX processed Join                                  | 0          |     |
TX processed Leave                                 | 0          |     |

Counters of group "IGMP snooping  V3 counters                        " overall there
are 18   counters
RX processed V3 AllOW NEW                          | 0          |     |
RX processed V3 BLOCK OLD                          | 0          |     |

 Type                                              | Value      | Ovr | Und
---------------------------------------------------+------------+-----+----
RX processed V3 MODE IS INCLUDE                    | 0          |     |
RX processed V3 MODE IS EXCLUDE                    | 0          |     |
RX processed V3 CHANGE TO INCLUDE                  | 0          |     |
RX processed V3 CHANGE TO EXCLUDE                  | 0          |     |
RX processed V3 Query                              | 0          |     |
RX processed V3 Group Specific Query               | 0          |     |
RX processed V3 GSS Query                          | 0          |     |
TX processed V3 ALLOW NEW                          | 0          |     |
TX processed V3 BLOCK OLD                          | 0          |     |
TX processed V3 MODE IS INCLUDE                    | 0          |     |
TX processed V3 MODE IS EXCLUDE                    | 0          |     |
TX processed V3 CHANGE TO INCLUDE                  | 0          |     |
TX processed V3 CHANGE TO EXCLUDE                  | 0          |     |
TX processed V3 Query                              | 0          |     |
TX processed V3 Group Specific Query               | 0          |     |
TX processed V3 GSS Query                          | 0          |     |
```

- **show ip igmp snooping mrouter**

  [**bd** *bd-id*]

  This command displays multicast ports, globally or by bridge domain.. This is a sample output from the command:

```
Router# show ip igmp snooping mrouter

Vlan    ports
----    -----
100    Gi0/3/4-efp1(dynamic)
 10    Gi0/4/5-tefp1(dynamic)
100    Po64-efp100(dynamic)
```

- **show ip igmp snooping querier**

  [**bd** *bd-id*]

  This command displays the IGMP querier information globally or by a bridge domain. This is a sample output from the command:

```
Router# show ip igmp snooping querier

Vlan      IP Address               IGMP Version   Port
-----------------------------------------------------------------
100       10.0.0.2                 v2            Gi0/3/4-efp1
10        10.0.0.2                 v2            Gi0/4/5-tefp1
100       30.1.1.12                v2            Po64-efp100
```

- **show ip igmp snooping group**

  This command displays the IGMP snooping information about multicast groups by VLAN. This is a sample output from the command:

```
Router# show ip igmp snooping  group

Flags: I -- IGMP snooping, S -- Static, P -- PIM snooping, A -- ASM mode
Vlan      Group/source             Type        Version     Port List
------------------------------------------------------------------------
100       226.0.1.1                I           v2          Gi0/1/1-efp100
10        225.1.1.1                I           v2          Gi0/4/2-tefp1
100       235.1.1.3                I           v2          Po64-efp1
```

- **show ip igmp snooping group bd**

  This command displays the BD level IGMP snooping information. This is a sample output from the command:

```
Router# show ip igmp snooping  group bd 100 226.0.1.1

Flags: I -- IGMP snooping, S -- Static, P -- PIM snooping, A -- ASM mode
Vlan      Group/source             Type        Version     Port List
------------------------------------------------------------------------
100       226.0.1.1                I           v2          Gi0/1/1-efp100
100       235.1.1.3                I           v2          Po64-efp1

For Scale scenarios: Check the Snooping groups count per BD level.

Router# show ip igmp snooping  group bd 100 count

Total number of groups in Vlan 100:   1
Total number of  (S,G) in Vlan 100:   0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mcl/allreleasemcl/all-book.html

Standards and RFCs

Standard/RFC	Title
No specific Standards and RFCs are supported by the features in this document.	—

MIBs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IGMP Snooping

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

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

Table 7: Feature Information for IGMP Snooping

Feature Name	Releases	Feature Information
IGMP Snooping	Cisco IOS XE Release 3.14.0S	This feature was introduced on the Cisco ASR 920 Series Aggregation Services Router (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D, ASR-920-10SZ-PD, ASR-920-24SZ-IM, ASR-920-24SZ-M, ASR-920-24TZ-M) .